Artificial Intelligence Methods and Evaluation Strategies for Detecting Future Customer Needs from User Generated Content

by

David Kilroy

This thesis is submitted to University College Dublin in fulfilment of the requirements for the degree of Doctor of Philosophy

in

School of Computer Science

Principal Supervisor: Dr. Simon Caton

Secondary Supervisor: Dr. Graham Healy

Head of School: Associate Professor Neil Hurley

RSP Members: Professor Pádraig Cunningham & Dr. Aonghus Lawlor

June, 2024

Acknowledgments

Foremost, I would like to offer gratitude to my supervisors Simon Caton and Graham Healy for providing guidance and support throughout the PhD. I'm genuinely very lucky to have had the opportunity to work with you both. Secondly, I'd like to thank the ML-Labs for giving me an opportunity to pursue this degree. Thirdly, I would like to thank the industry partner of this PhD and the people I worked with there. Finally, thanks to my family and friends for your support.

Statement of Authorship

I hereby certify that the submitted work is my own work, was completed while registered as a candidate for the degree stated on the Title Page, and I have not obtained a degree elsewhere on the basis of the research presented in this submitted work.

Signed: _____

Table of Contents

List of Tables

List of Figures

List of Publications

- Kilroy, D., Healy, G. & Caton, S. Using Machine Learning To Improve Lead Times In The Identification of Emerging Customer Needs. IEEE Access 10, 37774–37795 (2022).
- Kilroy, D., Caton, S. & Healy, G. The Trending Customer Needs (TCN) Dataset: A Benchmarking and Automated Evaluation Approach for New Product Development. In procededings of 56th Hawaii International Conference on System Sciences (HICSS). (2023).
- Kilroy, D., Healy, G. & Caton, S. Prediction of Future Customer Needs Using Machine Learning Across Multiple Product Categories (Under Review, Third Round). PloS One (2024).
- Kilroy, D., Caton, S. & Healy, G Comparing ML-based and Human Product Development Approaches at Predicting Future Customer Needs (in Preparation) (2024).

Abstract

Research has shown that listening to customers when developing products leads to increased satisfaction, which drives profits. In recent years, approaches have been applied to automatically extract customer needs from User Generated Content, such as social media (e.g. Twitter) and blog websites (e.g. Quora). Specifically, these studies use techniques from Artificial Intelligence, and in particular Machine Learning, to gather these requirements. The majority of these studies focus on summarizing currently discussed customer needs from specific product models (e.g. current needs for the Samsung Galaxy mobile). However, there is a lack of studies that focus on the challenging task of predicting needs from product categories that will be of importance in the future and are therefore unaddressed (e.g. future needs for mobile phones). Doing so would allow businesses to discover the "Next Big Thing".

Therefore, this thesis investigates techniques and evaluation approaches that address the task of predicting future customer needs for product categories on Reddit. The main method used to solve the task is keyphrase ranking/classification, which orders/classifies candidate keyphrases by the extent to which they are likely to be future customer needs. The thesis first presents two ground truth datasets to train/evaluate algorithms run over Reddit. To curate such datasets, needs are extracted from Mintel GNPD - a large database of new-to-market product descriptions. These datasets use human annotators in the curation process to ensure their quality. After, two separate algorithms for future customer needs prediction are presented. The first algorithm is a rule-based approach that applies various text mining techniques to rank keyphrases that are likely to be future customer needs. This approach is evaluated on the task of predicting future customer needs with one product category: Toothpaste. The second algorithm is a machine learning approach that uses Multivariate Time Series Classification techniques to classify candidate keyphrase instances represented by 1263 univariate time series features. These 1263 features come from 10 families of features, all selected for the task of predicting future customer needs. During the evaluation, this approach predicts 15 product categories, all in the area of Consumer Packaged Goods. Finally, a user study is undertaken with participants from a multi-billion dollar (USD) company to evaluate whether they found the output of the machine learning algorithm useful in discovering new product opportunities. This user study also proposes an evaluation methodology that would allow companies to partake in evaluation studies with researchers without disclosing any proprietary information about the products they plan to make.

The first outcome arising from this thesis is that future customer needs can be predicted with performance useful for the purposes of product development. This is shown by both the rule-based and machine learning approaches achieving high-performance metrics for the task when compared to baseline methods. The second outcome shows that ground truth datasets can be constructed for the task of predicting/evaluating future customer needs effectively. These datasets exhibited a high Inter Annotator Agreement between annotators

and high accuracy on a set of gold standard labels. The third outcome shows that a Multi-Task Learning model made from an extension of the machine learning approach can predict future customer needs effectively for a category it has seen and not seen in the training process. Such an outcome is significant as it allows needs to be predicted even when the product category is not in the ground truth dataset. Finally, the outcome of the user study shows that there is a high overlap between future customer needs predicted by participants from product development teams and the needs predicted by an algorithm in this thesis. The participants also noted that the algorithm output is novel and useful.

Chapter 1: Introduction

This chapter first introduces the overall context of the research described in this thesis. Based on this, a set of Research Questions (RQ) are proposed along with a list of corresponding Research Contributions (RC) that address them. Finally, an overview of the thesis's chapters is provided.

1.1 Research Context

The creation of new products has been defined as one of the "top goals" [1] as well as a "critical success factor" [2, 3] for Small and Medium-sized Enterprises (SMEs) [2] and large ones [3] trying to grow or survive. The process of generating new product ideas or features for products has been defined as the "product discovery" [4, 5] or the "new product development" phase [6, 7]. Businesses listen to the Voice of the Customer (VOC) to aid in the product development/discovery process [4, 8–12], as customer satisfaction is "vital" [6] for the success of new product ideas. Some research has even pointed out that customers on their own can provide better product ideas than professional developers, with [13] finding that customers are likely to produce more novel products that earn more sales and [9] finding that they provide better (albeit less feasible) product ideas.

The use of "user interviews" [14] is the most popular method for businesses to garner insights into their customers' needs and requirements. However, in recent years companies have also been turning to the use of statistical methods from Artificial Intelligence (AI), and in particular Machine Learning (ML), to gather requirements from User Generated Content (UGC) [15] such as product reviews, social media, etc. These methods have warranted exploration because user interviews have drawbacks that computational methods don't suffer from, namely: 1) they incur large monetary and time costs; 2) they suffer from small sample sizes; and 3) they have the potential for biased results due to participants already knowing what responses the company wants [16]. ML methods that are run over UGC are in such demand that they have become businesses within themselves, with companies such as Sprinklr, Blackswan, Meltwater and Sprout Social all engaging in the practice. The billion-dollar valuation of Sprinklr, giving the company a "unicorn" status, shows the interest of other businesses in the area.[1] Furthermore, businesses themselves use internal software to mine customer requirements. For example, the fast food chain McDonalds created an internal subdivision called the "McD Tech Labs" to better understand their customer's needs [17].[2]

[1] https://fortune.com/2015/03/31/sprinklr-unicorn-valuation/ - last accessed 07/06/2024

[2] https://www.nytimes.com/2019/10/22/business/mcdonalds-tech-artificial-intelligence-machine-learning-fast-html - last accessed 07/06/2024

Studies that extract the VOC from UGC have been coined by some as the customer needs mining literature [18–20]. A customer need has been defined in the marketing literature as a "description in the customer's own words of the benefit to be fulfilled by the product or service" [21] e.g. the need to *prevent chapping* for Vaseline lip balm.[3] Computational approaches analysing UGC have also included the features or attributes of a product in this definition [22, 23] as they themselves contain benefits e.g. the *coconut* flavour/scent which contains the need *fragrant* for Vaseline. Similarly, in this thesis, the definition of customer need contains both the benefiting descriptions of a product (e.g. *prevent chapping*, *fragrant*, etc.) as well as its features (e.g. *coconut*). The approaches in this thesis also assume a customer need is in the form of a single keyphrase, as it is the output generated by the algorithms presented in this thesis. Most studies in the area of customer needs mining perform text mining or Natural Language Processing (NLP) over UGC documents/posts. These approaches differ on various aspects including: 1) *data used*; 2) *methods performed*; 3) *application scenario*; and 4) *evaluation approach*.

The vast majority of studies use either product reviews [24–37] or social media [22, 38–49] to extract customer needs, although other data sources have also been used e.g. customer feedback data in warranty databases [50–52]. Studies consider many factors when choosing a data source including but not limited to: 1) accessibility - is the data easy to obtain through publicly accessible Application Programming Interfaces (APIs); 2) size - is the data big enough to represent a large number of consumer opinions; 3) relevant content - does the data discuss customer needs to be addressed in potential products; and 4) text limitations - is their textual restrictions which could constrain the computational methods performed (e.g. Twitters 280 character limit).

Methods used to mine customer needs can generally be split into three primary categories: 1) document classification; 2) document/keyphrase clustering; and 3) keyphrase ranking/classification. Document classification techniques are usually performed in a binary classification setting by predicting ones that are relevant to "customer needs", as per the definition in their respective studies [38, 53, 54]. The output of this method reduces the time spent by product developers by lowering the set of documents to only ones containing customer needs, however, it still leaves a lot of work to be done. These studies mainly use supervised classifiers from classical ML e.g. Support Vector Machine (SVM) [38, 42, 49]. However, more recently deep learning approaches have been used e.g. Convolutional Neural Networks (CNNs) [53], Long Short-Term Memory (LSTM) [55], etc. Clustering techniques work by either grouping documents or keyphrases into sets of related customer needs. Some effective studies using clustering can be seen as a type of hybrid method as they run a document classification model prior to clustering [53, 54], so to only cluster documents (or keyphrases from documents) containing customer needs. Document clustering approaches group similar documents together [26, 53, 56, 57] which removes the "winnowing" [53] process of having to read through documents containing the same customer need (as with document classification). Alternatively, keyphrase clustering approaches group similar keyphrases together [18, 58–60]. The output of keyphrase clustering largely depends on the text representation the algorithm is run over. For example, if the representation relates keyphrases by co-occurrence, such as a Bag-of-Words (BoW), keyphrases that occur fre-

[3] Where customer needs generally refer to requirements, demands, preferences, wants etc.

quently together are grouped [60]. However, if the representation relates keyphrases by their semantics, such as word embeddings, the output will be groups of keyphrases with similar meanings [58]. Keyphrase clustering provides product development teams with different insights into UGC data than document clustering, more directly useful for the detection of customer needs to form the basis of new products. For example, if coconut and strawberry are in the same cluster in a corpus of shampoo posts then it might be an idea to create a coconut-strawberry shampoo product. Both document and keyphrase clustering approaches use various algorithms such as hierarchical clustering [53, 60], Self-Organizing Map (SOM) [56], etc. Finally, keyphrase ranking/classification approaches work by analyzing a corpus of UGC and either ranking keyphrases from the documents based on how likely they are to be customer needs or classifying unique keyphrases as customer needs. Keyphrase ranking and classification are highly similar as the output of either one can be interchanged. For example, the output of ML-based keyphrase classification can be changed to a ranking of keyphrases due to how the likelihood output can be ordered, while keyphrase ranking can be seen as a type of classification technique by introducing a cut-off point in the ranking where keyphrases are classified as customer needs. The output of keyphrase ranking/classification is a highly useful method for product development teams, as it recommends them with customer need keyphrases that can form the basis of products. Approaches using these methods rank/classify using different techniques, considering various features when ranking e.g. frequency [61], sentiment [40, 41, 62, 63], etc.

There are various application scenarios for mining customer needs. Of particular note, is that most studies mine on the product model level (e.g. iPhone5) [40, 41, 45, 46, 63] rather than the product category level [53, 55] (e.g. mobile phones). When mining on the product model level, studies mine on a corpus of posts which discuss the model of interest e.g. [45, 46] extracts needs for Samsung Galaxy Note 5 by mining on the subreddit r/galaxynote5. Alternatively, studies mining on the product category level extract customer needs from a corpus of posts surrounding the general product of interest e.g. a corpus of posts all containing the term "dog food" when mining for Dog Food products. Studies also differ in the types of customer needs they detect with some mining general needs while other approaches mine for specific types of needs which may be of more business value e.g. unmet needs [16] or needs of future importance [35, 64, 65].

How studies in the customer needs mining literature are evaluated depends on the method being employed. For document classification, the evaluation approach is easy to define as it follows from previous classification literature, where general ML metrics (e.g. precision, recall and F1) are calculated [38, 42, 43, 49, 53–55, 66, 67]. However, for clustering and keyphrase ranking/classification tasks the evaluation is more difficult to define i.e. how do you determine whether a group of documents/keyphrases (clustering) or a single keyphrase (keyphrase ranking/classification) is a customer need? Due to this difficulty, many studies evaluate their algorithms through a manual examination [40, 41, 63] of their approach or a demonstration of it in their study [26, 31, 45–47, 56, 59, 60, 68]. However, ideally a ground truth specific for the evaluation of customer needs in the form of clusters or keyphrases is required for these approaches to give proper insight into how their algorithms perform.

This research described in this thesis investigates new algorithms and evaluation approaches for the extraction of customer need keyphrases. Specifically, it aims to develop and evaluate

algorithms capable of accurately predicting customer need keyphrases on Reddit which appear as the most important needs addressed in future products i.e. discover future customer needs.[4] It does this using the discussed family of methods in the keyphrase ranking/classification literature. Relevant to this thesis, a keyphrase surrounds the idea that it captures the main topics in a document or summarizes it best [69–72]. It is different from a keyword as it connotes a multiword lexeme [72]. A candidate keyphrase is a phrase an algorithm analyzes to predict whether it is a keyphrase. Many computational studies mention the term (i.e. candidate keyphrase) when referring to some of the initial sets of phrases that are first analyzed by an algorithm [69, 70, 72–74] e.g. by applying the Part-of-Speech (POS) filter 50 candidate keyphrases are removed. These two definitions (i.e. keyphrase and candidate keyphrase) are referenced throughout the thesis.

One of the main contributions of the work in this thesis is that it addresses the challenging task of predicting future customer needs. Identifying these types of needs allows companies to gain early access to new markets and increase their overall profitability [75]. Discovering these types of needs is of such importance to businesses developing products that companies such as Black Swan Data (a firm specializing in predicting future market needs) have developed into highly successful businesses, accumulating over £15.2M in investments in 2022 alone and attracting firms such as PepsiCo and McDonalds.[5] [6] [7] However, the current literature has noted the lack of methods addressing this task [64] which is unexpected given the value it provides businesses.

There are not many generic definitions of what a future customer need is in the literature. However, it is often mentioned alongside words like hidden [76] or unmet [77], hinting at the fact that they are sometimes undiscovered/unsolicited.[8] In this thesis, the definition is the same as the one used for a customer need i.e. a benefiting specification (e.g. *prevent chapping* for lip balm products) or a feature that has associated benefits (e.g. the ingredient *coconut* in lip balm). However, it has the additional trait that it is obtained at some future time period. This future time period is dependent on the method employed in various chapters of this thesis (e.g. 1-3 years).

There are two primary methods employed in this thesis for predicting future customer needs from Reddit. As these approaches are considered to be keyphrase ranking/classification methods, the output they generate is in the form of: 1) lists of keyphrases sorted by how likely they are to be future needs; or 2) binary indicators of whether a keyphrase is a future need. The first approach is a rule-based method (described in Chapter 4). It works by primarily removing posts that are not in subreddits relevant to discussing customer needs. Secondly, it removes candidate keyphrases that are not likely to be customer needs based on

[4] Due to the nature of the work in this thesis involving human subjects, it received an ethics waiver from the UCD Human Research Ethics Committee – Sciences (HREC-LS) under ref LS-E-20-81-Kilroy-Caton.

[5] https://www.crunchbase.com/organization/black-swan-data/company_financials - last accessed 07/06/2024

[6] https://www.blackswan.com/case-studies/serving-up-new-flavour-innovation-for-pepsico - last accessed 07/06/2024

[7] https://www.blackswan.com/case-studies/social-data-insights-mcdonalds-are-lovin-it - last accessed 07/06/2024

[8] The idea behind predicting future customer needs is that they are currently ones which are unmet (or not addressed enough) which is of high value to businesses

various criteria obtained from the customer needs mining literature (e.g. POS tag filtering, infrequent frequency, etc.). Finally, it ranks keyphrases by their rising frequency on Reddit and Google Trends as ones that are most likely to be future customer needs. The performance of the algorithm is evaluated at detecting future customer needs for Toothpaste products. In the absence of previous methods solving the task, it is shown that the approach can obtain significantly better performance than a simple baseline. It is also shown in a retrospective evaluation, that it can capture important customer needs identified by a large Multinational Corporation (MNC) with lead times of up to 25 months ahead of them trending in the marketplace. The second approach is an ML-based approach (detailed in Chapter 5). With the recent developments in Multivariate Time Series Classification (MTSC) [78], the method works by classifying candidate keyphrases using instances constructed of 1263 univariate time series. The 1263 univariate time series generated for each candidate keyphrase instance come from 10 families of features, all relevant for the task of predicting future customer needs e.g. sentiment-based, frequency-based, user-based, product-based, etc. During an evaluation of 15 different product categories (e.g. dog food, toothpaste, cereal, etc.), it is shown that the approach significantly outperforms the previously described approach in Chapter 4.

Another major contribution of the work in this thesis is the curation of a ground truth dataset possible for the evaluation of approaches predicting future customer needs. This is required as datasets like this are generally lacking in the customer needs mining literature (as discussed previously in this chapter). So to curate such a ground truth dataset, a list of the most heavily addressed keyphrase customer needs is extracted from a large database of new-to-market products. For each product category evaluated, a list of these gold standard keyphrase customer needs is extracted each month. The goal of the algorithms described in this thesis (which are run over Reddit), is to predict the top needs occurring at some defined future time period from the date of prediction. Mintel GNPD [79] is utilized as the source data to curate this ground truth dataset. GNPD is a large product information database used by industry and academics [79]. Of specific relevance to the work in this thesis is that for each record it records a textual product description detailing claims and features included in the product which in turn reflect customer needs [79]. Furthermore, each product added to the database is a newly registered one and timestamped (when it was first available for retail). These two aspects of GNPD allow lists of the most addressed customer needs each month in real products to be formulated. Two methodologies for the curation of the ground truth from Mintel GNPD are described in Chapter 3. The first method works by having annotators label a sample of product descriptions from GNPD which are used to train an NER model that detects needs over the entire database. These needs are tracked and a ranked list of the most popular ones each month makes up the ground truth. This method only curates a ground truth for 1 product category (i.e. Toothpaste) which is used to evaluate the rule-based approach (detailed in Chapter 4). The second method extracts keyphrases using text mining techniques which are then uniquely labelled as customer needs by annotators. Similarly, these needs are tracked and a ranked list of the most popular ones each month makes up the ground truth. This methodology is deemed to produce a more accurate ground truth output than the first approach, as it doesn't use a prediction model in its curation process. This method also curates a ground truth for 37 product categories across various categories of Consumer Packaged Goods (CPG) products (e.g. Toothpaste,

Cereal, Beer, Dog Food, Air Freshener, etc.). This dataset is used to train and evaluate the ML approach described in Chapter 5.

As detailed earlier in this chapter, most studies mine customer needs on the product model level (e.g. The Laughing Cow) rather than the product category level (e.g. Cheese). The approaches in this thesis mine on the product category level. Specifically, customer needs for 15 product categories in the area of CPG are mined for. A conventional thought process may assume that individual models are built for each product category having needs predicted for it e.g. train on future needs for Dog Food which predicts future needs for Dog Food. However, a model presented in Chapter 5 which incorporates MTL, is trained on multiple product categories (e.g. toothpaste, cereal, and beer) rather than one category (e.g. toothpaste).[9] This is a key contribution of the thesis and is significant because during evaluation it is shown that this model can still accurately predict for a category it doesn't use during training e.g. can be trained on Toothpaste, Cereal and Beer yet still predict for Cookies. Therefore, if no training data is available for a certain product category, future needs for it can still be predicted. It is assumed that this model works as it learns what a general future customer need keyphrase instance looks like on Reddit (over multiple product categories) rather than what it looks like for a single category. This is made possible as the univariate time series features are not generated with the mining of any particular product category in mind e.g. features which are particularly helpful in the prediction of Dog Food future customer needs. This time series features generated are instead category agnostic, helpful for the prediction of any product category e.g. frequency-based signals, sentiment-based signals, etc.

Along with the lack of studies using ground truth data to mine customer needs, only a very small number of studies involve humans in the evaluation process to either assess the output produced by an algorithm and/or compare it to their generated output. These studies involve the use of participants working in marketing companies or firms developing products who have the expertise to know whether a certain customer need is useful i.e. through user interviews or other mediums. There is a lack of studies involving these participants [53] as they work in firms that don't want to disclose important information that their competitors can use. A contribution of this thesis is that it carries out a user study involving a large MNC to evaluate whether the needs predicted by the ML model can potentially be addressed in their new product lines. The firm involved in the study, as of 2023, is estimated in the billions (USD) and has over 500 products in the marketplace. Along with the user study, an approach to having firms evaluate customer needs without having to disclose any important information is detailed.

1.2 Research Questions

The following RQs are addressed in this thesis:

[9] MTL has been described as having the aims of improving the learning of a model for a task by using the knowledge contained in a number of other learning tasks where these other tasks are related but not identical to the initial learning task [80].

RQ 1: Knowing which customer needs will be addressed in the market in the future is of obvious benefit to companies as it allows them to gain early access into markets and increases their overall profitability. **To what extent can future customer needs be detected with performance useful for the purposes of product development?**

RQ 2: There is a lack of datasets in the literature suitable for the training/evaluation of algorithms mining customer needs. Without them, a proper evaluation or use of supervised ML cannot be carried out. **How can a ground truth dataset be curated to allow for the training and evaluation of approaches that predict future customer need keyphrases?**

RQ 3: A model that can predict future customer needs for a product category it isn't trained for would be useful as ground truth data for a product category would not be required for it to be predicted e.g. train future customer needs in Dog Food, Toothpaste and Cereal products to predict future Eyeliner customer needs. The use of MTL has been used to improve the learning of a model for a task by using the knowledge contained in a number of other learning tasks not identical to the initial learning task. **How can MTL be used to train a generalizable model for which future customer needs can be predicted for a product category the model has seen and not seen during training?**

RQ 4: There is a lack of studies involving participants from marketing companies or firms developing products during the evaluation process to discover whether a predicted customer need is of use to a business. The work in this thesis involves the use of a large MNC which as of 2023 is valued in the billions (USD) and has over 500 products in the marketplace. **Do professional product developers from a large MNC believe the future customer needs generated from an algorithm developed in this thesis could potentially be ones addressed in their new product lines?**

1.3 Research Contributions

The main RCs to the literature described in this thesis are as follows:

RC 1: Lists of keyphrases representing future customer needs are predicted on Reddit using a rule-based approach (detailed in Chapter 4). This approach works by: 1) removing posts that are not in subreddits likely to be customer needs; 2) removing candidate keyphrases not likely to be customer needs; and 3) ranking candidate keyphrases by their rising frequency on Reddit and Google Trends which generate lists of customer needs which are predicted to be of future importance. When evaluating the approach at predicting future customer needs in Toothpaste products, it is shown that it obtains significantly better performance than a simple baseline (used in the absence of previous methods solving the task). In a retrospective evaluation, it is also shown that the

algorithm can detect important customer needs identified by a large MNC with lead times up to 25 months ahead of them trending in the marketplace. This contribution partially addresses RQ 1.

RC 2: Keyphrases are classified as customer needs on Reddit using an ML-based approach (detailed in Chapter 5). This approach uses techniques from MTSC to classify candidate keyphrase instances represented by 1263 univariate time series features. These 1263 features come from 10 families of features, all useful for the task of predicting future customer needs. The experiments, which are run over 15 product categories, show that this approach performs significantly better than the rule-based approach initially proposed in Chapter 4. This contribution partially addresses RQ 1.

RC 3: Two ground truth datasets necessary for the training and evaluation of algorithms predicting future customer needs are proposed (detailed in Chapter 3). To curate such datasets a ranked list of the most heavily addressed customer needs each month is extracted from a large database of new-to-market products i.e. Mintel GNPD. The first dataset is curated by having annotators label a sample of product descriptions from Mintel which are used to train an NER model that detects needs over the entire database. The second method uses text mining techniques to extract keyphrases that are uniquely annotated by humans. The first curation method is used to formulate a ground truth dataset for 1 product category (i.e. Toothpaste) while the second method curates for 37 product categories within the area of CPG e.g. Toothpaste, Cereal, Dog Food, Beer, etc. This contribution addresses RQ 2.

RC 4: A model that incorporates MTL by being trained on multiple product categories to learn what a general future customer need looks like is presented in Chapter 5. This is significant because during experiments it is shown that this model can accurately predict for a category it doesn't use in the training process e.g. can be trained on Toothpaste, Eyeliner and Shampoo yet still predict future needs for Dog Food products. Therefore, if no ground truth training data is available for a category, needs can still be predicted for it. This contribution addresses RQ 3.

RC 5: A user study involving participants from a large MNC to discover whether the needs predicted by a model described in this thesis could potentially be addressed in their new product lines is presented (Chapter 6). As of 2023, the MNC in question is estimated in the billion (USD) and has over 500 products on the marketplace. Furthermore, a useful framework for future studies to use is presented by having firms evaluate needs mined by algorithms without having to disclose any important information to researchers.

These contributions along with the research question they address and the publications that support them are summarized in Table 1.1.

Table 1.1. Contribution Summary

Research Contribution	Research Question	Chapter	Publication
RC 1 (rule-based)	RQ 1	Chapter 4	[81]
RC 2 (ML-based)	RQ 1	Chapter 5	[82]
RC 3 (ground truth)	RQ 2	Chapter 3	[81, 83]
RC 4 (MTL)	RQ 3	Chapter 5	[82]
RC 5 (user-study)	RQ 4	Chapter 6	[84]

1.4 Thesis Overview

The remainder of this thesis is structured as follows:

Chapter 2: An overview of the related work in the area of customer needs mining is provided. Additionally, a brief review of how product development teams identify customer needs and MTSC (an ML technique used in a critical part of the thesis) is detailed.

Chapter 3: A review of the ground truth and Reddit data used to evaluate/predict future customer needs is given. When reviewing the ground truth an overview of the two methods used to construct it is provided. Furthermore, a thorough review of Mintel GNPD (product information data used to construct the ground truth) is given. When reviewing Reddit, a brief overview of the data collected specific to the task addressed in this thesis is given.

Chapter 4: A detailed description of the rule-based approach of predicting future customer needs in lists of keyphrases is provided. An evaluation is provided showing its effective use in comparison to a simple baseline.

Chapter 5: Details of the ML-based approach of training an MTSC model to predict future customer need keyphrases are given. During experimentation, it is shown that this approach can significantly outperform the approach described in Chapter 4. It is also shown that a model that incorporates MTL can predict categories that it doesn't use in the training process.

Chapter 6: A user study involving participants from a large MNC is carried out to observe whether they think the future customer needs predicted by the model described in Chapter 5 could be integrated into their current product lines.

Chapter 7: A conclusion of the thesis is provided along with directions for future work.

Chapter 2: Related Work

This chapter summarizes the literature in the areas of product development, machine learning and customer needs mining. Firstly, a review of how firms identify customer needs is provided which is necessary given that this research is intended for product development teams (Section 2.1). A review of factors in which studies mining customer needs can be categorized is then given: 1) Data Used (Section 2.2); 2) Methods Performed (Section 2.3); 3) Application Scenario of Customer Need Mining Tasks (Section 2.4); and 4) Evaluation Approaches (Section 2.5). After, a review of the ML literature in MTSC is discussed as it is an important method used in this thesis (Section 2.6). Finally, a summary and discussion of the chapter is provided (Section 2.7).

2.1 How Firms Identify and Weight Customer Needs

In this section, a brief overview of the process companies developing products use to generate product ideas is provided. This is done as the algorithms created in this thesis are with firms developing products in mind, hence a review of how they come up with ideas and evaluate them is required.

2.1.1 Identifying Customer Needs

The product development process consists of eight stages [85]: idea generation, screening, concept development and testing, marketing strategy, business analysis, product development, market testing and commercialization. The first two stages are often referred to as the Fuzzy Front End (FFE) of innovation where firms identify opportunities [86, 87]. Here firms carry out: 1) idea generation - the creative process of coming up with a large number of new ideas [88]; and 2) idea screening - selecting from a large list of ideas the ones that warrant an extensive and expensive analysis [89].

During idea generation, organizations can avail of ideas both internally and externally [90]. Internal ideas mainly consist of employees within the organization or existing in-house market/research studies and surveys [88]. On the other hand, external ideas mainly come from customers (e.g. customer co-creation [91–94]), however, can also come from data banks, universities, public reports, competitors, etc. [88]. Some methods for getting people within the organization to come up with ideas consist of brainstorming [95], Substitute-Combine-Adapt, Modify-Purpose-Eliminate-Reverse (SCAMPER) [96], analogical thinking

[97], Teorija Rezhenija Izobretatelskih Zadach (TRIZ) [98], etc.[1] Traditional approaches for generating ideas externally mainly consist of focus groups and customer/user interviews [99–102], although other approaches have been employed e.g. analyzing patents [103]. However, recently modern approaches rely on algorithms that extract ideas from UGC. Idea screening should be performed in a multistage process by culling ideas at decreasingly less brutal screens so that high-quality ideas can be subject to thorough investigations at later stages [88, 104]. The most difficult part of this process is coming up with a formal criterion for removal, which is required in addition to intuition when providing explanations as to why an idea will/won't be productized [104]. Barring the initial criteria (e.g. can this product be made), all subsequent criteria can be quite extensive [88]. As pointed out in [88], stereotypical criteria may take the following into account: a) market (size, growth, appeal); b) product (uniqueness, exclusivity); c) feasibility (product development, technology, production, personnel, financial); d) compatibility (organizational infrastructure, personnel and managerial expertise); e) time (needed to develop the idea, needed to commercialize); f) financial (investment requirements, costs, profitability); and g) other (intuition, probability of success). There is no defined way to carry out the sequential procedure of idea generation and screening, with organizations still shown to have success regardless of their strategy [88] e.g. concerning variables such as idea sources, generation methods, screening methods, etc.

The algorithms that are presented in this thesis can be seen as a type of idea generation method, assisting product development teams in coming up with new ideas. They are developed in this thesis due to the drawbacks that traditional approaches suffer from including large monetary costs and small sample sizes [105]. It is the assumption, however, that a firm would carry out a screening process on the ideas generated by the algorithms discussed in this thesis. This is because idea screening can be a highly complex and firm-specific process [88] - not useful for the techniques used in text mining and ML.

2.1.2 Weighting Customer Needs

Although not specifically used in the FFE stages of product development (i.e. idea generation and screening), there is an array of frameworks used to weigh the importance of already identified customer needs. Although these techniques are not highly relevant to the work in this thesis, they are prevalent in the general business product development literature and more recently in the customer needs mining literature (with approaches implementing them), hence their review. Some of the most popular frameworks include: 1) the Kano model; 2) Kansei engineering; 3) Quality Function Deployment (QFD); and 4) Analytic Hierarchy Process (AHP). These frameworks weigh needs in products to the extent to which they satisfy customers.

The Kano model [106] is a product development framework to weigh how much needs satisfy/dissatisfied customers. It categorizes user satisfaction with an addressed customer need into 3 major attributes [107]: 1) Basic - attributes that if not present leave customers

[1] TRIZ is a organized and systematic approach to problem solving which comes from the Russian phrase meaning the "theory of inventive problem solving" [98].

extremely dissatisfied, however, if present bring no additional satisfaction (e.g. wheels on a car); 2) Performance - attributes that produce both satisfaction and dissatisfaction depending on performance levels (e.g. petrol consumption of a car); and 3) Excitement - attributes which produce potentially very high levels of satisfaction if delivered but no dissatisfaction if not (e.g. self-driving software in a car). There's a large number of studies in the business literature using this model to prioritize customer needs [108–110]. These studies span multiple domains including wine [110] products and airline [108] services.

Kansei (Japanese for "affective") engineering has been defined as "translating technology of a consumer's feeling and image for a product into design elements" [111]. The basis of Kansei engineering is that in terms of purchasing behaviour, consumers increasingly base their decisions on the impression a product leaves on them rather than the specific technical benefits it contains [112]. A popular example of this is how the company Mazda designed its cars based on how it made its users feel [113], even going as far as to base 6 commercials in the 1990s around the tagline "It just feels right".[2] Another example includes the various aspects the Apple iPhone makes in order to increase the impression left on its users [114] e.g. usability. Previous business studies have implemented this methodology through the use of questionnaires, in which respondents rate their feelings towards a customer need on a point scale between two opposite pairs of words (one positive and one negative) known as Kansei attributes [115, 116]. An example of two opposite words may include "soft" and "hard" (e.g. for the casing on a phone).

QFD has been described as "a method for developing a design quality aimed at satisfying the customer and then translating the customer's demand into design targets and major quality assurance points to be used throughout the production phase" [117]. In QFD, the main design tool called the "House of Quality" is built which identifies and classifies customer needs (What's), identifies the importance of those needs, identifies engineering characteristics that may be relevant to those needs (How), correlates the two, allows for verification of those correlations, and then assigns objectives and priorities for the system requirements [118].

Finally, AHP is an approach to quantifying the weights of decision criteria [119]. It has various applications in group decision-making as well as product development when weighting customer needs which are to be most addressed to the extent to which they satisfy customers [120].

2.2 Data Used

This thesis focuses on mining data sources where customer opinions are contained (e.g. social media and product reviews). Additionally, for balance, other sources are also reviewed (e.g. patents). The data sources reviewed include: 1) patents; 2) warranty databases; 3) product descriptions; 4) google trends; 5) product reviews; and 6) social media. Specifically, the mentioned data sources are discussed in terms of: 1) accessibility; 2) represents the VOC

[2] https://www.youtube.com/watch?v=vDflgEhkYdM - last accessed 07/06/2024

i.e. contains real customer opinions; 3) audience/data size; and 4) usefulness for mining customer needs. It's of note that the ground truth data used to train the ML algorithms is not discussed in this section (discussed in the evaluation section of this chapter i.e. Section 2.5).

2.2.1 Patents

Many studies mine publicly available patents to obtain new product ideas and innovations [103, 121–128]. Patent data can be accessed easily through several large patent databases, with various studies extracting innovations from the United States Patent and Trademark Office (USPTO) [123, 127] or the European Patent Office (EPO) [128]. Approaches which mine patents are used to find innovations across many product domains e.g. automobiles [122] and telehealth [123]. A useful advantage of mining patents is that they are rich in highly relevant information discussing new innovative ideas [103]. However, they are not relevant to the research in this thesis which aims at listening to the VOC i.e. customer's thoughts and opinions on products. They are also quite small in size compared to the other data sources discussed in this section e.g. product reviews and social media. Another obvious drawback of patent mining includes the risk of copying a firm's idea [129], leading to potential legal issues.

2.2.2 Warranty Databases

Warranty claims as well as other forms of customer feedback are commonly stored as text in warranty databases.[3] Warranty claims have been classified into 4 families of failures [130]: 1) hardware failures; 2) software failures; 3) human errors (e.g. product usability issues); and 4) organisational errors (e.g. customer care having poor product knowledge). Studies use this information (i.e. hardware/software/human errors) to enhance their products, with many approaches being applied to the automotive industry [50–52, 56]. Information contained in warranty databases does not apply to the work in this thesis, as although it discusses drawbacks of a product it is less likely to discuss customer needs to be included in new products e.g. [50] used warranty databases to mainly find product failures rather than new customer needs. They are also very difficult to obtain as only companies selling products have this type of information.

2.2.3 Product Descriptions

Descriptions/specifications of on-the-market products are also a source of innovative ideas. Product descriptions contain highly informative content as they discuss benefits and features from competitors in real products. Tagging product attributes from these descriptions is a thoroughly explored area of research [131–134] e.g. tagging "cotton" for a vest clothing

[3] A warranty claim is a claim by a customer/product owner for a product under warranty, and usually entails either repair or replacement. - https://www.apqc.org/what-we-do/benchmarking/open-standards-benchmarking/measures/warranty-claims-rate - last accessed 07/06/2024

item description. The tagging of these attributes can be used for downstream tasks that are related to the research in this thesis i.e. finding innovative attributes from competitors' products that could disrupt the market. However, like patents, they are not applicable to the research in this thesis which aims to extract needs from customer's opinions i.e. UGC. Although there are databases of product descriptions (e.g. Mintel GNPD [79]), they are only accessible through corporate API endpoints.

Product descriptions have also been used as a source of evaluation data, with approaches correlating the needs extracted from sources of UGC to attributes from descriptions [135]. In this thesis, product descriptions are used extensively as a source of ground truth evaluation data to discover whether customer needs on Reddit can be mapped to future needs in real products (discussed extensively in Chapter 3). Specifically, Mintel GNPD [79] is used for this purpose.

2.2.4 Google Trends

The predictive power of Google Trends has been used before in many business intelligence studies, most notably in sales prediction [136, 137]. More recently, however, it has been used to estimate the importance of future customer needs, with [65] showing that it can be a predictive data source when applied in conjunction with product reviews. Google Trends has the distinct advantage of representing a massive number of people which can be useful for the prediction problem addressed in this research i.e. predicting future customer needs. It also has the distinct advantage for use in product development by providing a filter for a type of category that can be used to extract needs for a particular type of product (e.g. Oral & Dental Care, Small Kitchen Appliances, etc.).[4] It can also be applied to a specific location(s), that can be used to filter needs for certain markets.

Google trends data can be accessed through publicly accessible APIs.[5] In Chapter 4, it's used in conjunction with Reddit to rank customer needs by the likelihood to which they will be addressed in products. During an evaluation, it was found that this method was found to be a better indicator than ranking by its occurrence on Reddit alone, most likely because Google Trends makes up a very large user base.

2.2.5 Product Reviews

Product reviews are used in many text mining and NLP tasks. Such include predicting review helpfulness [138, 139], summarization [140, 141], sentiment analysis [141–143], etc. They are also used extensively when mining for customer needs [24–37]. Amazon is the main review site used to extract customer needs [24–27, 30–32], however, studies also use data from other sites such as CNET [35, 36], Best Buy [30], eBay [30, 37], Trustpilot [33], Google Play Store [32], etc. Product review data has been compared to social media data

[4] https://github.com/pat310/google-trends-api/wiki/Google-Trends-Categories - last accessed 07/06/2024

[5] https://pypi.org/project/pytrends/ - last accessed 07/06/2024

for its use as a data source suitable for the prediction of customer needs. Specifically, [40] has pointed to the advantages social media has over product reviews including:

- Accessibility: Although social media data is more of a challenge to access in recent times (e.g. tightening of Twitter and Reddit's APIs, [144]), up-to-date product reviews necessary for the mining of new customer needs are very difficult to obtain [40].
- Size: The volume of data and the size of the audience on social media is much larger than on product review sites.
- Immediacy: Web content (such as product reviews) takes longer to publish as it requires verification and proofreading [145]. In comparison, social media data is published instantly.
- Heterogeneity: There is a wide variety of opinions on social media [146], which results in its users discussing thoughts on products that are seen outside review sites [40] e.g. social media users discussing do-it-yourself solutions to beauty products before they've become popularized in the market [147].

One of the main drawbacks of using social media in comparison to product reviews is that it contains more irrelevant content when mining for customer needs e.g. users discussing the product "mobile phone" for reasons other than needs they desire for it - "I texted them back on my mobile phone". Product review data suffers this same issue, however, not at the same scale e.g. users critiquing the delivery time of their product on Amazon [40]. Similarly, it is also worth noting that product reviews also suffer from a lot of spam content [148, 149] e.g. fake reviews.

2.2.6 Social Media

Social media (alongside product reviews) is the most used data source for mining customer needs.[6] The main platforms used to extract needs include: Twitter [22, 38–43], Facebook [44], Reddit [45–47], Instagram [48] and Quora [49]. Other platforms that are not used specifically for the mining of customer needs, however, may have the potential to due to other text mining studies being analyzed on them include: Pinterest [151], Tumblr [152], Discord [153], Telegram [154], Tiktok [155], Sina Weibo [156], WeChat [157], QZone [158] etc.[7] The following factors were considered when selecting the data source to mine: 1) Accessibility; 2) Text limitations; 3) User base; 4) Assistive for feature engineering; and 5) Large number of historical posts.

In terms of **accessibility**, there are only a few open opinion-based API platforms available since the "Post-API Age" [159] of platforms like Twitter and Facebook restricting access

[6] Blogs (e.g. Quora), microblogs (e.g. Twitter) and general social networking sites (e.g. Facebook) have been included in past definitions of social media [150], and hence are included for this review.

[7] For the video/picture social networks mentioned here (i.e. Instagram, Pinterest, Tiktok), the textual video/picture description or the comment data is mined on

following major data scandals (e.g. Facebook Cambridge Analytica [160]) or changes in company policy (e.g. Twitter restricting access following the takeover by Elon Musk [161]).[8]

In terms of **text limitations**, three high-level questions are considered: 1) Is the platform primarily text-based?; 2) Are there any limitations of the text data?; and 3) Is there a large number of English posts? A platform that is fundamentally text-based is preferred (e.g. not Instagram or TikTok), as this thesis focuses on language-based approaches. A platform that does not impose any limitations on the text is also preferred such as Twitter's 280-character limit. As the ground truth data in this thesis is English-based and the language spoken by all the participants in a performed user study is English, a corpus of English posts is preferred.

Although the work in this thesis is not run on a massive number of posts, a platform with a **user base** containing a high number of Monthly Active Users (MAUs) is still favoured. Platforms such as Facebook (3.05 billion), Twitter (550 million), Reddit (430 million), WeChat (1.32 billion) and Instagram (2.04 billion) boast a massive number of MAUs.[9]

A platform that is **assistive for feature generation** is also preferred as it better enables the task of predicting future customer needs to be learned. The following two criteria are considered when looking for platforms that are useful in this sense: 1) having unique qualities that allow for feature generation; and 2) having an API that doesn't restrict access to a lot of information (e.g. user gender/age, post date, etc.). For feature generation, a platform that has unique qualities in this respect includes Reddit with its subreddit structure (a stated benefit in other studies [45–47]), as well as a large number of other unique post-level attributes which other platforms don't have e.g. stickied, pinned, locked [162]. Platforms can also vary in the amount of useful information they provide which can aid in the task of predicting future customer needs. For example, Reddit restricts access to personal user information (e.g. gender, age, number of friends, etc.) or friendship/follower information which could be used to form a network graph.[10] [11] This therefore makes it more difficult to generate features. On the other hand, Twitter doesn't restrict a lot of critical information e.g. user information such as gender, age, geolocation, follower/followee information, etc.[12]

Due to the work in this thesis predicting future customer needs from past social media data, the platform used requires a **large number of historical posts** as far back in time as possible to train/evaluate algorithms while also reducing experimental bias. This renders sites like TikTok (released in 2016 [163]) infeasible to use.

[8] https://twitter.com/TwitterDev/status/1641222782594990080 - last accessed 07/06/2024

[9] https://buffer.com/library/social-media-sites/ - last accessed 07/06/2024

[10] Pushshift API attributes - https://github.com/pushshift/api#list-of-endpoints - last accessed 07/06/2024

[11] General Reddit API attributes - https://praw.readthedocs.io/en/stable/code_overview/models/redditor.html - last accessed 07/06/2024

[12] https://developer.twitter.com/en/docs/twitter-api/v1/data-dictionary/overview - last accessed 07/06/2024

[13] https://developer.twitter.com/en/docs/twitter-api/v1/data-dictionary/overview - last accessed 07/06/2024

[14] https://developers.facebook.com/docs/graph-api/reference/ - last accessed 07/06/2024

[15] https://github.com/pushshift/api - last accessed 07/06/2024

[16] https://developers.facebook.com/docs/instagram-api/reference - last accessed 07/06/2024

Table 2.1. Summary of the social media platforms considered for the work in this thesis

Platform	Accessibility	Text Limitations	User Base	Feature Generation	Historical Posts
Twitter	Restricted	280 character limit	550 million MAUs	Yes[13]	Yes
Facebook	Highly Restricted	No	3.05 billion MAUs	Yes[14]	Yes
Reddit	Restricted	No	430 million MAUs	Yes[15]	Yes
Instagram	Highly Restricted	not text-based platform	2.04 billion MAUs	Yes[16]	Yes
Quora	No Official API	No	300 million MAUs	n/a	Yes

All information for the "User Base (2023)" column is retrieved from
https://buffer.com/library/social-media-sites/

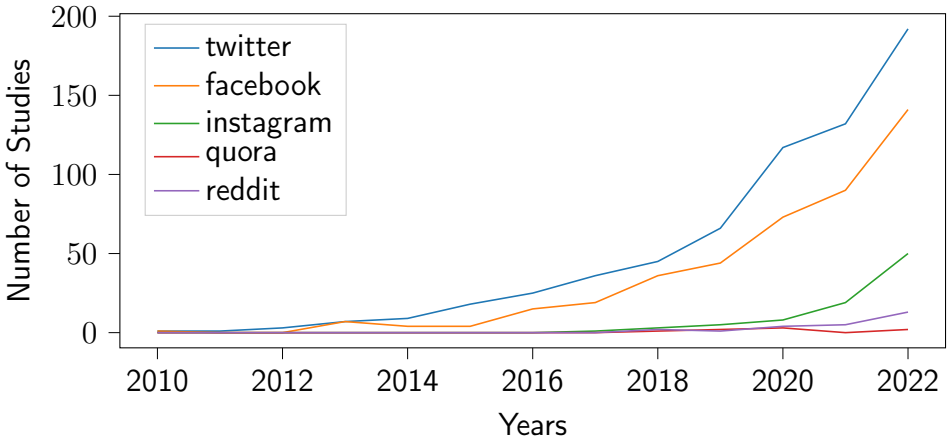

Figure 2.1. Indication of the number of papers published in the customer needs mining literature using the respective social media platforms. See Appendix A for a detailed description of how this graph is generated.

Table 2.1 summarizes the main social media platforms that were considered regarding the mentioned factors. Twitter and Reddit stand out as the platforms most suitable due to their less restricted accessibility (compared to other platforms), small number of text limitations, large user bases, potential for feature engineering and availability of historical posts. A drawback of Twitter, however, is that it has already been used in a lot of customer needs mining studies. This is shown in Figure 2.1 which shows an indication of the number of papers that have used the mentioned platforms in Table 2.1 (see Appendix A for a detailed description of how this graph is generated). Reddit on the other hand has only been used in a small number of studies, posing new research problems and challenges.

2.2.7 Summary and Discussion

At the start of this section (i.e. Section 2.2), it was mentioned that a review of the major data source candidates would be given in terms of: 1) accessibility; 2) whether they

represent the VOC; 3) audience/data size; and 4) usefulness for mining customer needs. Table 2.2 summarizes the discussed data sources in terms of the mentioned factors. Social media is picked due to it being a data source that is: 1) accessible (at the time of collection); 2) representative of real customer opinions; 3) large in terms of audience and size; and 4) suitable for the mining of customer needs.

Table 2.2. Comparison of data sources used to mine for customer needs

Data Source	Accessibility	Represent VOC	Size	Mining Customer Needs
Patents	Yes	No	Large	Yes
Warranty Databases	Highly Restricted	No	Dependent on Company	Yes
Product Descriptions	Highly Restricted	No	Large	Yes
Google Trends	Highly Restricted	Yes	Represents Large Audience	Yes
Product Reviews	Restricted	Yes	Large	Yes
Social Media	Restricted	Yes	Large	Yes

Specifically, the social media platform Reddit is chosen because it is a data source that has not been heavily researched in the literature along with the mentioned factors when considering a social media data source to mine on (as discussed in Section 2.2.6 i.e. 1) Accessibility; 2) Text limitations; 3) User Base; 4) Assistive for feature generation; and 5) Large number of historical posts. In terms of **accessibility**, at the time of data collection, Reddit was chosen due to it being one of the few open opinion-based API platforms. At the time of writing, Reddit has since restricted access to its APIs [144]. However, given its ease of retrieval at the time of data collection, its benefits in terms of accessibility are acknowledged.[17] Specifically, when obtaining data the work in this thesis uses the Pushshift API which has a limit "five times greater" [164] than the official Reddit API further facilitating the data acquisition process. Reddit also meets all the stated criteria for being a platform with little to no **text limitations** i.e. is a mainly English-written text-based platform where posts are generally longer than other platforms [45–47]. Even though not as large as other platforms, Reddit also has a large **user base** i.e. 430 million (as stated in Section 2.2.6). Such a number is sufficient for the experiments in this thesis. Another advantage of Reddit is that it has unique qualities useful for **feature generation** such as its subreddit structure and other unique post-level attributes (discussed in Section 2.2.6). Finally, Reddit has a **large number of historical posts**, with 46 and 90 million MAUs in 2012 and 2013 respectively. Its historical Pushshift API also allows past posts to be accessed easily [164].

[17] At present it is unclear how Reddit data will be accessed in the future. Currently, it seems that a user licence (similar to that of Twitter) will be put in place for academics - https://techcrunch.com/2023/04/18/reddit-will-begin-charging-for-access-to-its-api - last accessed 07/06/2024

2.3 Methods to Extract Customer Needs

Previous approaches in the literature which mine customer needs from UGC (e.g. social media, product reviews, etc.) mainly follow the same general process. As discussed, the data is first collected, which may relate to an exact product [29, 45, 46] (e.g. iPhone 5) or a group of products [16, 47] (e.g. cell phones). Secondly, an amount of preprocessing of the data is carried out. Such preprocessing usually consists of tokenization, normalization (e.g. stemming or lemmatization [16, 26, 40]) and removal of uninformative words from the documents themselves (e.g. stop words [47, 54]). Finally, an analysis is carried out using methods that fit the task to be solved. The three main methods used in the customer needs mining literature will be discussed in this section.

Figure 2.2 shows an example output (applied to the domain of Toothpaste) of the 3 main methods used to mine customer needs from textual data sources: 1) Document Classification; 2) Clustering; and 3) Keyphrase Classification/Ranking. Document classification methods work by categorizing documents into ones which contain a need or not e.g. "this coconut toothpaste makes my teeth feel fresh" (i.e. contains a customer need) or "my shampoo was in my bag with my toothpaste" (i.e. does not contain a customer need). Clustering methods work by grouping documents or keyphrases into clusters representing families of customer needs. Finally, keyphrase ranking/classification methods work by extracting keyphrases from a collection of documents which are then uniquely classified as customer needs or ranked by how likely they are to be customer needs e.g. Figure 2.2 shows an output of ranking keyphrases which states the 3 most likely keyphrases to be customer needs in the Toothpaste domain are coconut, charcoal and enzymes. The output of the following methods are reviewed in terms of three criteria, which are critical for choosing the best method that is fit for the task of automatically predicting future customer needs for product development teams (i.e. the purpose of this research): 1) usefulness to teams developing products; 2) ease of evaluation; 3) ability for future prediction.[18]

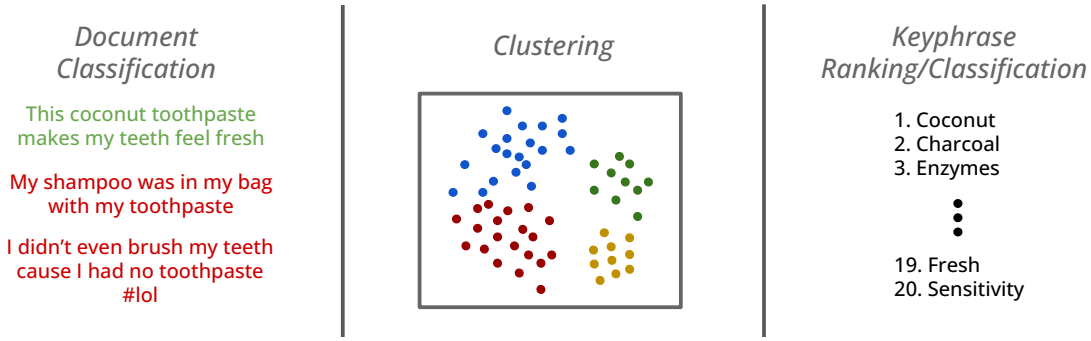

Figure 2.2. Output of the main methods used to mine customer needs from textual data sources (Toothpaste needs example): 1) Document Classification; 2) Clustering; and 3) Keyphrase Ranking/Classification

[18] Only a high-level review of the methods in terms of their evaluation is given in this section. Section 2.5 provides a more in depth examination.

2.3.1 Document Classification

Document classification methods reduce the number of documents under analysis to ones that are "informative" from the standpoint of mining for customer needs. This definition of informative changes depending on the study at hand. Some definitions for this include whether the document contains: 1) a product "wish"; 2) purchasing intent; 3) customer needs; or 4) product innovations.

In the general document classification literature, a wish has been described as a "desire or hope for something to happen" [165]. Work in [165] detects wishful from non-wishful documents from data obtained from the Times Square New Year's Eve "ball drop" where people offered their wishes be printed on confetti and dropped from the sky. Later work in [66] focused on wish mining from a commercial perspective by finding suggestions of "buy" wishes from product reviews. Similarly, to work on mining "buy" wishes, studies mining "purchasing intent" [42, 43, 49, 67] identify documents showing "a desire to purchase a product or service in the future" [49]. Purchase intent studies mine on various UGC data sources such as Quora [49], Yahoo Answers [49] and Twitter [42, 43]. Other studies specifically classify documents based on whether they contain "customer needs" [38, 53, 54] from the definition in their respective studies. [53] gives examples of what customer needs are when obtaining labeled data for their classification task e.g. they state that the sentence "this product can make your teeth super-sensitive" is a need as it is informative from the sense of providing information about the product, however, "this product can be found at CVS" is uninformative as it only mentions the store it can be purchased in.[19] In other works, studies have tried to identify documents that contain "product innovations" [55]. Specifically, [55] created a manually labelled dataset from 10,000 Amazon product reviews which are annotated based on being an innovation from the definitions in the Merriam-Webster dictionary [166] and the Business Dictionary [167]. It is of note that the mentioned models usually predict a binary output label (e.g. is/isn't a customer need). However, they have also been used in multi-class classification settings, for example, [43] additionally classifies the product category of documents containing buying intent e.g. Food, Drink, etc.

The methods used to solve document classification tasks range from classical ML approaches [38, 42, 49] (e.g. decision tree) to more recent deep learning approaches [53, 55] (e.g. neural networks).[20] For the classical ML approaches, the text representation used usually consists of a BoW [38, 42], although other similar representations are also used e.g. Delta Term Frequency - Inverse Document Frequency (TF-IDF) [49].[21] Instead of single tokens (as in a BoW model), multi-word tokens are also commonly included [42, 49] (e.g. "red candy"). Features other than token-based attributes are also frequently used e.g. dependency grammar [49], POS tags [42, 49], sentiment [49], etc. After feature engineering, classical ML algorithms are then run over these simple representations. The

[19] Consumer Value Store (CVS) is a retail pharmacy in the United States of America - https://en.wikipedia.org/wiki/CVS_Pharmacy - last accessed 07/06/2024

[20] Classical ML refers to algorithms which don't perform feature engineering as part of the learning process (as in deep learning) e.g. decision tree, naïve bayes etc.

[21] Similar to TF-IDF, Deta TF-IDF is a method used to weight terms which has been shown to improve text classification performance [168]

algorithms employed include SVM [38, 42, 49], Nearest Neighbour classifiers [42] (e.g. K-Nearest Neighbors), Bayesian classifiers [38, 42] (e.g. Naïve Bayes), Tree-based classifiers [38] (e.g. Random Forests), etc. In comparison to the classical ML approach, deep learning approaches aim to learn the classification task with a single model. As a first step however, a lot of algorithms use word embeddings for text representation instead of a sequence of tokens [53, 55] e.g. [55] combines 3 commonly used pre-trained word embeddings (i.e. BERT [169], XLNet [170] and GloVe [171]) to represent each document. As with classical ML, various algorithms are used to learn the task e.g. CNN [53] and LSTM [55].

By reducing the number of documents under analysis to a more distilled set, document classification can be very useful to teams who already read social media posts for the purposes of product development. However, for teams not performing this task, this method still leaves a lot of work to be carried out as they still need to "winnow" [53] through documents that may contain customer needs already stated in previous ones e.g. "brushing makes my teeth sensitive" and "my teeth are sensitive after brushing" both discuss the same need of sensitive teeth. In comparison, other methods provide more automation in the process i.e. clustering and keyphrase ranking/classification. That being said, document classification is the easiest of the discussed methods to evaluate as the ground truth label is an indicator extracted from a relatively straightforward annotation process i.e. manually labelling a sample of documents into classes. In terms of predicting future customer needs using document classification techniques, the annotation guidelines could be changed to only label ones containing future customer needs, hence there's no reason why it'd be difficult to perform.

2.3.2 Clustering

Clustering approaches work by grouping documents or keyphrases into families of customer needs. They can be split into three families of approaches: 1) document clustering; 2) keyphrase/term clustering; and 3) topic modelling.

Document clustering refers to methods that arrange similar documents into groups [172]. It is the most frequent type of clustering method used in the general text mining literature [172], with many studies applied in the area of mining for customer needs [26, 53, 56, 57]. Some document clustering methods in the customer needs mining literature can be seen as a type of hybrid method, as they first apply document classification models (such as the ones discussed in Section 2.3.1) to only analyze ones containing needs, which are then subsequently clustered [53]. The methods and application scenarios for clustering documents change depending on the study. [26] represents reviews of recliner products from Amazon as a document-term matrix, which is then clustered using a SOM. [56] similarly represents 2,367 customer complaints of air conditioner products (from a South Korean firm specializing in electronics) as a document-term matrix, however, uses hierarchical clustering. Documents can also be analyzed in other formats, as in [53], which clusters document embeddings using hierarchical clustering.

Keyphrase/term clustering involves methods that group similar keyphrases/terms together from a corpus of documents [172]. Although less used than document clustering, keyphrase

clustering is still commonly used in the customer needs mining literature [18, 58–60]. The methods and application scenarios that use keyphrase clustering change depending on the study at hand. [58] analyzes features from Amazon smartphone products to assess changing customer needs following COVID-19 by forming a term embedding matrix using Word2vec [173], which is then clustered using spectral clustering. [59] trains a Continuous Bag of Words (CBOW) model on customer requirements from product reviews of upper limb rehabilitation and mini-fridge products, which are then clustered using Affinity Propagation (AP). [60] performs hierarchical clustering over a document-term matrix to group keyphrases into clusters that co-occur frequently. It is of note that similar semantic keyphrases are grouped in the studies which use word embeddings (e.g. "screen size", "large screen", "screen resolution", "screen technology" [58]) while keyphrases that co-occur are frequently grouped when a document-term matrix is clustered (e.g. "light" and "weight" [60]). Therefore, picking a suitable text representation is required for the task when clustering. Similar to keyphrase clustering, frequent itemset mining has also been used to group similar customer needs [68].

Topic modeling can be seen as both document and keyphrase clustering as each document is a probability over topics, which are in turn distributions over keyphrases. Many approaches use topic models in the customer needs mining literature [31, 45, 46, 54]. Like document clustering, some of these approaches run a document classification model identifying ones that contain needs prior to running a topic model [54]. Various topic models are applied, however, Latent Dirichlet Allocation (LDA) is predominantly used [31, 45, 46, 54]. It is used across various domains e.g. fashion trends [31], smartphones [45, 46], multiple Amazon product ecosystems [54], etc.

Approaches that cluster either documents or terms are very useful for product development teams. Document clustering removes the "winnowing" [53] process (which is required after document classification) with instances discussing the same need being grouped. This can lead to large reductions in the time spent by product development teams by not having to read similar documents [53]. Similarly, keyphrase clustering can also provide reductions in time spent by grouping terms with similar meanings e.g. if the solution is run over a text representation that groups keyphrases by their semantics such as word embeddings [58, 59]. Keyphrase clustering can also group terms by their co-occurrence [60] e.g. if the solution is run over a text representation such as a document-term matrix. Using a technique like this can provide different insights, perhaps prompting a firm using the solution to make a coconut-strawberry shampoo product if the solution is run over a corpus of shampoo posts where the words "coconut" and "strawberry" appear in the same document frequently. How clustering solutions are visually displayed to the user can also be useful to product development teams. Some solutions display a sample of the most representative documents/terms of the cluster [59]. Other solutions display graphic visualizations e.g. [46] uses Polaris [174] (a visual framework for showing relationships between instances) to link the keyphrases they analyze. Similar to Polaris, other popular topic model visualizers could be used e.g. LDAvis [175].

In light of their usefulness for product development teams, the clustering of customer needs has two main drawbacks. The first is that they are difficult to evaluate. Due to this, most solutions in the literature perform no defined evaluation and instead present the output

from their analysis in an example case study e.g. visual output of clusters [26, 31, 45–47, 56, 59, 60, 68]. Other studies use intrinsic cluster or topic model evaluation methods [30, 54, 57, 176–179] e.g. Silhouette score [180]. However, as discussed in the event detection on social media literature and general Information Retrieval (IR) tasks [181], the proper evaluation of clustering solutions requires a manually labelled dataset fit for the task i.e. ground truth cluster of terms representing future customer needs. Obtaining such a dataset for the evaluation of future customer need clusters would be a very difficult task. This would require knowledge of when a group of keyphrases form to become a future customer need that is a highly more challenging task compared to knowing when a single keyphrase will become a future customer need i.e. keyphrase ranking/classification (Section 2.3.3). The second drawback is that it is a difficult method to perform future prediction on in comparison to other methods e.g. it is easier to generate features on the keyphrase level rather than the cluster level for the prediction task. That being said there are a small number of studies in the general text mining literature that have attempted to predict the output of clustering algorithms using techniques such as supervised learning [182] and financial Autoregressive Integrated Moving Average (ARIMA) models [183].

2.3.3 Keyphrase Ranking/Classification

Approaches that work on the keyphrase level mine over collections of documents to either: 1) produce a unique ranking of keyphrases based on how likely they are to be customer needs; or 2) classify unique keyphrases as customer needs.[22] Keyphrase ranking and classification are highly similar as their output can be interchanged. For example, keyphrase classification algorithms can be seen as a type of ranking approach due to how their likelihood output can be ordered e.g. the keyphrase "charcoal" with a likelihood output of 0.97 can be ranked above "table" with a likelihood output of 0.02. Similarly, keyphrase ranking can be seen as a type of classification technique by introducing a cut-off point in the ranking where keyphrases are "classified" as customer needs e.g. the top 20 keyphrases are classified as customer needs.

A body of studies has focused on applying Aspect-Based Sentiment Analysis (ABSA) techniques in UGC [184–187] e.g. product reviews and social media. These studies identify the sentiment of opinion targets/words such as features of products e.g. butter for popcorn products. These methods can assist product development teams to understand more about what features their customers are satisfied/unsatisfied with. However, they also have many other functions (e.g. assisting in purchasing) and are not only designed to detect (potentially new) features in products. Similarly to work on ABSA, studies have focused on classifying product features into predefined categories of customer needs. For example, [188] classifies keyphrases from Amazon product reviews into categories relating to printer product characteristics e.g. Mac compatibility, ease of use, noise, etc. However, these studies do not specifically rank new keyphrases which can be integrated into products. Other work has focused on extracting ranked lists of the most important keyphrases [29, 189–192]. These keyphrases are found using rule-based approaches, with various studies considering

[22] For clarity, by keyphrase classification means classifying unique keyphrases in collections of documents as customer needs, not some task related to the tagging of keyphrases in documents i.e. NER

factors such as frequency or sentiment when ranking [29, 189]. However, the ranking of these keyphrases is not specifically designed for finding customer needs for product development, with many of these studies noting their uses for assistive purchasing information for future buyers based on previous ones [29, 189–192].

Approaches that rank keyphrases for the purposes of product development use various different methods. Basic approaches rank purely based on the frequency the keyphrase occurs in UGC. For example, [61] ranks customer need keyphrases from epinion.com reviews of camcorder products based on 3 defined frequency measures, two including term corpus frequency and document frequency. Other studies include additional features when ranking. The most commonly used additional measure is sentiment [40, 41, 62, 63]. [62] proposes the FEATURE framework, which extracts customer needs from product specifications and then obtains their sentiment in product reviews, which is used in their *Innovativeness* ranking criteria. Other studies [40, 41, 63] use sentiment to rank keyphrases into 3 groups: 1) strong features - which have high sentiment therefore making the product outstanding; 2) weak features - which have low sentiment making the product inferior; and 3) controversial features - which have both high and low sentiment therefore leading to discussion points about it. Approaches other than (or along with) sentiment are also used to rank keyphrases e.g. [193] combines the relationship a candidate customer need keyphrase has with other opinion words (modelled in a bipartite graph) along with sentiment when ranking.

Studies mining customer needs for the purposes of product development also focus the output of the ranking dependent on the problem they solve. For example, [62] proposes two rankings of keyphrases based on whether customer needs are innovative or viable (according to their profit margin). As discussed, [40, 41, 63] ranks customer needs into 3 separate lists according to whether they are strong, weak or controversial features.

Approaches that rank keyphrases are very useful for teams developing products. A drawback of the approach is that keyphrases with similar meanings can occur in the returned ranked list of keyphrases given to product development teams e.g. "coffee" and "java" occurring in the same list.[23] That being said, a large amount of the "winnowing" [53] of information is performed, unlike document classification approaches where analysts are required to sift through documents rather than keyphrases.

There is currently a lack of well-defined evaluation methodologies to assess keyphrase ranking/classification algorithms in the area of mining for customer needs. This is unusual given studies in the general literature that rank keyphrases have proper evaluation methodologies [29, 189–192]. There is also a lack of approaches predicting future customer needs in the keyphrase ranking/classification literature. Similarly, this is unusual given the fact that there is a large number of studies in the general text mining literature performing future keyphrase prediction tasks [194–196] e.g. early detection of whether hashtags are organic or promoted/advertised [194]. Additionally, there is a lack of supervised ML approaches to solving the problem of ranking keyphrases which is unusual given the number of keyphrase classification approaches in the general text mining literature [194–200]. For example, [200] defines some popular families of features for predicting hashtags which include: a) hashtag

[23] According to the Oxford dictionary, "java" means coffee - https://www.oxfordlearnersdictionaries.com/definition/english/java - last accessed 07/06/2024

content features (e.g. number of words in hashtag); b) global tweet features (e.g. sentiment); c) graph features (e.g. average number of followers of users using the hashtag); and d) temporal features (hourly/daily/weekly counts of hashtag occurrences). Studies in the customer needs mining literature analyzing keyphrases could perform a similar task by predicting whether a candidate keyphrase is a customer need. A major obstacle in attaining this mentioned aim is the lack of a ground truth dataset of keyphrases that are customer needs.

2.3.4 Summary and Discussion

At the start of this section, it was mentioned that the 3 primary methods employed in customer need mining tasks would be reviewed in terms of: 1) usefulness to teams developing products; 2) ease of evaluation; and 3) ability for future prediction. Table 2.3 summarizes the discussed methods in terms of the mentioned factors on a 3-point Likert scale. On the Likert scale: 1 aligns with it being difficult to do or not useful; 2 aligns with it being possible but difficult to do or providing some value; and 3 aligns with it being easy to do or providing a lot of value. Document classification provides some use to product development teams (i.e. 2), is easy to evaluate (i.e. 3) and is easy to change to be able to predict future customer needs (i.e. 3) - as discussed in Section 2.3.1. Clustering would be useful for product development teams (i.e. 3), however, it is very difficult to evaluate (i.e. 1) and is very difficult to perform future prediction with (i.e. 1) - as discussed in Section 2.3.2. Finally, Keyphrase Ranking/Classification would be useful for teams developing products (i.e. 3), is possible but difficult to evaluate (i.e. 2) and is again possible but difficult to perform future prediction with (i.e. 2) - Section 2.3.3.

Table 2.3. Comparison of methods used to mine customer needs

Method	Usefulness for Developing Teams	Evaluation	Future Prediction
Document Classification	2	3	3
Clustering	3	1	1
Keyphrase Ranking/Classification	3	2	2

*1 - difficult to use / not useful

*2 - possible / provides some value

*3 - easy to use / provides a lot of value

In this thesis, keyphrase ranking/classification is chosen as the primary method to solve the task of predicting future customer needs. This is because it provides a lot of value to product development teams, yet also presents unique under-addressed research challenges in terms of new prediction methods and evaluation strategies. As discussed in Section 2.3.3, due to the lack of evaluation strategies for approaches ranking/classifying keyphrase customer needs, the discussed Mintel database [79] is used to build a ground truth dataset that evaluates these algorithms (detailed in Chapter 3). Specifically, it is used to check whether the predicted keyphrases produced by the algorithms run over Reddit can predict top trending customer needs occurring in the database at a future time period. Building

such a dataset along with an evaluation strategy to use for the assessment of keyphrase ranking/classification algorithms is a significant step in the thesis that addresses RQ 2.

In addition, this ground truth dataset is also used when building a supervised ML model to predict customer needs addressed in future new-to-market products (Chapter 5). This is significant as there is a lack of approaches in the literature using ML to predict future customer needs (as discussed in Section 2.3.3). This is done by using techniques from MTSC by classifying keyphrases based on 1263 univariate time series coming from 10 families of features (e.g. product-based, linguistic-based, sentiment-based, frequency-based etc.). Doing this (along with the rule-based approach in Chapter 4) addresses RQ 1 that future customer needs can be predicted.

Furthermore, this thesis also explores the use of MTL (a subarea of supervised ML) using techniques from keyphrase ranking/classification that address RQ 3. The general approach of MTL has been applied in many applications of ML, including but not limited to image classification [201], NLP tasks (such as sentiment analysis) [202, 203] and time series classification [204, 205]. It has also been used in unrelated customer needs mining tasks e.g. understanding customer needs from vehicle behaviour [206]. However, it has not been used in customer needs mining tasks related to this thesis let alone in the keyphrase ranking/classification family of methods. In this thesis, it is used to learn an instance of a general customer need across multiple product categories, so that it can be used for a category it has or hasn't seen during training e.g. build a model on needs from toothpaste, lip balm and soda to predict for a seen category like toothpaste or even an unseen category like pizza. During experiments conducted (in Chapter 5), it is shown how using MTL in this manner results in a high-performing model capable of accurately predicting for categories it has and hasn't seen during training.

2.4 Applications of Customer Need Mining

Other than data source and methodological differences, studies in the area of customer needs mining also differ on the application level. Relevant to the work in this thesis, is how approaches differ on: 1) the types of products analyzed; 2) the types of needs the approaches mine for; and 3) whether they analyze needs over time.

Of note regarding these differences is how studies mainly: 1) analyze the product model level rather than the category level e.g. Toyota instead of cars (the types of products analyzed); 2) mine for general needs rather than ones that could be of higher business interest e.g. unmet needs (the types of needs the approaches mine for); and 3) generate needs in a once-off analysis rather than over multiple time periods (whether they analyze needs over time). This section discusses these differences in more detail.

2.4.1 Types of Products Analyzed

The majority of studies in the literature focus on analyzing customer needs for a specific product model such as a smartphone product e.g. [45, 46] both extracts needs for the Samsung Galaxy Note 5 by mining on the subreddit r/galaxynote5.[24] Other studies extract needs for multiple product models e.g. [63] finds needs for 4 smartphone models (e.g. iPhone4, Motorola Droid RAZR, etc.), [41] extracts for 4 automobile models (e.g. Tesla Model 5, Honda Civic, etc.) while [40] extracts for 4 models of smartphones and 4 models of automobiles. However, there is a lack of studies mining on the product category level e.g. mining for general smartphone needs on social media rather than a particular model (e.g. iPhone4). Some that do include a document classification technique [53], which not only extracts needs on the category level but also for multiple categories i.e. toothpaste, kitchen appliances, skin treatment products and prepared foods. Likewise, [55] classifies documents containing innovation ideas across 20 categories of Amazon products.

Similarly, the approaches detailed in this thesis analyze the product category level over multiple categories e.g. dog food, toothpaste, cereal, etc. The application of mining on the category level has different outcomes to mining on the model level, mainly the enhanced capability of discovering needs that customers generally want in products in comparison to ones they want for a specific product model. For example, general needs in mobile phone products may consist of ones that can monitor health information while needs for a particular product may be more specific to the model under analysis e.g. improved battery time for the Samsung Galaxy 5. It could be argued that the approach of mining on the product category level is a lot more important than on the product model level for product development. This is because needs that are predicted to be of future importance on the category level represent needs in the general market for a product (e.g. automated cleaning technology for toothbrush products), while needs that are predicted to be of importance on the model level only represent needs for that product model (e.g. enhanced camera for iPhone). Therefore, analyzing at the category level allows more important questions to be answered (e.g. what will be popular in shampoo products in 3 years) compared to analyzing at the model level (e.g. what will be popular in Dove Radiant Shine shampoo products in 3 years).

Due to the lopsided number of studies analyzing the product model level, the algorithms developed in this thesis focus on the product category level. Due to this, the corpus of social media posts analyzed consists of posts containing general product terms instead of specific models of products e.g. collect a corpus of posts containing either "perfume" or "fragrance" for perfume products rather than posts containing "Coco Mademoiselle".[25] Perhaps one of the main reasons why studies are hesitant to analyze on the product category level is because they are difficult to evaluate as they require a ground truth dataset of customer needs representing multiple products. However, with access to Mintel GNPD (a massive database of new-to-market products), it is possible to formulate this dataset of top trending customer needs from products in the marketplace which makes it possible to train/evaluate novel approaches in customer needs mining on the product category level.

[24] https://www.reddit.com/r/galaxynote5 - last accessed 07/06/2024

[25] Coco Mademoiselle is a Chanel brand of ladies perfume - https://en.wikipedia.org/wiki/Coco_Mademoiselle - last accessed 07/06/2024

2.4.2 Types of Customer Needs

Many studies mining customer needs in the literature do not go beyond detecting general needs in UGC to identify ones that may be of more business interest e.g. specifically looking for unmet needs. For example, some document classification studies only detect "purchasing intent" [42, 43, 49, 67] or only distinguish posts containing a customer need [38, 53, 207] without determining whether the document contains information that is of higher business interest e.g. an innovation that can disrupt the market. Similarly, some clustering approaches only detect groups of documents/terms forming needs discussed in UGC without highlighting ones that have more value e.g. [31] detects groups of general fashion needs in Amazon and Rakuten reviews. The same can be said for some keyphrase ranking approaches that consider the factors of frequency and sentiment when sorting phrases, without specifically identifying ones that are of perceived business value [29, 189–192].

However, some studies do focus on mining needs that are of higher business interest. [55] classifies documents that contain customer needs detailing product innovations that could be seen as more important than detecting general needs. Similarly, [16] mines for needs that are of higher business value by discovering unmet needs in home appliances using a Context Tree.[26]

Other methods find needs of higher business value by implementing previous business models/methodologies to extract needs that are unmet/unsatisfied e.g. Kano model, Kansei engineering, etc. (as discussed in Section 2.1.2). Although some of these business models/methodologies focus on purely weighting already identified needs (as discussed in Section 2.1.2), computational approaches that implement them also address the task of finding them. Identifying customer needs through business methodologies is an interesting problem as it allows firms wanting to detect customer needs to narrow their focus on the ones that are not addressed (or only partially addressed). In [45, 47], the idea of an "opportunity algorithm" is implemented to find new or existing customer needs. As initially described in [208], this algorithm works on the basis that if a need has high importance but low satisfaction then a business opportunity is present. In [45, 47], these importance and satisfaction values are computed based on the frequency at which a need is discussed (importance) along with the sentiment of it (satisfaction). As implemented in [45, 47], needs that have high keyphrase frequency and low sentiment are defined as being unmet. This approach to identifying needs of greater interest to business builds on previous literature, however, it has been described as simplistic [208–210] and is criticized for this reason i.e. not all unmet needs conform to having high importance and low satisfaction. Another business model implemented by computational methods is the Kano model [24, 28, 54, 211, 212] (discussed in Section 2.1.2). The Kano model goes beyond detecting general customer needs by providing a categorization/prioritization framework of needs into 3 groups i.e. Basic, Performance and Excitement (as discussed in Section 2.1.2). For example, [211] classifies LDA topics into the 3 mentioned categories including 2 more (reverse and indifferent). [54] also implements the Kano model by applying sentiment analysis to the output returned by LDA to get the levels of satisfaction and dissatisfaction of customer needs in the form

[26] The Context Tree refers to a data structure that extends branches by searching other detailed keywords related to a specific keyword iteratively [16]

of topics. The studies implementing this model go beyond just detecting general needs by providing businesses with more information on whether they should use them in their product and hence may be of more use in certain product development scenarios. Kansei engineering (initially discussed in Section 2.1.2) is another business model that has been attempted to be automated by computational approaches. Previous business studies have implemented this method through the use of questionnaires, in which respondents rate their feelings on a point scale between two opposite pairs of words (one positive and one negative), known as Kansei attributes [115, 116]. However, recently automated studies using text mining and ML have tried to solve this problem using UGC to reduce the time it takes to gather requirements by carrying out questionnaires [26, 213–215]. An example of this is [213], who shows that their algorithm run over Amazon product reviews can extract emotions towards a customer need with high precision and recall. By generating different types of sentiment for a particular customer need, the framework allows different types of needs to be identified rather than general needs.

There are also various studies attempting to find future customer needs in the literature [35, 64, 65]. These studies predict the frequency of a keyphrase in UGC (e.g. social media) using regression-based techniques. For example, [35] extracts needs and the frequency of the needs from customer reviews of cell phones which are then predicted using Holt-Winters exponential smoothing. In their evaluation, they show how features such as "battery life" are predicted over time. In [64], a fuzzy time series model is applied to predict the future importance (based on frequency and sentiment) of customer needs from Amazon reviews of electric iron products. In their experiments, they compare their approach to other time series models and show that their fuzzy time series models can predict the importance of customer needs better than other classical time series models (e.g. Simple Moving Average). Similarly, the work in this thesis tries to predict future customer need keyphrases (i.e. RQ 1), however, with two key distinctions to other studies. First, the output of the keyphrases is presented in a ranked list of top customer needs. Secondly (and most importantly), the keyphrases are predicted based on whether they will appear as top trending needs in a database of new-to-market products instead of forecasting how popular they will be in UGC. This is a more interesting task to solve as keyphrases that will be popular in the marketplace are more useful to teams developing products than ones that will trend on social media. The main ideology of finding future customer needs that will become popular in the marketplace is that firms address needs in products in response to demand for them, hence these needs are unmet. Therefore, the approach of finding future customer needs in this thesis is analogous to discovering currently unmet needs in the marketplace.

2.4.3 Analyze Needs Over Time

Approaches in the literature that perform clustering or keyphrase ranking/classification can be split on whether they analyze needs over time or not.[27] Most studies don't, instead opting to analyze static collections of documents with no associated timestamp [26, 31, 40, 41, 54, 60–63, 193, 216]. Comparatively, a small number of other approaches analyze needs over time by embedding time-associated information in their algorithmic process e.g.

[27] Document classification algorithms can't analyze needs over time, hence their exclusion.

timestamp. For example, [47] tracks the evolution and growth of customer need topics using an Event Detection and Tracking clustering model. Similarly, the approaches applied in this thesis are analyzed over time by generating lists of keyphrase needs each month.

As pointed out in a recent study [47], methods that fail to analyse needs over time consequently hinder their effective use in business contexts that require constant timely feedback in order to make products or new product features that are required by customers. Methods which analyze over time are also more secure from an evaluation standpoint as they are analyzed each time they generate needs (as in this thesis). This is compared to the once-off evaluation seen when evaluating a static collection of documents, which can lead to an increased chance of the algorithm overfitting on the testing set.

2.4.4 Summary and Discussion

Relevant to the research in this thesis, approaches tend to differ on the application level through the following criteria:

- **Types of products analyzed**: Most studies analyze needs on the product model level (e.g. Samsung Galaxy 5) rather than the product category level (e.g. mobile phone).
- **Types of customer needs detected**: Many studies detect general customer needs in UGC, while others highlight ones that are of more interest to firms developing products (e.g. unmet needs).
- **Analyze needs over time**: The majority of studies detect customer needs from static collections of documents, while others analyze needs over time which is more aligned with real-world business environments where customer requirements are needed consistently.

From the following application-level differences, the approaches detailed in this thesis can be described as ones that analyze the product category level (types of products analyzed), detect future customer needs (types of customer needs detected) and analyze needs over time (analyze needs over time).

2.5 Evaluation

In this section, the evaluation approaches used in the literature are discussed. First, studies used to evaluate customer need mining algorithms are reviewed. Here, the lack of approaches that use a ground truth dataset is highlighted. Secondly, a brief review of a small number of studies that evaluate algorithms through user studies is given. Similar to the lack of a ground truth dataset for automated evaluation purposes, there is a lack of approaches conducting user-based evaluations involving firms developing products. This is reviewed as

a user study is conducted in Chapter 6. Here an overview of user studies in recommender systems is also given, as the approaches developed in this thesis can be seen as a type of recommender system - suggesting customer needs to product development teams. Thirdly, evaluation approaches from two branches of literature relevant to the task discussed in this thesis are discussed: 1) keyphrase-based evaluation; and 2) future prediction evaluation. These two branches are chosen due to the approaches in this thesis predicting future customer needs in the form of keyphrases. Finally, a summary and discussion is provided.

As touched upon, two areas in which the literature in evaluating customer needs is lacking are: 1) ground truth dataset; and 2) user studies involving product development teams. These two areas are addressed as RQ 2 (ground truth) and RQ 4 (user study) in this thesis. The lack of these evaluation approaches is referred to throughout this section.

2.5.1 Evaluation Approaches in the Customer Needs Mining Literature

As previously discussed in Section 2.3, the difficulty of defining an evaluation approach depends on the method being used. For document classification tasks (e.g. does this document contain a customer need), the evaluation approach uses the ground truth label as an indicator. This label could be extracted from a relatively straightforward annotation process, i.e. manually label a sample of documents into classes. Here, general ML metrics such as accuracy, precision, recall and F1 are calculated [38, 42, 43, 49, 53–55, 66, 67].

However, as discussed in Section 2.3.2, for clustering and topic modeling, the evaluation can be more difficult to define, i.e. how do you say that a cluster/topic of keyphrases represents a customer need that is useful for product development? Due to this, some studies don't evaluate their approaches at all and instead demonstrate the practicability of their approach by manually outputting the keyphrase clusters/topics [26, 31, 45–47, 56, 59, 60, 68] while some use general intrinsic cluster validation measures [30, 31, 54, 57, 176–179] e.g. Silhouette score [180], Davies–Bouldin index [217], perplexity [218], topic coherence [219], Bayesian Information Criterion (BIC) [220], etc. However, as discussed in the event detection on social media literature and general IR tasks [181], the proper evaluation of clustering solutions requires a manually labelled dataset fit for the task i.e. ground truth cluster of terms representing future customer needs. In the clustering literature, there is a lack of well-defined automated evaluation approaches using a ground truth dataset. [53] is one viable approach to evaluating the output of a clustering solution producing customer needs. It evaluates by conducting a large user study that compares needs extracted from its clustering solution to needs identified by a product development firm using traditional techniques (e.g. user interviews). Although the use of user studies to evaluate algorithms capable of detecting customer needs is an area that is not heavily addressed (discussed later in this section), a drawback of the approach is that it is a once-off non-repeatable evaluation strategy.

Like clustering, the evaluation for keyphrase ranking/classification can be difficult to define (although not as difficult) e.g. how do you determine whether a keyphrase is a customer

need? One common methodology applied to evaluate keyphrase algorithms is a simple manual examination of the top set of keyphrases produced to obtain performance metrics [40, 41, 63]. For example, [63] examines a list of the top keyphrases generated from tweets about 4 models of mobile phones to decide whether they are relevant customer needs. One of the main pitfalls of this approach is that it is only able to calculate precision but not recall, as a list of the total set of needs is not generated. By not calculating recall, no measure is provided on the total set of customer needs missed by the algorithm. Another drawback is that to properly assess using this methodology, human evaluators with experience in product development are required. However, evaluators with such experience are not used in the mentioned studies [40, 41, 63].

For the proper evaluation of keyphrase ranking algorithms, a ground truth dataset of keyphrases is required. A group of studies ranking keyphrases have curated ground truth lists that allow metrics such as precision and recall to be calculated [29, 189–192]. However, these studies don't generate these lists with the specific task of mining needs for product development in mind (used to assess the ability of an algorithm to aid in ranking keyphrases for assistive purchasing information). To formulate a ground truth dataset of keyphrases for the assessment of algorithms used in product development, ones that are known to be needs important to customers are required. These important keyphrase needs may be obtained from various data sources, such as product descriptions/specifications (particularly ones that are sufficiently rich in information expressing customer needs addressed in the products) or transcripts from user interviews that express needs stated by real customers.[28] Another data source that is widely used in the general ML literature is product sales [222–224], however, it is not useful for the evaluation of customer needs mining tasks as it is not rich in textual information detailing needs required by customers. As discussed previously, Mintel GNPD is used heavily in this thesis to extract ranked lists of keyphrases that make up the ground truth (RQ 2) and are used to evaluate the algorithms developed that are run over social media and train supervised algorithms capable of predicting future customer needs. Mintel GNPD is a large database of product descriptions across multiple product categories e.g. shampoo, toothpaste, popcorn, etc. Of specific relevance is that the description for each product in the database details claims and features included, which in turn reflect customer needs (discussed more in Section 3).

2.5.2 User Studies

Along with the lack of ground truth datasets available for the evaluation of customer needs mining tasks, there is also a lack of studies involving humans in the evaluation process to either assess the output produced by an algorithm and/or compare it to their generated output. Such studies are scarce as they involve the use of participants who likely work at marketing firms or in companies producing products - who already researched to understand whether a particular need is useful e.g. through user interviews or reading relevant web content such as product reviews. These participants are difficult to persuade to assist in such studies (as pointed out in [53]), likely because they work in firms that don't allow

[28] In user interviews, customers are usually invited to the office premises to use the product and state their opinions about it e.g. what they wish it included or what they wish it could do better [221].

them to disclose important information that could assist in making useful products that their competitors don't address.

There are a few approaches that do involve human evaluators in the process to fully/partly assess an algorithm detecting customer needs. As discussed, some approaches evaluate lists of keyphrase customer needs [40, 41, 63], however, give no indication of who evaluated the output of the algorithm or how they defined a generated need as correct. Currently, there is only one study that carries out a user evaluation with a firm specializing in product development. It uses developers from a professional marketing firm to read through Amazon review documents classified from a CNN model as containing "customer needs" and then clustered using hierarchical clustering [53]. The total number of unique needs identified by the marketing firm through reading the classified documents is compared to a separate analysis done by them which identified needs using traditional methods e.g. user interviews. A major contribution of the study is that it provides valuable insights into the intersection of needs detected by both methods (high number) as well as non-overlapping needs detected by the individual methods (low number). This way it can properly evaluate the usage of the approach in comparison to traditional methods and state the additional benefits of their automated ML approach e.g. less expensive in terms of time and money costs. Similarly, an approach in Chapter 6 of this thesis conducts a user study that recommends customer needs to a product development team from a noteworthy MNC, which as of 2023 is valued in the billions (USD) and has over 500 products in the marketplace (addresses RQ 4). In addition, another small user study is also performed in a sub-evaluation that checks whether one of the algorithms detailed in this thesis can detect the top 6 needs identified by the same described company (detailed in Chapter 4).

2.5.2.1 User Studies in Recommender Systems

As discussed in this section, user studies are a valuable evaluation approach, sparingly used in the customer needs mining literature. Because algorithms mining customer needs can also be seen as a type of recommender system (suggesting customer needs that it predicts will become popular in the future), a brief review of them is given. When reviewing, the salient topics performed during the user studies are highlighted and summarized e.g. questions asked, types of participants, etc. Some of the studies analyzed are from product development/business mining topics related to the research in this thesis [225–230] while some are not [231–234], however, still contain valuable lessons from user evaluations relevant to this thesis.

Studies ask quantitative and sometimes qualitative questions to participants about the quality of the received recommendation(s). On the quantitative side, respondents are usually asked questions that are relevant to the overall aim of the study e.g.: 1) for a model recommending products, users are asked to rate the suggestion [226, 227]; 2) for a model that ranks review helpfulness, users are asked to rank the reviews which are then compared to the model's ranking [230]; and 3) for a model that provides career suggestions, users are asked to rate the profession recommendation [231]. When performing quantitative evaluations, the described approaches are also usually compared against some baseline relevant to the study e.g.: 1) a model summarizing customer requests in the software

development life cycle is compared to the approach of manually creating tickets [225]; 2) a model which aims to mitigate discriminatory bias (e.g. gender, age, race, etc.) for career recommendation is compared to an approach which doesn't [231]; 3) a model (that is a plugin to a Jupyter Notebook) which automatically does a certain number of automated tasks for ML model building recommendation for a given dataset is compared to the usability if a user just uses google to build the ML model; and 4) an emoji recognition model which is trained on the conversation level is compared to the general approach of training it on the sentence level [234]. Furthermore, to compare the approach and the baseline for quantitative evaluation questions some metric is recorded for a user to exhibit their thoughts towards a recommendation e.g. Likert/star-rating scale [228, 233], ranking items [229], blindly choosing between the recommendations produced by the approach and the baseline [232], etc. Some studies also allow for some free-form text answers to allow users to discuss any further thoughts on the recommendations [232, 233] i.e. qualitative feedback. Full user responses or text snippets are then reported in the study [232, 233]. This allows the study to relay any further information not found during quantitative questioning.

When sourcing participants for user studies, previous research has aimed to find relevant candidates for the task at hand e.g.: 1) there are "21 CRM experts and 33 project managers" involved in [225] which summarizes software requests for the software development life cycle; 2) a study in a restaurant review comparison system only recruited participants with prior experience with "OCR browsing and online shopping" [229]; 3) for a career recommendation system, only "students from all majors" [231] were invited to participate in the study; and 4) for a recommendation system which provides suggestions in the ML model building process only participants which had a background in "computer science and engineering" [233] are chosen. As pointed out in [235], a good diversity of participants (e.g. age/gender balance) is also a good way to get more varied opinions.

In the user study performed in Chapter 6, the described literature is followed by asking quantitative questions. Although no qualitative feedback is gathered, additional feedback in the form of a questionnaire is also recorded. A relevant source of participants from a product development team from a major MNC specializing in making CPG products is also used in the study. These participants are diverse in various attributes e.g. job title, age, gender, etc.

2.5.3 Evaluating Lists of Future Customer Needs

Considering the approaches developed in this thesis mine future customer needs in the form of keyphrase ranking/classification, two main branches of literature are considered when evaluating: 1) keyphrase-based evaluation; and 2) future prediction evaluation. These two branches are considered for an overall evaluation approach as well as specific metrics that are suitable for the task. It is noteworthy that the literature reviewed here is mainly from the general text mining tasks rather than specific to customer needs mining, as there is a lack of evaluation approaches for these types of problems.

2.5.3.1 Keyphrase-based Evaluation

As discussed in Section 2.3.3, keyphrase-based approaches can be outputted in either: 1) a ranked list of keyphrases that are most likely to be customer needs; or 2) a classification output detailing whether a keyphrase is associated with a class label. How these two outputs are evaluated differs.

For keyphrase ranking tasks, the main evaluation approach taken is to generate a ground truth (or gold standard) list of ranked keyphrases which are then compared to the algorithm output [29, 189–192, 236, 237]. Depending on the task at hand, this ground truth list can be formulated relatively easily by having annotators read through a sample of the documents being analyzed and producing a ranked list e.g. [237] have annotators read through lecture transcripts and produce a list of the top keyphrases they think are most important for the evaluation of a keyphrase ranking algorithm run over the same transcripts. However, for the task described in this thesis, formulating the ground truth list is not as simple, as it has to represent future customer need keyphrases occurring in real market products that can't be constructed successfully by only having annotators read through documents. Therefore, this is the reason why the ground truth in this thesis is constructed using analytical techniques by tracking the most important keyphrases addressed in real products which are labelled as customer needs by annotators (detailed in Chapter 3). When comparing the algorithm's output to the ground truth, typically two types of metrics are calculated: 1) list ranking specific metrics e.g. Discounted Cumulative Gain (DCG) or Normalized Discounted Cumulative Gain (NDCG) [29, 189, 191, 192, 236, 237]; or 2) slightly altered ML metrics e.g. of precision, recall, F1, etc. [190].[29]

Similarly, to keyphrase ranking tasks, approaches that classify keyphrases require a ground truth dataset for both the learning of the supervised model used to make predictions and the evaluation of it [194–196]. For example, [195, 196] forms the ground truth based on whether a keyphrase appears in Twitter's trending list the next day for their virality prediction task framed as a binary classification problem. [194] forms its ground truth based on whether a Twitter keyphrase is trending because it is promoted or organic, for its task of detecting promoted campaigns. When comparing the ground truth and prediction output, these studies use general classification metrics used in the ML literature e.g. accuracy, precision, recall, F1, AUC-ROC, etc.

2.5.3.2 Future Prediction Evaluation

There are a limited number of studies that focus on predicting future customer needs in the literature i.e. using UGC to predict keyphrase customer needs that will be of importance in the future. Approaches that do are mainly framed as a regression problem [35, 64, 65] which then go on to apply metrics such as Prediction Error [64] or Mean Absolute Percentage Error [35] to evaluate. These approaches have large prediction errors indicating the difficulty of the future customer needs prediction problem. A problem with these approaches is that they don't predict a meaningful dependent/target variable for product development (e.g. if the need will be addressed in future products) and instead forecast other variables based on

[29] DCG and NDCG are common measures for comparing two lists of items [238]

the sentiment or frequency of a keyphrase in UGC. An example of an unrelated task that does predict a meaningful dependent variable is [239], which correlates the "chatter" for a movie on Twitter to actual box office revenue.

More aligned with the task in this thesis, many studies in the ML literature perform the future prediction of some piece of UGC framed as a binary classification problem. Some consist of predicting the popularity of hashtags [195, 196, 198], early detection of political campaigns as promoted or organic [194], predicting the success of books [240], predicting the popularity of scholars [241], etc. The evaluation approaches in these studies take into account two main criteria during assessment: 1) is the prediction correct; 2) is the prediction detected with some amount of lead time useful for the task employed? For example, [195, 196] only reported the metrics for the hashtags they predicted a day before they trended on Twitter. Similarly, [240] only reports the AUC-ROC of its model 1 year into the past at predicting the success of books. [194] also reports the AUC-ROC, however, it does this across multiple time periods e.g. 20 minutes away, 40 minutes away, etc. Similarly, the approaches in this thesis only report the output of the results if a need is detected in some time period in the future.

2.5.4 Summary and Discussion

As discussed at the start of this section, there is a lack of studies evaluating the output of their algorithms detecting customer needs using: 1) ground truth datasets; or 2) user-based evaluations. Baring document classification techniques (where one is required to perform the task), there is a lack of techniques using a ground truth dataset. This is because it is difficult to construct such a dataset, which requires high-quality input data from real products on the market. There is also a lack of studies that perform user-based evaluations with participants from teams developing products. This is because said participants are difficult to persuade to assist in such studies [53], due to firms not wanting them to disclose important information that could assist in making useful products.

In Chapter 3, a ground truth dataset is constructed along with an evaluation methodology that allows the automatic evaluation of keyphrase ranking/classification algorithms that generate customer needs (addresses RQ 2). To help guide the construction of this dataset and methodology, studies in the text mining and ML literature were reviewed which perform related tasks to the algorithms developed in this thesis i.e. studies that perform 1) keyphrase-based evaluations; and 2) future prediction evaluations. In Chapter 6, a user study is performed with participants from a product development team from a large MNC (addresses RQ 4). To assist in conducting this study, some general techniques used to perform user studies in the recommender systems literature are reviewed as the algorithms developed in this thesis can be seen as a type of recommender system (suggesting customer needs to product development teams).

2.6 Multivariate Time Series Classification

In this section, the MTSC techniques used in this thesis are detailed. Knowledge of general statistical, ML, text mining and NLP concepts are assumed and are therefore not discussed such as BoW, Principal Component Analysis (PCA), general ML classifiers (e.g. Linear Ridge Regression), etc. However, a brief overview of MTSC techniques is given due to the recent and fast advances in the area along with the potential unfamiliarity of these techniques between ML researchers. MTSC is not necessarily a commonly used technique in the customer needs mining literature, however, a review of the techniques is given due to its relevancy in this thesis.

Unlike univariate time series classification where an instance is a time series with a number of temporally ordered observations and output class, MTSC is a list of vectors with a number of dimensions along with a number of observations and an output class [78]. As of 2018, an archive of 30 multivariate time series datasets has been released (diverse in series length, number of dimensions and number of output classes) allowing for the benchmarking of algorithms run over these data types which has led to an increase in research in this area [242]. Additionally, the inclusion of libraries implementing many popular algorithms in the field has been made publicly available, allowing studies showing the applicability of these algorithms to be made. Two popular ones include 1) sktime - a python-based package compatible with sklearn [243] and 2) tsml - a java-based package compatible with Weka.[30][31] Although these algorithms have not been applied often when mining for customer needs, they have been applied in related areas of this thesis subject area e.g. smart manufacturing [244] and customer churn prediction [245].

So to solve a keyphrase classification task (discussed in Section 5), where each keyphrase is represented by multiple univariate time series, the multivariate algorithm MINImally RandOm Convolutional KErnel Transform (MINIROCKET) [246] is used. MINIROCKET is a faster version of the RandOm Convolutional KErnel Transform (ROCKET) algorithm [247], which has been shown to obtain better results in terms of speed and accuracy than comparative approaches [78]. ROCKET is an algorithm for transforming a 2D multivariate time series into a 1D vector using random convolutional kernels. This 1D vector is then used as an instance to train a linear classifier such as Ridge/Logistic Regression [247] to solve the classification task. MINIROCKET on the other hand, is a deterministic algorithm that speeds this ROCKET transformation process up to 75 times faster on large datasets [246]. The speed of MINIROCKET is particularly useful for a specific analysis in this thesis where the collection of time series generated is very large - many thousand (keyphrase) instances each with 1263 univariate time series of dimension 36 (representing 36 months in the past).

[30] https://www.sktime.org/en/stable/ - last accessed 07/06/2024
[31] https://github.com/time-series-machine-learning/tsml - last accessed 07/06/2024

2.7 Summary and Discussion

In this chapter, the literature relevant to this thesis was reviewed. Firstly, a brief review of how teams at firms developing products identify customer needs was provided (Section 2.1), with the research in this thesis being geared towards aiding these types of people with their jobs. Secondly, the data sources used by computational methods were discussed in terms of (Section 2.2): 1) accessibility; 2) whether it represents the VOC; 3) audience/data size; and 4) usefulness for mining customer needs. It was shown that most studies use either product reviews or social media when mining customer needs. After the main methods employed in customer needs mining studies were detailed (Section 2.3): 1) Document Classification; 2) Clustering; and 3) Keyphrase Ranking/Classification. Studies using these methods were discussed and the main methods themselves were compared in terms of themes relevant to the work in this thesis: 1) usefulness to teams developing products; 2) ease of evaluation; and 3) ability for future prediction. How studies differ on the application level was then discussed (Section 2.4). The differences discussed include: 1) types of products analyzed (e.g. product model level or product category level); 2) types of customer needs analyzed (e.g. unmet needs); and 3) whether studies analyze needs over time. How studies evaluate their approaches was then discussed (Section 2.5). Specifically, the lack of studies that use ground truth datasets or conduct user studies involving firms developing products was highlighted. Finally, a high-level overview of the techniques used in MTSC was given as it is an important technique used in Chapter 3 of this thesis (Section 2.6).

Table 2.4 summarizes some of the key studies relevant to the work in this thesis and compares them in terms of some of the main points discussed in this chapter. As seen in the table, there is a lack of studies addressing the RQs asked in this thesis. The first RQ is addressed by developing an algorithm capable of predicting future customer needs. This is done by proposing rule-based (Chapter 4) and supervised ML-based (Chapter 5) keyphrase ranking/classification approaches to the problem. During evaluation, it is shown that these techniques provide significant lead times useful in aiding product development teams. The second RQ is addressed by proposing methods to curate a ground truth dataset capable of evaluating/training algorithms that predict future customer needs (Chapter 3). This is done by extracting ranked lists of keyphrases representing customer needs from Mintel GNPD - a large database of new-to-market products. The third RQ is addressed by employing MTL to train a single model capable of predicting future customer needs (Chapter 5). This is significant as this model can still accurately predict future needs for a category it doesn't use in the training process e.g. can be trained on Toothpaste, Eyeliner and Shampoo needs yet still predict needs for Dog Food products. Finally, the fourth RQ is addressed by conducting a user study involving participants from a large MNC to discover whether the needs predicted by a model proposed in this thesis can potentially be addressed in their new product lines (Chapter 6). This is significant as there is a lack of studies carrying out research of this nature.

Table 2.4. Summary of Studies Mining Customer Needs

Study	Data Source	Methodology	Types of Products	Types of Customer Needs	Analyze Needs Over Time	Supervised ML	Future Prediction (RQ 1)	Ground Truth Dataset (RQ 2)	Multitask Learning (RQ 3)	User Study (RQ 4)
[38]	Social Media (Twitter)	Document Classification	Product Category (E-Mobility)	General Needs	n/a	Yes	No	Yes	No	No
[55]	Product Reviews (Amazon)	Document Classification	Multiple Product Categories	Product Innovations	n/a	Yes	No	Yes	No	No
[56]	Warranty Databases	Document Clustering	Product Category (Eletronics)	General Needs	No	No	No	No	No	No
[26]	Product Reviews (Amazon)	Document Clustering	Product Category (Recliner)	General Needs	No	No	No	No	No	No
[53]	Product Reviews (Amazon)	Document Clustering	Multiple Product Categories	General Needs	No	No	No	No	No	Yes
[57]	Product Reviews (Unspecified)	Document Clustering	Product Model (Diaper Bag)	General Needs	No	No	No	No	No	No
[58]	Product Reviews (Amazon)	Keyphrase Clustering	Product Category (Smartphone)	General Needs	Yes	No	No	No	No	No
[59]	Product Reviews (Multiple Websites)	Keyphrase Clustering	Multiple Product Categories	General Needs	No	No	No	No	No	No

Table 2.4 Continued: Summary of Studies Mining Customer Needs

Study	Data Source	Methodology	Types of Products	Types of Customer Needs	Analyze Needs Over Time	Supervised ML	Future Prediction (RQ 1)	Ground Truth Dataset (RQ 2)	Multitask Learning (RQ 3)	User Study (RQ 4)
[60]	Product Reviews (Amazon)	Keyphrase Clustering	Product Category (Violin)	General Needs	No	No	No	No	No	No
[31]	Product Reviews (Amazon & Rakuten)	Topic Model	Product Category (Fashion)	General Needs	No	No	No	No	No	No
[45]	Social Media (Reddit)	Topic Model	Product Model (Samsung Galaxy)	Unmet Needs	No	No	No	No	No	No
[46]	Social Media (Reddit)	Topic Model	Product Model (Samsung Galaxy)	Product Opportunities	No	No	No	No	No	No
[61]	Product Reviews (Epinion)	Keyphrase Ranking	Product Category (Camcorder)	General Needs	No	No	No	No	No	No
[62]	Product Reviews	Keyphrase Ranking	Product Model (Samsung Galaxy)	Innovative & Viable Needs	No	No	No	No	No	No
[63]	Social Media (Twitter)	Keyphrase Ranking	Multiple Product Models (Smartphone)	Strong, Weak & Controversial Needs	No	No	No	No	No	No

Table 2.4 Continued: Summary of Studies Mining Customer Needs

Study	Data Source	Methodology	Types of Products	Types of Customer Needs	Analyze Needs Over Time	Supervised ML	Future Prediction (RQ 1)	Ground Truth Dataset (RQ 2)	Multitask Learning (RQ 3)	User Study (RQ 4)
[40]	Social Media (Twitter)	Keyphrase Ranking	Product Models (Smartphone & Automobile)	Strong & Weak Needs	No	No	No	No	No	No
[41]	Social Media (Twitter)	Keyphrase Ranking	Product Models (Automobile)	Strong, Weak & Controversial Needs	No	No	No	No	No	No

Chapter 3: Data

This chapter provides details on the data used in this thesis i.e. ground truth and Reddit social media data used for the customer need keyphrase prediction task. First, a visual overview of two ground truth datasets described in this chapter is detailed (Section 3.1). Secondly, an overview of Mintel GNPD is given, which forms the basis of both the ground truth datasets (Section 3.2). Specifically, Mintel GNPD's applicability as a high-quality source of information suitable for the curation of ground truth datasets able to evaluate customer needs mining algorithms is detailed. The curation of the first ground truth dataset is then detailed (Section 3.3), which is formed for only 1 product category i.e. Toothpaste. This works by first detecting customer need keyphrases using a NER model and then ranking them by their occurrence in Mintel to form a list of keyphrases representing customer needs each month. The second ground truth dataset is then described (Section 3.4), which is curated for 37 product categories all in the area of CPG. This dataset first ranks keyphrases using techniques from text mining then has humans label the ranked keyphrases to form lists of keyphrases representing customer needs each month. The evaluation approaches used to assess the algorithms run over Reddit are detailed, which includes a discussion of how the ground truth datasets are used in evaluation scenarios (Section 3.5). Finally, a summary of the Reddit social media data used in each of the remaining chapters of this thesis is detailed (Section 3.6).

As stated in Chapter 1, the goal of this chapter is to evaluate RC 3 (which states that two ground truth datasets are curated for the training and evaluation of algorithms predicting future customer needs). This contribution addresses RQ 2 which questions how ground truth datasets can be curated to allow the training/evaluation of approaches predicting future customer needs. This question is addressed throughout this chapter. Additionally, it's of note that a large portion of the work in this chapter is taken from [81] (evaluation only) and [83].

3.1 Overview of Ground Truth Datasets

In this section, a brief overview of the ground truth datasets curated in this chapter are discussed. Figures 3.1 and 3.2 show an example visual output of the ground truth for the categories Toothpaste and Dog Food respectively. To summarize, the output shows a ranked list of the top 20 keyphrases each month from 2011-2021 for each product category. The keyphrases represent the most heavily addressed customer needs in real products in the marketplace for the month of interest. These are shown in Figures 3.1 and 3.2, where the most heavily addressed customer needs for Toothpaste in January 2011 are "Apple" followed by "Lime" while the most heavily addressed needs for Dog Food in December 2021 are "Leek" and "Superfood". Although only shown for two categories in Figures 3.1 and 3.2,

ground truth datasets of this kind are generated for 37 product categories in this chapter.

As stated in Section 2.5.3.1, datasets of this kind (i.e. ranked lists of keyphrases) have also been curated for many other NLP tasks [29, 189–192, 236, 237], although not for predicting future customer needs. They're mainly used to evaluate algorithms for predicting certain tasks e.g. [236] had annotators build a dataset of top-ranked cybersecurity keyphrases to evaluate their algorithms run over forum discussions performing the same task. However, it is of note that they can also be used to train supervised ML models, as in this thesis (detailed in Chapter 5).

It is noteworthy that the time period of 1 month for the datasets in Figures 3.1 and 3.2 is chosen because the algorithms in Chapters 4 and 5 also predict future customer needs at this granularity (hence the dataset needs to reflect this). This time period was picked in accordance with the product development literature, which has frequently made reference to generating ideas at this granularity [248–252].[1] Additionally, it has been noted that service companies like Expedia analyze social media posts each month to be able to gather their customer's needs [253]. Logically, it makes sense to generate needs at this 1-month granularity for most products as shorter time periods may not be necessary (e.g. every day), whereas longer periods may not be sufficient (e.g. every year) for teams developing products.

2011-01	2011-02	...	2021-11	2021-12
1. Apple	1. Recyclable		1. Charcoal	1. Bleeding
2. Lime	2. Organic		2. Coconut	2. Charcoal
⋮	⋮		⋮	⋮
19. Detox	19. Abrasive		19. Cherry	19. Mint
20. Coconut	20. Periodontitis		20. Turmeric	20. Soothing

Figure 3.1. Example ground truth output used to train/evaluate algorithms predicting keyphrases (Toothpaste)

2011-01	2011-02	...	2021-11	2021-12
1. Chicken	1. Dry		1. Vegan	1. Leek
2. Chewable	2. Calcium		2. Grain Free	2. Superfood
⋮	⋮		⋮	⋮
19. Duck	19. Carrot		19. Low Sodium	19. Beef
20. Lamb	20. Vitamin		20. Turkey	20. Wet

Figure 3.2. Example ground truth output used to train/evaluate algorithms predicting keyphrases (Dog Food)

[1] Having algorithms predict customer needs at this 1 month granularity was also largely influenced by the industry project sponsor of this thesis (a large firm that develops products).

3.2 Overview of Mintel GNPD

Mintel Group Ltd is a leading global consumer research and marketing intelligence firm.[2] GNPD, one of the proprietary products it provides, is a large product information database used by industry and academics [79].[3] It catalogs various brands of products from 86 countries over several different CPG product categories (e.g. dog food, cereal, soda, etc.) [79, 254]. New products added to the database are collected by "mystery shoppers" from around the world, who scan shops selling CPG goods [254] adding ≈42,000 products per month.[4] Of specific relevance to the work in this thesis is that for each record in GNPD, Mintel records a textual product description detailing claims and features included in the product which in turn reflect customer needs [79]. Furthermore, each newly registered product added to the database is timestamped (when it was first available for retail), which allows new and emerging customer needs to be tracked accurately over time.

Mintel GNPD is a high-quality source of information suitable for the tracking of customer needs in real products (i.e. applicable for the curation of the ground truth datasets detailed in this chapter). It is described in terms of the following attributes: 1) the description for each product is sufficient (i.e. each description contains multiple customer needs and is of adequate length); 2) the number of products in the database is large enough for accurately tracking needs; and 3) the products in the database represents ones from multiple companies therefore representing needs detected by many product development teams over a large period of time.

The product categories used from Mintel GNPD for the work in this thesis are shown in Table 3.1. In total, 37 product categories are extracted across various categories within CPG. However, as seen in the table, only 15 categories are analyzed in this thesis due to the heavy computational requirements of the algorithms run over Reddit (detailed in Section 5). The reasons for choosing these specific categories for analysis (e.g. popcorn, lip balm, etc.) are detailed later in this chapter (Section 3.6). Furthermore, for 19 of the product categories data is collected from 2007-01-01 to 2021-12-31 while data is collected from 2014-12-01 to 2021-12-31 for the remaining 18.[5] This affects how the models are trained model in Section 5.

Product description text is sufficient Here the text data is shown to be sufficient enough by: 1) displaying a sample entry in GNPD, showing it discusses multiple customer needs; and 2) plotting a distribution of the description length for each product category analyzed, therefore showing that the actual text for each product is large enough for all records in the database.

Previous research has shown the descriptions in Mintel GNPD are a rich source of customer needs e.g. [254] uses GNPD to curate a dataset of product formulations. To illustrate this

[2] https://www.mintel.com/about/ - last accessed 07/06/2024

[3] The work in this thesis has access to GNPD through a project sponsor.

[4] 42,000 is the latest estimate provided directly by Mintel

[5] Due to the agreement the thesis project sponsor has with Mintel only data from 2014-12-01 could be collected for some categories.

Table 3.1. Mintel data collected for each product category

Category	Mintel Date Range	Analyzed with Reddit?	Number of Products	Category	Mintel Date Range	Analyzed with Reddit?	Number of Products
dog food	2007-01-01 to 2021-12-31	Yes	19,743	dishwashing liquid	2007-01-01 to 2021-12-31	No	14,489
eyeliner	2007-01-01 to 2021-12-31	Yes	15,936	eyeshadow	2007-01-01 to 2021-12-31	No	27,057
lip balm	2007-01-01 to 2021-12-31	Yes	22,190	lipstick	2007-01-01 to 2021-12-31	No	44,747
nail polish	2007-01-01 to 2021-12-31	Yes	25,605	mascara	2007-01-01 to 2021-12-31	No	17,387
perfume	2007-01-01 to 2021-12-31	Yes	84,985	shower gel	2007-01-01 to 2021-12-31	No	86,720
shampoo	2007-01-01 to 2021-12-31	Yes	84,806	soap	2007-01-01 to 2021-12-31	No	69,309
toothpaste	2007-01-01 to 2021-12-31	Yes	23,818	toilet roll	2007-01-01 to 2021-12-31	No	12,790
beer	2014-12-01 to 2021-12-31	Yes	29,325	vitamin	2007-01-01 to 2021-12-31	No	53,686
cereal	2014-12-01 to 2021-12-31	Yes	27,985	cheese	2014-12-01 to 2021-12-31	No	53,522
coffee	2014-12-01 to 2021-12-31	Yes	36,804	crackers	2014-12-01 to 2021-12-31	No	18,050
cookie	2014-12-01 to 2021-12-31	Yes	80,253	olive oil	2014-12-01 to 2021-12-31	No	11,828
pizza	2014-12-01 to 2021-12-31	Yes	10,142	pasta	2014-12-01 to 2021-12-31	No	32,195
popcorn	2014-12-01 to 2021-12-31	Yes	6,782	pasta sauce	2014-12-01 to 2021-12-31	No	11,174
soda	2014-12-01 to 2021-12-31	Yes	23,776	potato chips	2014-12-01 to 2021-12-31	No	23,010
soup	2014-12-01 to 2021-12-31	Yes	16,948	rice	2014-12-01 to 2021-12-31	No	19,965
air freshener	2007-01-01 to 2021-12-31	No	28,364	tea	2014-12-01 to 2021-12-31	No	36,489
candle	2007-01-01 to 2021-12-31	No	13,100	wine	2014-12-01 to 2021-12-31	No	12,732
cat food	2007-01-01 to 2021-12-31	No	21,273	yogurt	2014-12-01 to 2021-12-31	No	41,384
deodorant	2007-01-01 to 2021-12-31	No	46,889				

visually, Figure 3.3 shows an example Toothpaste product entry in GNPD.[6] The product description text in Figure 3.3 shows that it is ample in customer needs, mentioning various benefits (e.g. "naturally derived", "cavity protection", "freshness sensation", "recyclable") as well as features associated with the product (e.g. "mint"), thus reflecting customer needs.[7]

Figure 3.4 shows a distribution of the product description character length for all the 37 categories of products that are used in the ground truth datasets curated in this thesis. The distribution indicates the description text is large e.g. mean character length is 417.9. This along with the fact that the text is shown to discuss multiple customer needs demonstrates the product description text is sufficient for the tracking of these needs.

Number of products in database is large

[6] The fields scraped when curating the datasets discussed in this thesis are: i) Category i.e. product type (e.g. Toothpaste); ii) Market i.e. where the product is sold (e.g. Hungary); iii) Date Published i.e. when the product is first in the marketplace (e.g. Jan 2022); and iv) Product Description i.e. the text containing customer needs.

[7] It's not possible to release more examples of product descriptions due to GNPD being a proprietary database

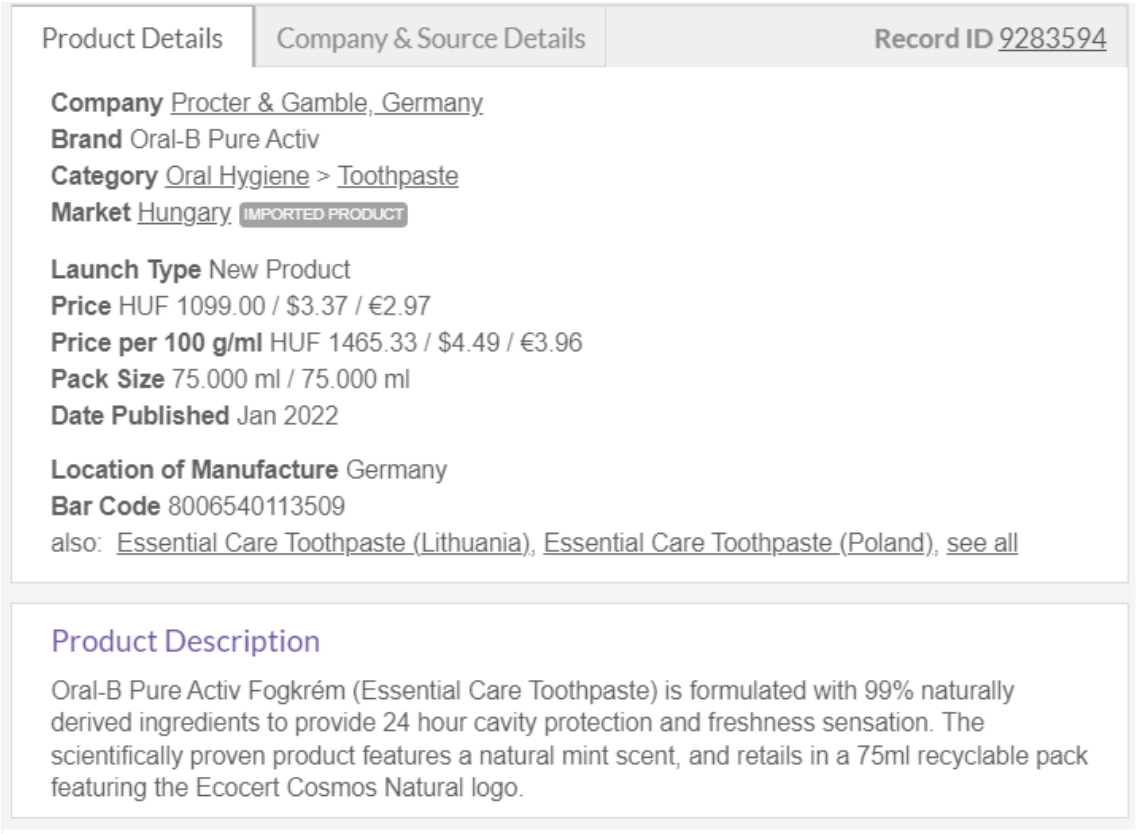

Figure 3.3. Sample Mintel Partial Product Record

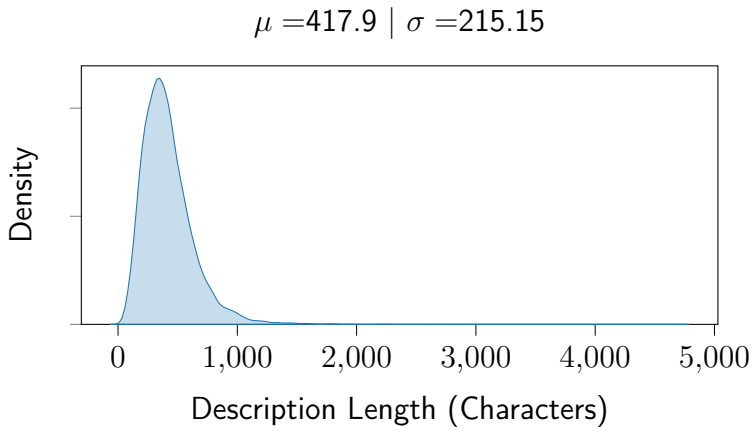

Figure 3.4. Distribution of product description length (in characters) across all of the 37 product categories analyzed.

As seen in Table 3.1, the number of products available for each category is large, ranging from 6,782 (popcorn) to 86,720 (shower gel) between the dates 2007-01-01 to 2021-12-31. This is a sufficient number of products to properly represent the ground truth of top customer needs addressed in products on the marketplace. Furthermore, enough products are coming out each month for every category (see Appendix B), which is required as the ground truth is curated every month (as discussed in Section 3.1).

Category data contains multiple companies

The data provided in Mintel GNPD contains products from multiple companies. To show this, a histogram of the total number of companies in each of the 37 product categories is

plotted in Figure 3.5. As seen in the figure, the number of companies ranges from 1,284 to 13,853 (with a mean of 4,627.78). It is of note that firms address customer needs in products in response to demand by carrying out research into what their customers require. Therefore, by having the data come from multiple companies (meaning multiple product development teams) it represents the thoughts of many professionals coming from different organizations. This is obviously more valuable than the data being represented by a small number of companies.

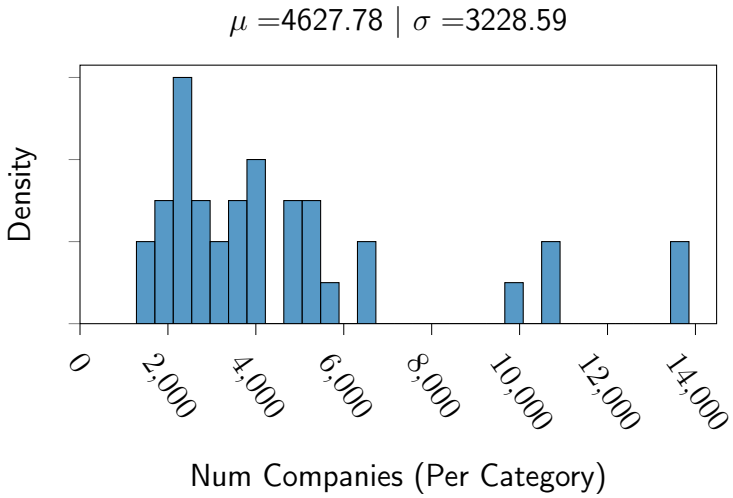

Figure 3.5. Histogram of the number of companies for each category

3.3 NER-T (Named-Entity Recognition based Toothpaste) Customer Needs Dataset

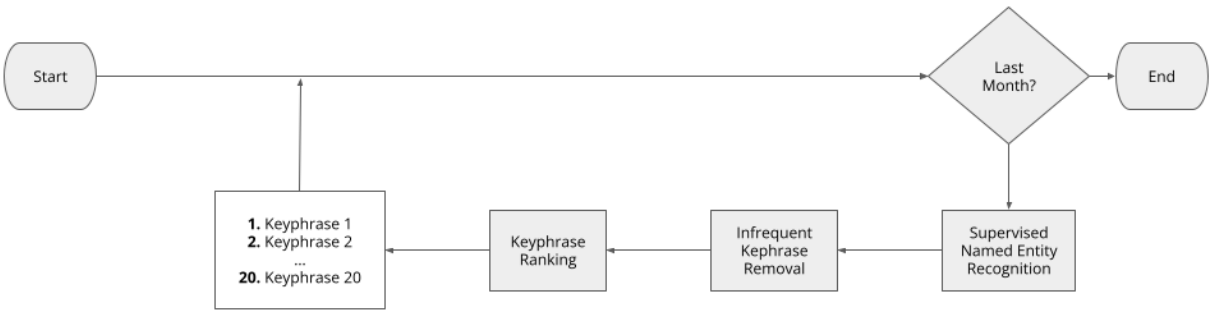

Figure 3.6. NER-T Curation Methodology - identify customer needs using NER then rank them each month

In this section, the methodology of formulating NER-T is detailed. As discussed in Section 3.1, it consists of lists of ranked keyphrases (as shown in Figures 3.1 and 3.2). It is formulated using NER to tag customer needs in product descriptions that are then ranked each month by the extent to which they are addressed in products. It is used to evaluate the algorithm in Chapter 4 that generates lists of future customer need keyphrases for Toothpaste products. Hence, this dataset is only curated for the product category Toothpaste.

To formulate NER-T, the product description data (detailed in Section 3.2) is collected from

Mintel GNPD. The initial set of 23,818 Toothpaste product descriptions (shown in Table 3.1) is filtered to 1,778 descriptions for the curation of this dataset. This is done for two reasons: 1) only descriptions between the dates 01/01/2012 to 31/12/2020 are collected (due to the experimental time period in Chapter 4); and 2) only descriptions from the USA, U.K., Canada and Australia are collected - these nations are selected as Reddit (the data in which the algorithm in Chapter 4 is run over) is mainly comprised of users from these areas, thus resulting in a fairer experiment.[8] It is of note that due to how the curation approach works (explained later in this section), ranked lists of keyphrases representing customer needs are only generated 36 months after the date the product data is collected. Hence, this dataset is made up of ranked lists of customer needs from 01/01/2015 to 1/12/2020 (as the product descriptions data is collected after 01/01/2012).

The curation methodology of NER-T works by applying a NER model to first detect customer need keyphrases from Mintel product descriptions. These keyphrases are then ranked by their frequency from 36 previous months to form a sorted list of ones that are most heavily addressed in products on the market.[9] Figure 3.6 shows the process this curation method uses to generate these ranked lists of customer need keyphrases each month. First customer need keyphrases are identified from product descriptions by using a supervised NER algorithm (Supervised Named Entity Recognition). Here, three sub-steps are performed: a) Annotation; b) Data Augmentation; and c) Run Ensemble of Supervised Models. Secondly, infrequent keyphrases that are not discussed enough are removed as candidate customer needs (Infrequent Keyphrase Removal). Finally, keyphrases are ranked based on the extent to which they are most heavily addressed in products in the month of interest compared to the extent to which they were previously discussed in the last 36 months (Keyphrase Ranking). The output of these processes results in a sorted list of keyphrases that represent the most heavily addressed customer need keyphrases in products each month for a specific product category e.g. the top 20 keyphrases for Toothpaste in March 2017 (as shown in Figures 3.1 and 3.2).

3.3.1 Annotation (Supervised Named Entity Recognition)

To tune the NER model, customer need keyphrases are annotated using a random sample of 90 Toothpaste product descriptions. For this dataset, the definition of "customer needs" includes the benefiting specifications of the product as well as the features or attributes of a product that have benefits associated with them (as discussed in Chapter 1). To follow this definition, customer needs are split into two main categories: a) direct needs - directly stated benefits a user gets/overcomes from using a product (e.g., *"the fresh product is recyclable and helps in curing the user's hay fever"* $\rightarrow$ {"fresh", "recyclable", "curing", "hay fever"}); b) indirect needs - features of a product which contain benefits (e.g., *"the product contains mahogany"* $\rightarrow$ {'mahogany'}). This definition is used as both sources of information are of interest to people who could be using this research (e.g. innovation teams who are interested in upcoming features that could be used in products).

[8] https://www.similarweb.com/website/reddit.com/#overview - last accessed 07/06/2024

[9] The choice for using 36 previous months will be elaborated more throughout this section. In brief, it is used to provide a accurate score in the phrase importance ranking step of the methodology.

Table 3.2. Data Annotation Guidelines For Labelling "Customer Needs" From Product Descriptions

Rule No.	Rule	Example
1	Mainly tag entities as unigrams	The trolley contains a rechargeable battery → {"rechargeable", "battery"}
2	In a negation case, only tag the entity	The shampoo does not contain triclosan → {"triclosan"}
3	Do not tag any of the brand name or title	Tesco Luxury Strawberry FaceMask ... → { }
4	Do not tag additional products that are promoted alongside the product of interest	This shampoo comes free with the product → { }
5	Do not tag entities recommended by/targeted at a group of people	Doctors recommended this product for the elderly → { }
6	Do not tag the color of a product	This brown product ... → { }

Table 3.2 summarizes the main guidelines annotators are given when labelling customer needs from product descriptions. In Rule 1, annotators only label entities as separate unigrams unless the pair/group of words is imperative to the meaning of the entity, e.g. *"the product contains tea tree oil"* → {"tea tree oil"}. This rule is applied because entities need to be grouped together and counted over time after running the NER algorithm. For negation cases (as in Rule 2), annotators only label the entity. This is followed for the same reasons as Rule 1, with many different forms of "does not contain" occurring before the main entity e.g. "free from", "excludes" etc. It could be argued to not tag these entities at all (as they are not in the product), however, as it is explicitly mentioned in the description it is important to do so. In reality, most of these types of entities included needs that would definitely not be a benefit to be included in a product, such as for environmental reasons (e.g. plastics) or health reasons (e.g. added sugar). Finally, rules 3-6 are applied as they are not needs relating to a product itself. Some of these rules could be argued as needs (e.g. color of a product), however, they are made for labelling consistency purposes.

During annotation, one annotator (the author of this thesis) tags customer needs from 90 product descriptions. Similarly to [255], a random sample of 20 of these descriptions is doubly annotated by a second annotator (independent from the study) to verify the gold standard to find the Inter Annotator Agreement (IAA). This annotator has experience with marketing toothpaste products which is specifically useful when labelling customer needs from these products. Although Cohen's Kappa [256] is considered the standard agreement measure for IAA, it is not the most relevant measure for named entity annotation, as pointed out in many previous studies [257–260]. This is because Cohen's Kappa requires the number of negative instances which is unknown for named entities as they are sequences of tokens (no known number of items to consider) [259, 260]. As a solution to this, Cohen's Kappa has been used on the token level which is known to have two drawbacks: 1) annotators look at sequences of tokens for entities, not individual tokens; and 2) the number of "negative" unannotated tokens will be much larger than the "positive" annotated tokens which would overestimate the Kappa score [259, 260]. Due to this, the F1-score (which does not require the number of "negative" unannotated tokens) has been used alongside the token-level Kappa score [259, 260]. Table 3.3 shows the token-level F1 and Cohen's Kappa scores which are calculated for the agreement between the two sets of annotations. The token-

wise scores are similar to those of other accepted NER datasets in the literature [259, 260]. The token-level kappa indicates "almost perfect" agreement [261] (i.e. the score is in the range of 0.81-1), although the token-wise version is overestimated in this case.

Table 3.3. Token-Wise F1 and Cohen's Kappa Annotator Agreement Scores

Token-level F1	Token-level Kappa
0.856	0.824

As discussed, annotators label "direct" and "indirect" customer needs. In the context of toothpaste products, the annotators seemed to mainly label the direct needs into health problems (bleeding, plaque, gingivitis), health claims (whitening, strengthening, fresh) and other non-health related benefits (recyclable, vegan, kosher) while they mainly labelled indirect needs into flavours (mint, berry, cinnamon) and ingredients (charcoal, fluoride, SLS). The distribution of the main POS tags relating to customer needs is shown in Table 3.4 (found using the Python library spaCy). Low-occurring POS tags are not recorded as they are simply used to separate tokens in a tagged entity (e.g. a hyphen with the POS of punctuation in the entity "tea-tree oil"). It is noteworthy that almost all of the tokens are nouns, adjectives, verbs and adverbs.

Table 3.4. Distribution of POS Tags For Customer Needs Using spaCy

Nouns	Adjectives	Verbs	Adverbs
0.557	0.214	0.203	0.026

3.3.2 Data Augmentation (Supervised Named Entity Recognition)

The use of data augmentation has been proven to be effective for many language processing tasks e.g. text classification [262, 263]. More recently, [264] showed that performing simple augmentation methods can significantly increase the accuracy of NER tasks, especially when the number of annotated samples is small. In light of this, the work in this thesis makes use of all 4 methods proposed in [264] to increase the size of the training data as well as the accuracy of the model. Three out of four of these methods are centered around substituting entities with similar ones (e.g. synonyms/entities with the same tag) while the final method deals with shuffling segments of text to augment the data. It is also important to note that each of the augmentation methods has two hyper-parameters: 1) *number_of_additional_generated_instances* per annotated instance; and 2) the random probability of replacing a token for an annotated instance p. When augmenting this data, these values are tuned to 2 (*number_of_additional_generated_instances*) and 0.4 (p). These values are chosen as they represent the average or most common values used in the grid search experiments of [264]. The augmented version of the data along with the original training data (810 samples in total), is then used to train the NER models used to learn the task.

3.3.3 Run Ensemble of Supervised Models (Supervised Named Entity Recognition)

Ensemble learning is an ML technique that uses multiple base ML models instead of only one model to increase prediction performance and stability [265]. Recently, ensemble learning has shown to be effective in NER tasks [266–268], especially when using neural models that can produce varying predictions from one instance of model training to the next due to the element of randomness present in deep learning techniques [266]. In this approach, ensemble learning is used because the NER technique uses a neural model as well as the discussed augmentation technique adding an additional aspect of randomness to it (i.e. the random probability of replacing a token for an annotated instance p). The base NER models used in this approach are from the Python NLP library spaCy. SpaCy uses the neural language processing framework "Embed-Encode-Attend-Predict".[10] In brief, each model works by embedding words into a vector representation (embed), then encoding word vectors into a sentence matrix which allows context to be accounted for (encode), after which the sentence matrix is converted into a single summary vector which allows for a more condensed representation (attend) before finally making a single prediction from the summary vector (predict). In the ensemble, a number of base models are trained which each contain their own version of augmented data based on the original labelled data. The entities classified for each description are then picked based on a majority vote from all of the trained models (as in [266]). In total 19 base models are chosen. An odd number of base models is chosen to allow for the ensemble to always arrive at a majority vote. The number of models picked could have been smaller (e.g. 3 base models), however, this high value is chosen to increase the performance of entity extraction.

The ensemble of supervised models is evaluated using 3-fold cross-validation on the annotated data. On each iteration of cross-validation, each model in the ensemble is run for 5 epochs. To evaluate the entities produced by the algorithm to the gold-standard test set, the same "strict" NER evaluation strategy as in [269] is used (i.e. exact-boundary and type matching).[11] The precision and recall scores of the strict evaluation with 3-fold cross-validation are 91.64% and 95.35% retrospectively (rounded to 2 decimal places). These results show that the model is capable of extracting entities that are representative of customer needs with high performance which is an important part of evidencing the validity of the ground truth data.

After classification, each entity is lemmatized and converted to lowercase. These entities are then counted and grouped according to how often they occurred in the last 36 months, in order to form a *keyphrase:date* $\rightarrow$ *occurrence* accessor (e.g. feather: 2015-01-01$\rightarrow$20). This transformation is performed for the next stages of analysis so that top trending entities can be found for each month of interest.

[10] https://explosion.ai/blog/deep-learning-formula-nlp – last accessed 07/06/2024
[11] https://github.com/davidsbatista/NER-Evaluation – last accessed 07/06/2024

3.3.4 Infrequent Keyphrase Removal

Phrases that don't exceed a minimum document frequency from the recent past are removed as potential customer needs. This is done due to the technique performed in the keyphrase ranking stage of the methodology (i.e. z-score in Section 3.3.5), which can be perceptible to ranking keyphrases highly with a low document frequency. It would be wrong to say a keyphrase is a highly addressed customer need for a product if it occurs infrequently, thus the reason for having this filter. Specifically, phrases that don't exceed a document frequency of 5 in the past 36 months are removed. These values are used because there is an average of $\approx$ 593 products over each of the previous 36-month sliding time windows of data. Thus, if a need didn't occur in at least 5 of these 593 products, it would be difficult to say that it is the most heavily addressed need in products.

3.3.5 Phrase Importance Ranking

The goal of this dataset is to extract customer needs that are trending at a specific point in time. Past approaches to this problem (in the area of topic detection) have tried to find what is currently popular by detecting "bursts" in activity [270, 271] using Kleinberg's burst detection algorithm [272]. Similarly, z-scoring has been used to detect "bursts" in keyphrase activity [273, 274]. In this approach, this measure of a z-score is used to rank the remaining candidate keyphrases. This is computed by taking into account the occurrence of a keyphrase at some number of previous months in the past as well as the current month, computing the z-score and then extracting the z-score for the current month. These phrases are then ranked by their z-score value in the current month. The top number of ranked keyphrases are then used to represent the top customer needs in the month of interest. The previous 36 months are used as input to this process to understand if the current month has an increased frequency. This 36-month buffer is used as it represents a large number of previous months to accurately compare to observe if there is a real increase in frequency in the current month. It is important to note that companies develop new products in response to a perceived customer need in their target market(s). This process cannot be observed, and thus the emergence of new trends in product descriptions is used as a proxy for customer needs.

3.3.6 Summary and Discussion

In this section, the curation of NER-T is detailed. This dataset is made for only one product category i.e. Toothpaste. It is curated by first training a NER model to detect customer needs keyphrases from product descriptions based on human-labelled data from a set of annotation guidelines. These customer need keyphrases are then ranked each month based on how often they appear in product descriptions. This results in lists of ranked customer need keyphrases being produced each month from 2015-01-01 to 2020-12-01 (as illustrated in Figures 3.1 and 3.2).

This dataset allows the tracking of customer needs in real new-to-market Toothpaste prod-

ucts. This is beneficial as it can be used to evaluate algorithms run over social media that predict customer needs (i.e. task addressed in this thesis). However, it is limited in the sense that it is only available for one product category i.e. Toothpaste. Another drawback is that due to the NER model used in its curation process, it doesn't perfectly detect customer needs from product descriptions, with it achieving 91.64% precision and 95.35% recall during evaluation. In the second ground truth dataset (i.e. Section 3.4), these two issues are dealt with by providing a dataset that spans 37 product categories which doesn't use any prediction model in its curation process, thus strengthening the validity of the ground truth.

3.4 TCN (Trending Customer Needs) Dataset

In this section, the methodology of formulating TCN is detailed. As discussed in Section 3.1, it consists of lists of ranked keyphrases (as shown in Figures 3.1 and 3.2). It is formulated using a semi-automated approach in which candidate keyphrases are automatically extracted and ranked from product descriptions, which are then annotated as customer needs by humans. TCN is used to train/evaluate the ML approach described in Chapter 5, which is evaluated across multiple CPG product categories. In total, ground truth lists of keyphrases for 37 product categories make up this dataset.

To create this dataset, the product description data (detailed in Section 3.2) is collected from Mintel GNPD. As seen in Table 3.1, for 19 of the product categories data is collected from 2007-01-01 to 2021-12-31, while data is collected from 2014-12-01 to 2021-12-31 for the remaining 18 product categories. This affects the months in which ground truth lists of keyphrases are available for each product category. Additionally, as with the NER-T dataset (Section 3.3), ground truth lists of keyphrases are only generated 36 months after the date of data collection. Hence, for each product category in this dataset, any data collected from 2007-01-01 to 2021-12-31 has ground truth lists of keyphrases generated from 2010-01-01 to 2021-12-01 (e.g. toothpaste, lip balm, cat food, etc.) while any data collected from 2014-12-31 to 2021-12-31 has ground truth lists generated from 2018-01-01 to 2021-12-01 (e.g. cookie, olive oil, wine, etc.).

TCN is assembled in 3 steps using a semi-automated approach. First, product data is scrapped from Mintel GNPD, as this dataset contains customer need keyphrases as part of real product descriptions. As the scrapping of this data has already been detailed in Section 3.2, it is not described in this section. Second, candidate keyphrases are extracted from product descriptions using techniques from text mining (Section 3.4.1). These keyphrases are then ranked using 36 previous months of data to produce a set of top trending keyphrases for each month, regardless of whether they represent customer needs.[12] Finally, candidate keyphrases are labeled by human annotators to indicate if they are a need or not (Section 3.4.3). By following this process, a list of top trending keyphrases that represent customer needs is obtained for each month for every product category. This semi-automatic process

[12]The choice for using 36 previous months is the same as Section 3.3 i.e. more reliability for the phrase ranking step of the methodology.

of constructing a dataset in this manner (i.e. pre-processing followed by human labeling) is similar to [275], which annotates terms based on whether they are technology-related.

3.4.1 Ranking Trending Keyphrases in the Product Data

Figure 3.7 describes the process of ranking trending keyphrases each month from the product data for a specific product category. This step aims to first remove keyphrases unlikely to represent customer needs before ranking the remaining keyphrases by how often they appear in the month of interest compared to how often they occur in previous months.

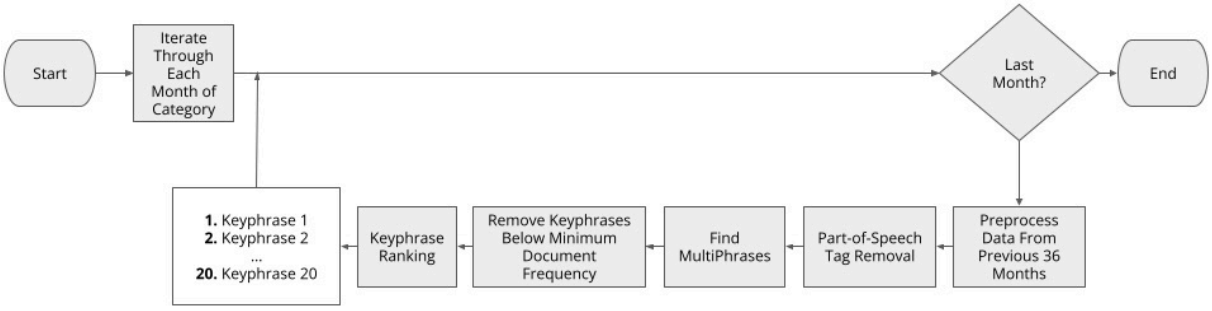

Figure 3.7. TCN Ranking Methodology - keyphrases are ranked each month using techniques from text mining which are later annotated by humans

Iterate Through Each Month per Category: The first step of the approach is to iterate through each month of new products in a category. The start date used changes depending on the product category basis. Specifically, the start month for each category is 36 months after the data was first collected for each category e.g. for "Perfume" the start month is 2010-01-01 because data was first collected in 2007-01-01 (as seen in Table 3.1). 36 months of data are used (as in the NER-T dataset i.e. Section 3.3) to allow for sufficient consideration of category vocabulary to allow for category-calibrated keyphrase rankings.

Preprocess Data From Previous 36 Months: For each month in the product data, data from the past 36 months is extracted. The product descriptions from this data are tokenized and lemmatized using spaCy's en_core_web_lg model.[13]

Part-of-Speech Tag Removed: From the tokenized and lemmatized product description data, tokens are removed based on whether they contain a certain POS tag. The approach is similarly taken by [275] when considering specific phrases to be later annotated by humans. [275] provides a set of defined POS tag combinations that must be conformed to for a keyphrase to be a technological term.[14] In this approach, the same restrictions are not imposed to allow a wider range of phrases with POS tags to be annotated. Instead, only tokens that are not in the following POS tags are removed: i) adverb, ii) verb, iii) adjective, iv) noun, v) proper noun. The following POS tags are used as tokens without these tags are not customer needs according to the annotation guidelines detailed in Section 3.4.3 e.g.

[13] https://spacy.io/models/en#en_core_web_lg - last accessed 07/06/2024

[14] https://github.com/languagerecipes/the-acl-rd-tec/blob/master/misc/pos_sequence_filter - last accessed 07/06/2024

determiners (e.g. the), interjections (e.g. ahem), pronouns (e.g. it) etc.[15]

Find Multiphrases: To extract multi-word phrases from the product descriptions, Gensim's Phrase Model is used with the default parameters to group unigrams as multi-word phrases based on whether they seem to collocate.[16] As in [275], n-grams in the range of 2-5 in length are grouped.

Remove Keyphrases Below Minimum Document Frequency: Keyphrases are removed depending on whether they fall below a set document frequency or raw frequency threshold. As in Section 3.3.4, this step is used to prevent keyphrases with a low document frequency from being ranked highly in the list of keyphrase customer needs. It is also performed to sift through a lot of keyphrases that would be annotated by humans otherwise (as detailed in Section 3.4.3). In this analysis, these thresholds are set at 0.01 (i.e. 1 occurrence in 100 product descriptions) and 18 (i.e. on average at least once every 2 months given that the analysis is over 36 months) retrospectively. This prevents (relatively) unique phrases being annotated; that are not "trending".

Rank Keyphrases: The keyphrases are then ranked to determine whether they are considered to be trending in the given month of interest. This is done by using z-scoring of terms as in the NER-T dataset (i.e. Section 3.3), which also ranks keyphrases coming from Mintel GNPD in this manner. A z-score is computed for a keyphrase for a given month using the standard deviation of its document frequencies in the previous 36 months. These phrases are then ranked by their z-score value in the month of the analysis. As a result, keyphrases are ranked by how often they are mentioned in new-to-market products for a given month. By ordering the keyphrases in this manner, a list of ranked keyphrases for each month is obtained for every product category. At this point, some keyphrases might not be related to customer needs at all and may instead be general phrases e.g. "product". Section 3.4.3 describes how the ranked keyphrases are labeled by human annotators to identify customer needs. These annotations are used to filter the initial list of keyphrases to ones that solely contain customer needs for each product category every month.

3.4.2 Repeat Ranking Trending Keyphrases For Multiple Evaluation Datasets

The process explained in Section 3.4.1 is carried out 4 times to obtain 4 different rankings of keyphrases for each product category every month. To do this, the geolocation of the product information under analysis is restricted (according to its "Market" in Figure 3.3) to find the top customer need keyphrases in different categorizations of regions e.g. only consider customer needs from USA, UK and Australia. This is done to provide fairer evaluation scenarios, e.g. to ensure the ranked lists align to the same regions as specific UGC users. Put simply, as most UGC data is in English, opportunities are offered to remove, for example, German or French products. This results in an evaluation that is

[15] It's noteworthy that the aim of this section is to remove candidate keyphrases unlikely to represent customer needs, hence why this step is performed.

[16] https://radimrehurek.com/gensim/models/phrases.html - last accessed 07/06/2024

more reasonable as customer needs from certain countries may be addressed a long time before/after they are addressed in others e.g. the need "calming" for soap was addressed in products in Asia in 2015 and only in English-speaking countries in 2019.

The 4 different regions analyzed are based on the evaluation of 4 different types of UGC sources: 1) Reddit; 2) English-speaking UGC platforms (like Twitter); 3) UGC platforms where the users are from the USA; and 4) general UGC platforms which consider all regions. For evaluations using Reddit data, only the regions of "United States", "United Kingdom", "Canada", "Australia" and "Germany" are considered, as these regions mainly make up the user base from the platform.[17] The reason why this is done for the platform Reddit and not any other one (e.g. Facebook, Twitter, etc.) is due to the recent increase in the use of its Pushshift API [164], which has led to many recent studies using the platform to mine customer needs [45, 46]. For the evaluation of the English-speaking data, only product data in regions where English is recognized as the primary language according to Wikipedia is considered.[18] This type of evaluation could be most useful when comparing needs on a platform where only English posts are considered or on a platform where English is the main language e.g. Twitter. For the evaluation of the USA only UGC data, only product data coming from the USA is considered. For the evaluation of the general UGC platforms, data from all the regions in the product data is considered i.e. all 86 countries in Mintel GNPD (allows for cases where the Reddit or English-only data is not usable).

3.4.3 Annotating Potential Phrases as Customer Needs

To annotate keyphrases as customer needs across multiple product categories, all ranked keyphrases for each product category are obtained, which is made up of lists of keyphrases generated each month (as described in Section 3.4.1). These keyphrases are then annotated based on whether they are considered customer needs or not by annotators. In this approach, each keyphrase is only annotated once, even if it is mentioned in multiple product categories. This drastically reduces the time required for annotation, as otherwise, every duplicate keyphrase would have to be re-annotated for each of the 37 different product categories. This approach does however come with the drawback that a keyphrase may be a customer need in one category but not in another. For example, in the eyeliner category, the keyphrase "carbon" is an ingredient in the product and is therefore a customer need, however, in other product descriptions "carbon" is referred to when talking about the environment (i.e. "carbon footprint") and is therefore not a customer need. These edge cases are dealt with in the annotation guidelines.

When annotating keyphrases, annotators used the Google Cloud Platform (GCP) text classification tool. When displaying candidate keyphrases for labeling, annotators see a sample of product descriptions in which the keyphrase appeared. An example of this can be seen in Figure 3.8, which is the view the annotators see when labeling. Here, the keyphrase for annotation is "marula oil" and it occurs in two product categories (lip color and eye

[17] https://www.statista.com/statistics/325144/reddit-global-active-user-distribution/ - last accessed 07/06/2024

[18] https://en.wikipedia.org/wiki/List_of_countries_and_territories_where_English_is_an_official_language - last accessed 07/06/2024

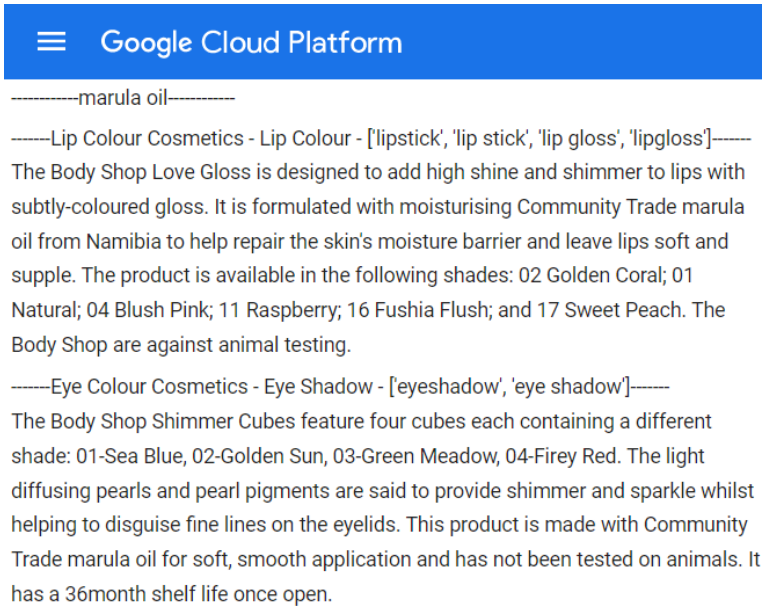

Figure 3.8. Overview of Google Cloud Platform Annotation Framework

shadow). Annotators label the keyphrase based on their domain knowledge and from the provided annotation guidelines. Additional product descriptions are also provided to give more context and aid their judgment when deciding whether a keyphrase is a customer need.

Before labeling, annotators were given a 3-page guideline document detailing definitions and example cases for what the definition of a customer need is and is not. These annotation guidelines are detailed in Appendix C. As in [276], to extend these definitions to contain edge cases of what a customer need is, a pilot annotation study was conducted using Amazon product descriptions from the product categories of Pet Supplies, Office Products and Cell Phones & Accessories [277].[19] In the pilot study, the same keyphrase extraction as discussed in Section 3.7 was carried out and an annotator (independent from the main annotation experiment) labeled 50 keyphrases as customer needs from the initial guideline definitions. After the pilot study, the guidelines were updated concerning any edge cases that occurred that the initial guidelines did not address.

One of the main areas that the guidelines address is in naming and defining the 5 label classes to be annotated: 1) not a customer need; 2) direct need; 3) indirect need; 4) direct-indirect need conflict; and 5) need-not need conflict. The "not a customer need" label accounts for the vast majority of annotations and indicates that the phrase is probably just a general phrase e.g. wonderful, think, fall. As in [22, 23], this study includes the benefiting specifications of products as well as the features or attributes that have benefits associated with them. To abide by this definition, customer needs are split into two label types: a) direct need - directly stated benefits stated as product claims which a user gets/overcomes from using the product (e.g. fresh, antioxidant, bleeding); b) indirect need - actual features or attributes of a product which contain benefits (e.g. mint, charcoal, coconut).

The two remaining label classes were introduced after the pilot study due to conflicts

[19] Amazon product description data can be access under "metadata" - http://deepyeti.ucsd.edu/jianmo/amazon/index.html - last accessed 07/06/2024

between the previous definitions for 1) not a customer need; 2) direct need; and 3) indirect need. These conflicts occur due to a keyphrase having different labels for different categories of products. The first of these conflicts is the "direct-indirect-need conflict" label which occurs when a keyphrase is a direct need in one category but an indirect in another e.g. "lid" is a customer need for "Eye Shadow" products but not for "Yogurt" products. The second conflict is the "need-not-need conflict" which occurs when a keyphrase is either a direct or indirect need in one category but not a customer need in another e.g. "vintage" is a customer need for "Cheese" products but not for "Soap" products. These "need-not-need" conflicts are not used in the final ranking of needs in the dataset. Various edge case details were also added after the pilot study defining what a customer need is in situations where the label class was not clear. These edge cases along with explanations of why they were added are available in the annotation guidelines in Appendix C.

Four annotators were involved in the study. Each annotator had native-level English proficiency and relevant CPG marketing experience, i.e. key domain-specific knowledge for labeling customer needs in product descriptions. The total number of unique keyphrases annotated was 9322 which was split equally between the annotators. In addition, several sets of keyphrases for quality control purposes were included (discussed in Section 3.4.4). To complete the annotations, the annotators met up over 4 days for 3 hours per day, during which time they were completed. During the 3-hour annotation sessions, annotators took 5-minute breaks every hour to reduce the effects of fatigue and mislabelling.

3.4.4 Annotation Quality & TCN Evaluation

To measure and control the quality of the keyphrase annotations, two general measures were put in place: i) IAA and ii) Hidden Gold Standard Questions.

3.4.4.1 Inter Annotator Agreement (IAA)

To measure the IAA, each annotator labeled a set of the same 80 keyphrases. These 80 keyphrases were randomly sampled from the list of all keyphrases and are distributed across many categories. For each annotator, the 80 keyphrases were spread evenly across the 4 days of annotation (20 keyphrases per day) and were also spread evenly across the 3 hours per day when annotating (i.e. 5 IAA keyphrases randomly annotated every 45 minutes). This even spread of the IAA keyphrases was used to reflect the real quality of the annotations when human factors such as annotator fatigue took place.

Table 3.5. Kappa Agreement Between Annotators

	1	**2**	**3**	**4**
1	1			
2	0.603	1		
3	0.642	0.596	1	
4	0.72	0.638	0.564	1

Table 3.5 shows the pairwise kappa agreement between each annotator across the 80 IAA

keyphrases. Annotators received a kappa score that was indicative of moderate (0.41-0.6) to substantial (0.61-0.8) agreement [261], which shows that annotators had a similar execution of labeling keyphrases from the guidelines.

3.4.4.2 Hidden Gold Standard Questions

Hidden gold standard questions have been used in previous studies to measure and control the quality of an annotation task by comparing known correct answers to the annotator's proposed solutions [278]. For this dataset, annotators labeled a set of these 80 gold standard questions in the form of keyphrases throughout the task. These 80 keyphrases were compiled by a committee of two expert annotators independent of the main annotation task who had experience in publishing studies in the area of linguistics and crowdsourcing and had also thoroughly read through the annotation guidelines. The 80 keyphrases used as gold standard questions were annotated at a slower rate by the expert committee compared to the main annotators in the study and therefore are perceived to be of higher quality (1.25 annotations/minute compared to 3 annotations/minute). A keyphrase was only added to the set of gold standard questions when the two expert annotators agreed on the label of the annotation and did not perceive any potential confusion or undue complexity in its annotation.

The gold standard keyphrases were distributed across the annotation sessions in the same manner as the IAA keyphrases. Specifically, each annotator received 20 gold standard questions every day (5 every 45 minutes) that were interspersed with the main annotations. Of additional note is that these keyphrases were evenly distributed by annotation label with 50% of the keyphrases being annotated as "not a customer need" by the experts and 50% of them being annotated as "customer need" (25% direct needs and 25% indirect needs). This was done to ensure that quality was controlled across each label. Thus, annotators labeled ten "not a customer need" keyphrases, five direct need keyphrases, and five indirect need gold standard keyphrases per day.

Each day, the annotators received feedback on the 20 gold standard questions to ensure the same mistakes were not being made across the 4 days. Figure 3.9 shows the accuracy obtained each day by every annotator when compared to the gold standard questions. Although the plot doesn't show perfect agreement by all annotators across each day, it indicates that the annotator's answers were generally aligned well with the gold standard. It is of note that the gold standard annotations on day 3 proved considerably more challenging to annotate (feedback from annotators).

3.4.5 Summary and Discussion

In this section, the curation of the TCN dataset is detailed. This dataset is made for 37 product categories within CPG e.g. cookies, toothpaste, cereal, etc. It is curated by first using text mining techniques to rank keyphrases by the extent to which they are addressed in products for each product category. These keyphrases are then labelled as containing customer needs by 4 annotators, who first read a 3-page guideline document built from

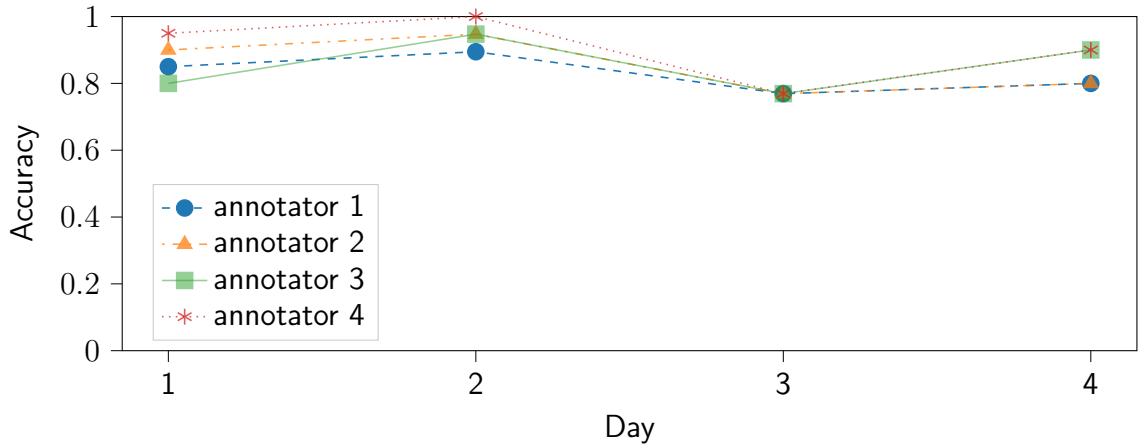

Figure 3.9. Annotators Accuracy on Gold Standard Questions Each Day

an initial pilot study. During evaluation, it was shown that the IAA between the human annotators was indicative of moderate to substantial agreement and that the annotators achieved high accuracy on a hidden gold standard test set built by two independent expert annotators.

As with the NER-T dataset (i.e. Section 3.3), TCN allows the ranking of customer needs in real new-to-market products which is beneficial as it can be used to evaluate algorithms run over social media that predict customer needs. It addresses the drawbacks of the previous dataset by curating for 37 product categories instead of just 1 (i.e. Toothpaste) and not using a prediction model (i.e. NER) in its curation process, which has the drawback of misclassifying a certain percent of keyphrases into the wrong class. By not using an error-based prediction model in the curation process and instead having annotators purely assemble the dataset, this strengthens the evaluation approach described in Chapter 5 that uses the dataset. Because the TCN dataset is made up of multiple product categories, it enables other opportunities to be explored, such as a multi-category evaluation as well as other algorithmic methods. This is seen in Chapter 5, which performs a multi-category evaluation as well as using MTL that uses information from multiple product categories when making predictions (a contribution of this thesis).

In light of these advantages, the TCN dataset has some drawbacks. First, the manual annotation process is much longer than the NER-T dataset. Secondly, the prediction model in the NER-T dataset can be reused to generate customer needs run over new product descriptions and thus can be used in the future. However, the same cannot be said for the approach described in this section, which is a once-off annotation process.

3.5 Evaluation Approaches

This section explains how the described NER-T and TCN ground truth datasets are used to evaluate the keyphrase ranking/classification algorithms run over Reddit. This information may not necessarily be highly relevant to this chapter (i.e. Data), however, since it uses

the ground truth datasets and is used in both the approaches detailed in chapters 4 and 5, it is detailed in this section.

The goal of the evaluation approaches used in this thesis is to observe whether the algorithms run over Reddit can predict customer need keyphrases occurring in the ground truth (i.e. lists of top 20 ranked future customer needs keyphrases as illustrated in figures 3.1 and 3.2) at a future time period. This future time period slightly changes depending on the analysis performed. In the first analysis performed in this thesis (i.e. Chapter 4), this time period is 1-36 months e.g. if the algorithm ran over Reddit predicted on March 2016 it would have to detect keyphrases in the ground truth between April 2016 to March 2019. However, in the second analysis (i.e. Chapter 5), a more strict evaluation is performed by decreasing the future time period to 12-36 months (or 1-3 years) e.g. if the algorithm ran over Reddit predicted on March 2016 it would have to detect keyphrases in the ground truth between March 2017 to March 2019. An upper range for these future time periods is chosen (i.e. 36 months) because it reduces the effect of a random chance correlation of a predicted keyphrase being matched with the ground truth keyphrase on a date far into the future (e.g. 80 months), which brings more validity to the evaluation approach. As stated, the lower range changes in Chapter 4 where it is 1 month and in Chapter 5 where it is 12 months. In Chapter 4, the lowest possible value for the customer need keyphrase to be considered a future customer need is chosen (i.e. 1 month ahead). While in Chapter 5, this value is stricter by stating the future time period must be at least 12 months ahead. The effects of this mean the algorithm will achieve lower results given the evaluation, however, it will be a more reliable result.

Specifically when assessing, two evaluation strategies are used. The first evaluates the model based on the lists of ranked keyphrases it produces each month, therefore assessing its capability of creating lists of keyphrases that it predicts as future customer needs (Section 3.5.1). The second evaluates the model on the instance level in a binary classification setting each month, thus assessing its performance on instances it predicts as future customer needs (Section 3.5.2).

3.5.1 List Evaluation

The list evaluation approach follows a similar methodology to the event detection on social media literature [181, 279], which evaluates topic detection algorithms by submitting lists of topics in fixed time windows (e.g. hourly/daily/monthly windows). These approaches are evaluated by comparing their algorithm's submitted topics to ground truth topics, therefore allowing well-known metrics similar to precision and recall to be calculated. Similarly, this evaluation approach works by comparing lists of keyphrases generated by an algorithm run over Reddit to a ground truth list of keyphrases (such as the ones illustrated in Figure 3.1 and Figure 3.2). Other than this approach being different from the event detection literature [181, 279], which evaluates topics (i.e. lists of words/phrases) rather than keyphrases, it is also different when considering which time windows to match the output of the algorithm run over social media (i.e. Reddit in this thesis) to the ground truth. [181, 279] make comparisons in the same time window to detect events e.g. comparing the algorithm's

topics and ground truth topics both in March 2016. However, as discussed, in this approach the output of the algorithm run over Reddit is compared to the ground truth output between ranges of future monthly dates (i.e. either 1-36 months or 12-36 months ahead). This is done because the algorithms in this thesis aim to predict customer needs in future products and not current ones. When specifically matching customer needs, similarly to [181, 279], keyphrases are considered a match if they are within a Levenshtein similarity of 0.8. This is to allow for some potential misspellings that can occur when comparing the ground truth to the output produced by the algorithms run over social media (i.e. Reddit).

As it is possible to determine if the keyphrases extracted from both the algorithm run over Reddit and the ground truth match, similar metrics to standard ML ones can be calculated (e.g. precision, recall, etc.). Similarly to [181, 279], these metrics are calculated over reduced numbers of produced keyphrases by the algorithm run over Reddit and the ground truth. This is because the keyphrases produced by each are ranked. The number of keyphrases K used to match from both the algorithm and ground truth output is 5, 10, 15 and 20. This is chosen as it is a large enough range to find highly important needs (e.g. top 5 needs) and also needs that are slightly less important but still relevant (e.g. top 20 needs).

Two main metrics are recorded using this evaluation approach: a) *List Mean Precision* and b) *List Recall*. To calculate *List Mean Precision*, List Precision is first calculated. List Precision is calculated each month and is defined as the number of correct keyphrases the algorithm run over Reddit can find in the specified future time period in the ground truth divided by the number of keyphrases K it produces.[20] As List Precision is calculated each month, *List Mean Precision* over all the months used in the analysis can be calculated, which is defined as the average of the List Precision scores. For example, if an analysis is run from January 2015 to December 2017, the *List Mean Precision* score is the average of the List Precision scores (calculated monthly) between these dates. List Recall is defined as the total number of unique keyphrases the algorithm run over Reddit can match in the ground truth data within the future time period divided by the total unique number of keyphrases K the ground truth produces in the entirety of the analysis. For example, if the algorithm can detect 5 unique keyphrases in the specified future time period in an analysis run from January 2015 to December 2017, but, generated 100 unique keyphrases the *List Recall* score is 0.05 or 5%.[21]

3.5.2 Binary Classification Evaluation

For the binary classification evaluation approach, the algorithm is evaluated on the instance level by checking whether the keyphrase instance is in the ground truth dataset (illustrated in Figures 3.1 and 3.2). Here, the definition of being in the ground truth consists of being in the top 20 keyphrases (as discussed in Section 3.1) within the specified future time period specified in the analysis i.e. either 1-36 months (Chapter 4) or 12-36 months (Chapter 5). For example, if the algorithms run over Reddit predicted that the keyphrase "apple" is a

[20] As discussed, this future time period is 1-36 months in Chapter 4 and 12-36 months in Chapter 5

[21] As *List Recall* is calculated over the entirety of the analysis, it's not possible to compute List Recall over multiple time windows as "List Mean Recall" (as with *List Mean Precision*).

future customer need, it would have to appear in the top 20 keyphrases in the ground truth dataset from March 2017 to March 2019 (if the future time period is 12-36 months) for it to be considered detected.

The task addressed in this thesis is a highly imbalanced classification problem i.e. the number of keyphrase instances that are not future customer needs highly outnumber the ones that are. Due to this, the metrics used to assess the problem are carefully considered. The metric chosen for this evaluation approach is F1. The F1 metric is a trade-off between precision and recall, which are both suitable metrics for the evaluation of the task of predicting future customer needs (and imbalanced classification tasks in general [280]). Recall is necessary for the task as it evaluates the number of future customer needs that can be found by the model while precision is needed so to keep the number of false positives low i.e. everything can't be predicted as a future customer need. It is of note that simple accuracy is not used as it is not a suitable metric for these types of problems [280] e.g. a model could predict everything as the majority class and still achieve high accuracy.

3.6 Overview of Reddit

As discussed in Section 2.2, Reddit has been used in many previous customer needs mining tasks [45–47] and is the primary source of social media data used in this thesis. The Pushshift API is used to obtain Reddit data due to its non-restrictive access to information (i.e. historical data and features) and speed of accessibility. Figure 3.10 shows a sample Reddit submission (i.e. post on Reddit) which discusses the Toothpaste customer need "charcoal" before it became popular in real products (as discussed in Section 4.2.3), which is a relevant submission given the context of this thesis. A submission is posted by an author (i.e. teethareyellow in Figure 3.10) on a subreddit (i.e. r/Dentistry in Figure 3.10). It contains a title along with some additional description text and can have likes, dislikes and comments.

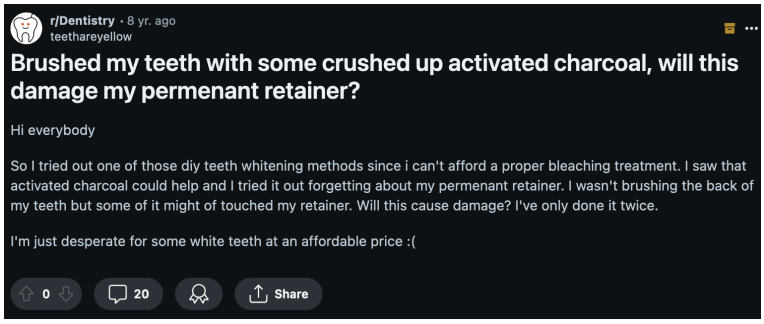

Figure 3.10. Sample submission (i.e. post) on Reddit

Table 3.6 shows the 15 product categories for which data is collected for and analyzed in this thesis for each chapter. Past approaches using Reddit data for mining customer needs have looked at specific subreddits to extract needs for a particular product type [45–47] e.g. r/chairs for chair products. Compared to [16], instead of looking for posts on a specific subreddit, the approaches in this thesis instead search for posts with a Target Keyphrase(s) that the product is associated with (as seen in Table 3.6). When determining

which keyphrases are to be used as search terms, the product category name as a target keyphrase is first included e.g. "beer" for the category Beer. Secondly, any obvious synonyms of the category name are included e.g. "soft drink" for the category Soda. Finally, any potential spelling variations or misspellings of the category name are included as it could account for a substantial number of posts missed on Reddit e.g. "eye liner" for the category Eyeliner.

The majority of the users from the platform Reddit are from the regions "United States", "United Kingdom", "Canada", "Australia" and "Germany" (as discussed in Section 3.4.2). It's noteworthy that this has an impact on the output produced by the algorithms in this thesis as the predicted future customer needs reflect the content posted by people from these regions.

The remainder of this section discusses the Reddit data used in each of the remaining chapters of this thesis i.e. Chapter 4, Chapter 5 and Chapter 6. Different date ranges and numbers of posts are used in each chapter depending on the analysis performed.

Table 3.6. Overview of Reddit data used in thesis. The table shows all the 15 product categories analyzed along with the searched Target Keyphrase(s) on Reddit. It also shows the product category data used for each Chapter along with the date ranges and the total number of posts for each of them.

		Chapter 4		**Chapter 5**		**Chapter 6**	
Product Category	*Target Keyphrases(s)*	*Date Range*	*Number of Posts*	*Date Range*	*Number of Posts*	*Date Range*	*Number of Posts*
Beer	beer	x	x	2015-01-01 to 2018-12-31	239,857	x	x
Cereal	cereal	x	x	2015-01-01 to 2018-12-31	229,239	x	x
Coffee	coffee	x	x	2015-01-01 to 2018-12-31	239,892	x	x
Cookie	cookie, biscuit	x	x	2015-01-01 to 2018-12-31	234,886	x	x
Dog Food	dog food	x	x	2011-01-01 to 2018-12-31	158,847	2020-04-01 to 2023-04-31	160,449
Eyeliner	eyeliner, eye liner	x	x	2011-01-01 to 2018-12-31	346,572	x	x
Lip Balm	lipbalm, lip balm, chapstick	x	x	2011-01-01 to 2018-12-31	193,000	x	x
Nail Polish	nail polish, nail varnish, nail lacquer	x	x	2011-01-01 to 2018-12-31	247,758	x	x
Perfume	perfume, fragrance	x	x	2011-01-01 to 2018-12-31	360,009	x	x
Pizza	pizza	x	x	2015-01-01 to 2018-12-31	239,828	x	x
Popcorn	popcorn	x	x	2015-01-01 to 2018-12-31	227,451	x	x
Shampoo	shampoo	x	x	2011-01-01 to 2018-12-31	415,857	2020-04-01 to 2023-04-31	184,973
Soda	soda, soft drink	x	x	2015-01-01 to 2018-12-31	233,138	x	x
Soup	soup	x	x	2015-01-01 to 2018-12-31	233,001	x	x
Toothpaste	toothpaste	2012-01-01 to 2017-12-31	231,291	2011-01-01 to 2018-12-31	290,180	2020-04-01 to 2023-04-31	184,954

An "x" means the category isn't analyzed for that chapter.

3.6.1 Chapter 4 Data: Toothpaste Customer Needs

The analysis in Chapter 4 looks at extracting future Toothpaste customer needs using a rule-based approach. As the analysis is just performed on Toothpaste, only a Reddit dataset of such posts is collected (as seen in Table 3.6). As the algorithm run in Chapter 4 isn't computationally expensive, all posts containing the searched "Target Keyphrase" on Reddit are collected. The analysis time period is between 2012-01-01 to 2017-12-31, as seen in Table 3.6 (i.e. "Date Range"). The total size of the dataset containing the Target Keyphrase "toothpaste" between the date range is 231,291, coming from 170,065 unique users across 8,303 different subreddits.

3.6.2 Chapter 5 Data: Customer Needs Over Multiple Product Categories

The analysis in Chapter 5 extracts future customer needs across multiple product categories using an ML-based approach. This is done to show that the proposed approach can predict the task on a wide range of categories. In total, Reddit data for 15 product categories is collected (as seen in Table 3.6).

When choosing which product categories to analyze, there are restrictions on which ones can be mined because only the categories collected from Mintel are the ones that can be considered (as shown in Table 3.1) as these are the only ones where ground truth data is obtained. From these 37, some categories have too few Reddit posts to analyze them using statistical techniques e.g. the product category Dishwashing Liquid has 14,161 posts from 2011-01-01 to 2018-12-31 on Reddit with the searched Target Keyphrase(s): "dishwashing liquid", "washing up liquid", "wash up liquid", "dishwasher detergent" and "dishwashing detergent". From the remaining categories in the Mintel data, only 15 of these are identified for the analysis, as a means to reduce computational requirements (which is very high for the approach described in Chapter 5). For the 15 categories shown in Table 3.6, a diverse range of category classes is selected, ranging from the ones stated within Mintel i.e. Health & Beauty (e.g. eyeliner), Pet (e.g. dog food), Food (e.g. cookie) and Drink (e.g. beer).

Additionally, Table 3.6 shows the total number of posts collected for each product category. To keep computational requirements reasonable for the following stages of the approach, the number of posts to analyze for each product category is limited. This is done by randomly sampling posts for each category for which data is obtained. Specifically, disproportionate stratified random sampling at each month (or strata) data is scraped for is employed [281, 282]. That is to say, in the case of this data collection, a maximum number of posts at each month data is sampled, regardless of its size proportional to the total number of posts. This is done to ensure that there is a sufficient number of posts each month for the computational approach described in Chapter 5 to work. Specifically, the maximum sampling rate each month is 5000 posts, even if some of the months don't have this amount of posts e.g. early in 2011 when Reddit uptake was low.

Furthermore, Table 3.6 shows the dates for which the data is available for each product

category. Because ground truth data is only available for some categories from 2018-01-01 (as described in Section 3.4), Reddit data is only collected from 2015-01-01. This is done because 2015-01-01 is 36 months away from 2018-01-01 which is the maximum time in which a trend can be considered detected (as detailed in Section 3.4.4).

3.6.3 Chapter 6 User Study

The analysis in Chapter 6 extracts future customer needs using the ML approach described in Chapter 5. The extracted data is used in a user study that tests whether product development teams from a large MNC find the output produced by the approach in Chapter 5 useful. The analysis is carried out on 3 product categories: Dog food, Shampoo and Toothpaste (as seen in Table 3.6). These categories are chosen as they are all actively developed by the MNC involved in the study.

The study extracts current future customer needs, which at the time of experimentation was on 2023-04-01. Data is collected 36 months before that (i.e. from 2020-04-01 to 2023-04-01), as this is the amount required to run the approach described in Chapter 5. As in Chapter 5, the same disproportionate stratified random sampling approach is used by collecting at most 5000 posts per month (to keep computational requirements reasonable).

3.6.4 Summary and Discussion

This section discusses the Reddit data used in the remaining chapter of this thesis. Reddit data for each product category is collected using searched Target Keyphrase(s) that relates to the category of interest e.g. "cookie" and "biscuit" for the category Cookie. A different number of product categories with different date ranges and number of posts are collected depending on the analysis at hand (as seen in Table 3.6). Chapter 4 collects ≈231,000 posts from 2012-01-01 to 2017-12-31 for just 1 product category i.e. Toothpaste. Chapter 5 collects between ≈193,000 to ≈416,000 for each of the 15 product categories used in the analysis between 2011-01-01 to 2018-12-31. Chapter 6 collects between ≈160,000 to ≈185,000 posts from 2020-04-01 to 2023-04-31 for 3 product categories.

Each of the approaches discussed in this thesis generates lists of predicted future customer need keyphrases each month. For this, an adequate amount of posts each month is required in each of the discussed datasets. Refer to Appendix D for a detailed visual on the exact number of posts collected each month for every product category for each chapter.

3.7 Summary and Discussion

In this chapter, the data used in the remainder of this thesis is discussed. First, a brief overview of the form of the ground truth data is shown with a visual example (Section 3.1).

The output of the ground truth is a ranked list of keyphrases representing customer needs addressed in real products (as shown in Figures 3.1 and 3.2). These lists of keyphrases are used to train and evaluate the algorithms detailed in chapters 4 and 5. Secondly, a review of Mintel GNPD is provided which is the data used when curating the ground truth (Section 3.2). GNPD is a database of new-to-market products with associated product descriptions and timestamps of when the product is available for retail. It has many desirable attributes that can be used to track customer need keyphrases which can then be used to form a ground truth dataset suitable for the training/evaluation of algorithms run over social media. Such attributes include: 1) product description text that discusses customer needs; 2) a large number of products in the database; 3) products coming from multiple companies.

The NER-T dataset is then detailed which curates lists of keyphrase customer needs for just one product category (Section 3.3). It works by first detecting keyphrase customer needs from product descriptions using a trained NER model from human-annotated data. These detected keyphrases are then ranked each month by the extent to which they are addressed in products to form lists of customer need keyphrases each month. The TCN dataset is then detailed, which curates lists of customer need keyphrases across 37 product categories in CPG e.g. toothpaste, cereal, beer, shampoo, etc (Section 3.4). It works by using techniques from text mining to first rank keyphrases each month for every product category. These keyphrases are then labelled in a large-scale annotation exercise. The evaluation approaches used in this thesis are detailed (Section 3.5). Here, two evaluation strategies are detailed which evaluate an algorithm's capability of generating: 1) lists of future customer need keyphrases; 2) predictions of future customer need keyphrases as binary indicators i.e. is or is not a future customer need. Finally, the Reddit data used in each of the remaining chapters is detailed (Section 3.6). To form a corpus for each product category analyzed, posts are obtained from the Pushshift API by using searched Target Keyphrase(s) associated with each category e.g. "cookie" and "biscuit" for the Cookie category.

By showing that 2 ground truth datasets can be curated for the training and evaluation of algorithms predicting future customer needs, RC 3 is proved. This also addresses RQ 2 which questions the stated research contribution i.e. how ground truth datasets can be curated to allow the training/evaluation of approaches predicting future customer needs.

Chapter 4: Rule-based Approach

This chapter describes the first approach in this thesis for predicting future customer needs from social media. It works by producing ranked lists of keyphrases representing future customer needs. There is a lack of such valuable approaches in the literature (as stated in Chapter 2), hence its inclusion.

The approach in this chapter is first detailed, which uses a step-by-step rule-based process to extract keyphrases from Reddit which are then ranked by how likely they are to be addressed as customer needs in future new-to-market products (Section 4.1). Key to the approach is a novel document filtering method (discovering potentially relevant social media content) and a keyphrase ranking method, which promotes keyphrases with rising frequency likely to be future needs. Secondly, the approach is evaluated on the task of identifying future customer needs in the domain of Toothpaste (Section 4.2). Here, the NER-T dataset described in Chapter 3 is used, which consists of lists of Toothpaste customer need keyphrases. Furthermore, an additional evaluation is carried out with a large MNC where it is shown that the output produced by the algorithm detailed in this chapter can capture important customer needs with lead times of up to 25 months in advance of their trending in the marketplace. Finally, a summary of the chapter is provided (Section 4.3).

As detailed in Chapter 1, the goal of this chapter is to address RC 1 which shows that future customer needs can be predicted on Reddit using a rule-based approach. This partially addresses RQ 1, which examines whether future customer needs can be predicted using social media data. During the evaluation in this chapter (Section 4.2), it is shown that future customer needs can be predicted using Reddit effectively. Additionally, it is of note that a large portion of the work in this chapter is taken from [81].

Although the Toothpaste use-case in this chapter may seem an unusual choice for the discovery of customer needs, it is used for two main reasons: 1) there are continually many new ingredients (e.g. charcoal [283]), and benefits (e.g. plant-based, disease prevention [284]), representing customer needs that are constantly being addressed in Toothpaste products, thus making it a suitable test case to investigate whether these needs can be detected on social media before they trend on the marketplace; and 2) the broader area of oral-care is a multi-billion dollar global industry that is still growing, therefore companies creating toothpaste products could benefit greatly from using this research.[1] It is also worthwhile noting that the oral-care sector has already used research to identify needs from UGC through computational and statistical techniques [53].

[1] https://www.statista.com/statistics/326389/global-oral-care-market-size/ - last accessed 07/06/2024

4.1 Methodology

Figure 4.1 outlines the keyphrase prediction problem addressed in this chapter. It is important to note that this is the same problem addressed in Chapter 5. In brief, the proposed approach aims to extract candidate keyphrases from social media data (i.e. Reddit) which predict keyphrases representing future customer needs i.e. in a future time period within the ground truth data. The algorithm makes use of the timestamp associated with each social media post to make predictions at each Fixed Time Window i.e. predict keyphrases as customer needs each month (as in Figure 4.1). For the social media algorithm to make predictions, it considers data from Previous Time Windows. This is seen in Figure 4.1, where the algorithm uses data from 3 Previous Time Windows (i.e. March 2014, April 2014 and May 2014) to produce its final predictions for an individual Fixed Time Window (i.e. June 2014). Data from these Previous Time Windows is required due to the nature of the method being used (requiring past data for its computation). The overall prediction task is then to determine to what extent keyphrases from social media can be used to predict ones that appear in the NER-T ground truth dataset (described in Chapter 3) at a future time period i.e. are future customer needs. This is seen in Figure 4.1 where the keyphrases predicted in the Fixed Time Window by the algorithm run over the social media data (i.e. June 2014) predict the keyphrases in the ground truth dataset for a specific product category (i.e. February 2017, March 2017, April 2017, May 2017 and June 2017).[2]

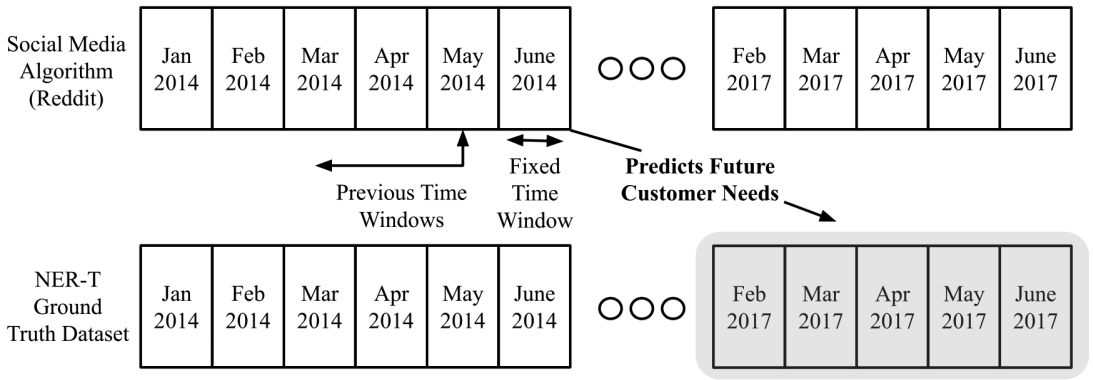

Figure 4.1. Overview of Task: Social Media Predicts Trending Keyphrases in Future Product Descriptions i.e. NER-T

For the experiments in this chapter, the NER-T ground truth is used to evaluate the output of the algorithm run over social media (i.e. Reddit). The algorithm run over the social media data is then optimized to identify the important keyphrases that will appear in the future top keyphrase lists produced by the ground truth product descriptions. The overall task is thus to determine to what extent there is a future time-lagged relationship between the keyphrases produced from the NER-T ground truth dataset and the social media algorithm. This is seen in Figure 4.1 where the keyphrases generated in the Fixed Time Window by the social media algorithm (i.e. June 2014) aim to predict the keyphrases produced 32-36 months in the respective future by the NER-T dataset (i.e. February 2017, March 2017, April 2017, May 2017 and June 2017). This also enables the measurement of how far in advance social media data can predict customer needs identified by important keyphrases

[2]The exact time frame seen in Figure 4.1 is not the one used in the experimental setup but is rather used to illustrate how the task is performed.

(e.g. product features) that will occur in future products. It is important to note that the methodology used to produce keyphrases for the social media algorithm does not require the keyphrases from NER-T, rather they're only used to evaluate it.[3]

Table 4.1 shows the key parameters used in the approach. Some of these have already been discussed (e.g. Fixed Time Window Length) while the rest will be detailed throughout the remainder of this chapter. Each parameter in this chapter is discussed and presented without reference to "good" or "suitable" values, instead, refining the values of many parameters forms a key part of the evaluation (Section 4.2.1).

Table 4.1. Description of Parameters Used in the Methodology For the Product Description and Social Media Algorithms. The following parameters are optimized during the evaluation: "% Most Similar to Gold Standard Subreddit", "Social Media Min Document Frequency" and "Min Chi Square P-value".

Parameter Name	Parameter Type	Description
Fixed Time Window Length	Experimental Setup	The time span in which to produce customer needs
Num. Past Time Windows	Experimental Setup	The number of past time windows of data to use in order to produce needs at each fixed time window
Future Prediction Time	Experimental Setup	The defined window of time the social media algorithm tries to predict needs in future product descriptions
Google Trends Category	Social Media Algorithm	The google trend category which is related to the target product being analyzed
Gold Standard Subreddit	Social Media Algorithm	The subreddit which is related to the product under analysis, which is used by the data reduction approach
% Most Similar to Gold Standard Subreddit	Social Media Algorithm	A parameter value used to control the number of posts used in the analysis by including/excluding posts based on their similarity to the *Gold Standard Subreddit*
Allowed POS Tags	Social Media Algorithm	The allowed part-of-speech tags for a keyphrase to be considered a customer need
Social Media Min Document Frequency	Social Media Algorithm	The min document frequency of a keyphrase in a set of social media posts in order for it to be considered a need
Min Chi Square P-value	Social Media Algorithm	The min chi square value a keyphrase must have when its frequency on Reddit is compared to a reference corpus

Figure 4.2 outlines the algorithm's approach when producing ranked lists of keyphrases for each Fixed Time Window. First, social media data is collected from Reddit which is used as the data for which ranked lists of keyphrases are formulated. Secondly, posts that are irrelevant from the standpoint of mining customer needs are removed (i.e. Data Reduction). Thirdly, candidate keyphrases likely to be customer needs are extracted from the remaining posts (i.e. Keyphrase Extraction). Finally, these keyphrases are ranked to the extent to which they are likely to be future customer needs (i.e. Phrase Importance Ranking From Reddit & Google Trends). This section focuses on providing an overview of the methods used in the approach. The exact libraries used to perform the task along with a specific

[3]This is different in Chapter 5, where the TCN ground truth dataset is used to train and evaluate the algorithm run over social media i.e. using ML.

case study example of Toothpaste customer needs extraction will be discussed in Section 4.2.1.

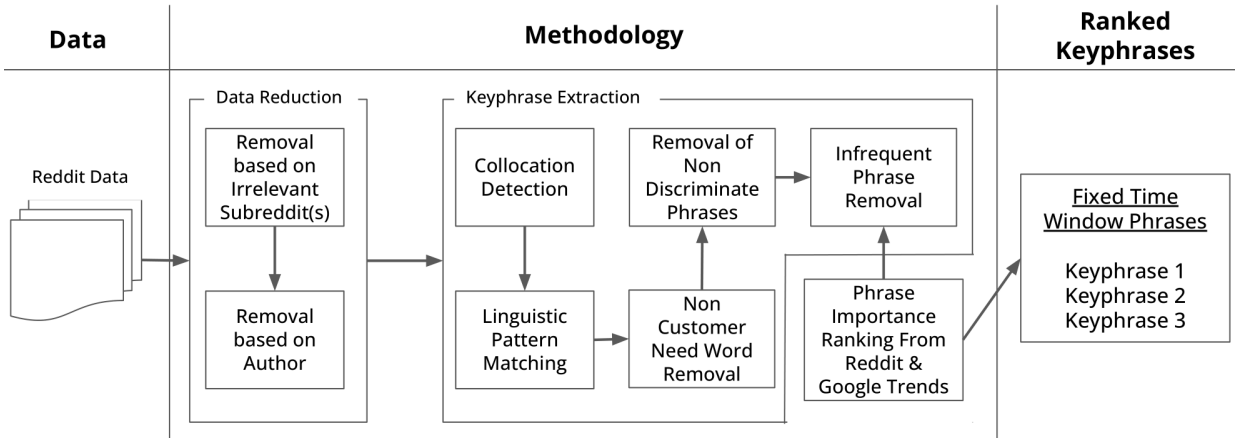

Figure 4.2. Overview of the methodology for the production of ranked lists of keywords from social media (customer need predictions)

4.1.1 Data

Social Media Data: As discussed in Chapter 3, the social media data used in this chapter is from Reddit i.e. 231,291 posts between 2012-01-01 to 2017-12-31 containing the phrase "toothpaste" (as shown in Table 3.6).

Ground Truth Data: The ground truth data used in this chapter is the NER-T dataset described in Section 3.6, which consists of ranked lists of keyphrases representing customer needs. These ranked lists are formulated from a database of new-to-market product descriptions.

Google Trends In addition to the main social media (i.e. Reddit) and ground truth dataset (i.e. NER-T), Google Trends is also used as a data source in this chapter. Recently, it has been used in studies to predict future customer needs [65]. In this chapter, it is used in conjunction with Reddit data when producing a final ranking of customer needs for a given corresponding Fixed Time Window (i.e. "Phrase Importance Ranking From Reddit & Google Trends" in Figure 4.2). In Section 4.2.4, the positive impact of using this ranking approach (i.e. Google Trends and Reddit) is shown to be better than only using Reddit or Google Trends for ranking alone. Google Trends itself returns a time series for a particular search term, indicating the search volume of different queries over time. Given that this time series is representative of a larger group of people (compared to Reddit), it gives a good measure of the importance of a particular customer need at a given Fixed Time Window. Google Trends also allows for the "category" of search to be selected when searching for a keyphrase, thus allowing keyphrases to be searched for concerning a given input category e.g. "grapes" in the "Non-Alcoholic Beverages" category (*Google Trends Category* - Table 4.1).[4] This parameter can be informative when observing how important a need is concerning a product category (e.g. how important is "grapes" in the "Non-Alcoholic Beverages" product

[4] https://github.com/pat310/google-trends-api/wiki/Google-Trends-Categories – last accessed 07/06/2024

category) rather than just the entire body of searches (e.g. how important is "grapes" in all of Google's searches).

4.1.2 Rule-Based Social Media Algorithm

In this section, the rule-based algorithm that extracts ranked lists of keyphrases likely to be customer needs from Reddit is detailed i.e. keyphrases that are contained in the ground truth data at a future date. Specifically, the algorithm performs the following main processes: 1) reduce the number of posts under analysis to a distilled set of ones that are more likely to discuss customer needs relating to a product (Data Reduction); 2) find candidate keyphrases in the posts which are most likely to be about customer needs (Keyphrase Extraction); and 3) ranking keyphrases based on how popular they are expected to be in the future (Phrase Importance Ranking). To do this (as seen in Figure 4.2), the algorithm performs eight steps: 1) Removal of Posts Based on Irrelevant Subreddit(s) (Data Reduction); 2) Removal of Posts based on Author (Data Reduction); 3) Collocation Detection (Keyphrase Extraction); 4) Linguistic Pattern Matching (Keyphrase Extraction); 5) Non Customer Need Word Removal (Keyphrase Extraction); 6) Removal of Non-Discriminate Phrases (Keyphrase Extraction); 7) Infrequent Phrase Removal (Keyphrase Extraction) and 8) Phrase Importance Ranking From Reddit & Google Trends. All the mentioned steps will be discussed throughout the remainder of this section. It is of note that before any of these steps are performed, the standard text-mining techniques of tokenization, lemmatization and lowering (i.e. converting to lowercase) are all performed.

Removal of Posts Based on Irrelevant Subreddit(s) (Data Reduction): With the collection of posts being distilled down to just posts that contain the keyphrase representing some product type (discussed in Section 4.1.1), posts on specific subreddits have a low probability of discussing customer needs relating to a product type. An example of this is on r/Gaming, where users often post about the keyphrase "toothpaste" when using the "toothpaste method" to clean their CDs, however, they rarely discuss toothpaste in terms of needs or features relating to a product.[5] To remove these subreddits, subreddits that are likely to post about customer needs are found by calculating their similarity to a *Gold Standard Subreddit* (Table 4.1). Similarly to how [285] found similar documents for automated essay scoring, this approach finds similar subreddits to the defined *Gold Standard Subreddit*. This works by collapsing all posts from a specific subreddit into one document to arrive at a subreddit-term matrix, which is further transformed by turning it into a tf-idf representation. The closeness of each subreddit (document) is calculated by finding its cosine similarity to the *Gold Standard Subreddit*, which is known to discuss customer needs. The subreddits are then ranked in accordance to how similar they are to the *Gold Standard Subreddit* and the top percentile of subreddits are retained along with the posts they contain while the others are removed (*% Most Similar to Gold Standard Subreddit* - Table 4.1). During evaluation, it is shown how this data reduction approach plays an important step in the methodology by illustrating that it can obtain statistically significant results at finding future customer needs compared to when no reduction is performed (Section 4.2.4).

[5] https://www.wikihow.com/Fix-a-Scratched-CD - last accessed 07/06/2024

Removal of Posts based on Author (Data Reduction) Reddit moderators are removed as these users are primarily there to point out lapses in other users' "reddiquette" [286] (i.e. an etiquette to follow whilst on Reddit), and hence do not discuss needs relating to products. Bots are also detected and later removed as they do not represent content from a real individual and also have a high probability of posting spam content [287, 288].

Linguistic Pattern Matching Various surveys in keyphrase extraction have noted the use of the linguistic properties of words when removing/including candidate phrases [72, 73, 289]. These surveys have pointed to the literature applying conditions on phrases such that they must contain a particular POS tag such as a noun when being considered. Similarly, this approach applies these restrictions by only allowing a candidate keyphrase to be selected if it has the POS tag of noun, adjective, verb or adverb (*Allowed POS Tags* - Table 4.1). These POS tags are chosen as they are the tags that were found to mainly discuss customer needs (discussed in Section 4.2.1).

Non-Customer Need Word Removal Stop words, URLs and curse words (as in [290]) are removed due to having a low likelihood of being related to a customer need.

Removal of Non-Discriminate Phrases As with similar approaches that extract features from review data [291, 292], non-domain-dependent phrases are removed. These phrases are removed as they do not relate to the needs of the product type being searched for. To find these phrases, the chi-square test [293] is used to discover if there is a statistically significant difference between the observed frequency of a phrase on Reddit to its expected frequency according to a large-scale reference corpus. Some commonly used reference corpora include the Brown Corpus [294] (1 million words of American English spanning various genres) and the Lancaster-Oslo-Bergen (LOB) Corpus [295] (a British equivalent of the Brown Corpus also consisting of 1 million words across various genres). Similarly to work in keyword extraction [296–298], the test is computed for each phrase using a 2-by-2 contingency table, in which a test statistic is returned along with its corresponding p-value. If the p-value associated with each phrase is above some set threshold (*Min Chi Square P-value* - Table 4.1), then the phrase is removed.

Infrequent Phrase Removal Phrases that do not exceed a minimum document frequency in the Previous Time Window of interest are removed (*Social Media Min Document Frequency* - Table 4.1). This is done because keyphrases that occur a low number of times are less likely to be future customer needs.

Phrase Importance Ranking From Reddit & Google Trends As the goal of this algorithm is to extract future customer needs, this approach applies a Mann-Kendall (MK) trend test to estimate whether there is an increase in a keyphrase's usage over time. The MK trend test itself has been used in similar applications in analysing trends across several domains e.g. tracking participation trends [299] and analysing trends in scholarly articles [300] and social media [301]. For an individual candidate keyphrase, the approach works by obtaining its time series from Reddit based off of data from its Previous Time Windows. The normalized time series corresponding to the candidate keyphrase is then obtained from Google Trends.[6] The MK trend test is then run on both of these time series (Reddit

[6] https://support.google.com/trends/answer/4365533?hl=en - last accessed 07/06/2024

and Google Trends). The slope values returned from each of these MK trend tests are then added together to get a final ranking value for a keyphrase in a Fixed Time Window. Before the series is inputted into the MK trend test, they are normalized using unit vector normalization. This is so that the slope from each platform is of equal weighting in the final ranking value and so that the slope value reflects the relative increase in a keyphrase rather than just a raw frequency increase. This ranking value therefore represents the keyphrase's increased usage and thus gives a measure of the future importance of the keyphrase. All the keyphrases are then sorted by this final ranking value and the top number of keyphrases are chosen as customer needs which are of future importance. During this chapter's evaluation (Section 4.2.4), the positive impact of the introduction of the Google Trends data and how this ranking approach is performed is detailed (compared to if only the Reddit data is used to rank the keyphrases).

4.2 Evaluation

This section aims to detail the solution concerning an example case study while also seeking to address the RQ in this chapter i.e. can future customer needs be predicted using Reddit? In addition, the positive effect the data reduction approach (i.e. Removal of Posts Based on Irrelevant Subreddit(s) described in Section 4.1.2) has on the task of predicting future customer needs is detailed. To show the effectiveness of the approach, it is evaluated to find future customer needs for Toothpaste products (as discussed in Chapter 3). The reason for picking the domain of Toothpaste was described at the start of this chapter i.e. many new customer needs are addressed in Toothpaste products and the oral-care industry is a multi-billion dollar industry.

This section first details the steps of the methodology applied for finding future customer needs for the defined case study of Toothpaste needs (Section 4.2.1). The rule-based social media algorithm's performance at finding future customer needs addressed in new-to-market products is then evaluated (Section 4.2.2). After, a case study evaluation is presented comparing the social media algorithm's performance on needs obtained from a large MNC (Section 4.2.3). The impact of the steps in the methodology is detailed (Section 4.2.4). Finally, a summary and discussion of the evaluation is given (Section 4.2.5).

4.2.1 Illustrative Example of Methodology: Toothpaste Customer Needs

In this section, the steps of the methodology for finding toothpaste customer needs for the social media algorithm (initially detailed in Section 4.1.2) are described. The ground truth data is formulated each month from 2015-01-01 to 2020-12-01 (as described in Chapter 3) (72 months). The social media algorithm then predicts future customer needs every month from 2015-01-31 to 2017-12-01 (36 months). Both the ground truth and the social media algorithm operate on a *Fixed Time Window Length* (Table 4.1) of 1 month where ranked

lists of customer needs are produced each month. They also both use 36 months (i.e. 3 years) for the *Number of Past Time Windows* (Table 4.1) to produce needs at each *Fixed Time Window* as described in Section 4.1.2 for the social media algorithm and Section 3.3 for the ground truth. The overall task addressed in this chapter is thus to determine the level at which the social media algorithm can predict needs at each window in the future for the ground truth. The specific future time period used in this chapter is 1-36 months (*Future Prediction Time* - Table 4.1), which is the reason why the ground truth generates needs 36 months in the future respective to the time frame the social media algorithm is operating on.[7]

The parameter values for the social media algorithm are mainly tuned based on commonly accepted values applied previously in the literature. The remaining values are found based on an analysis from a grid search experiment (i.e. *% Most Similar to Gold Standard Subreddit*, *Social Media Min Document Frequency* and *Min Chi Square P-value*).

The algorithm uses the processing library spaCy to tokenize, lemmatize and POS tag words. For the removal of posts based on irrelevant subreddits (Removal of Posts Based on Irrelevant Subreddit), the *Gold Standard Subreddit* (Table 4.1) used in the context of finding toothpaste customer needs is r/Dentistry.[8] This subreddit is chosen as it is most likely to discuss "toothpaste" customer needs. To remove bots and moderators (Removal of Posts based on Author), simple regex rules are used to detect authors with the words "mod" or "bot" in their usernames. However, more complex ways of detecting these authors have been proposed in the literature (e.g. bot detection [302]). The "Phrase Detection" model in Gensim is used with the default parameters for both the minimum Normalized Pointwise Mutual Information (NPMI) score and minimum frequency to collocate words together (Collocation Detection).[9] In the literature, it is common to restrict keyphrases that relate to product features (indirect customer needs) to only noun phrases [303–305]. However, since the ground truth contains nouns, adjectives, verbs and adverbs (as shown in Table 3.4), the flexible restriction that any phrase can be a customer need as long as it contains words with these POS tags is applied, thus allowing the *Allowed POS Tags* (Table 4.1) parameter to be any of these POS tags (Linguistic Pattern Matching). Stop words, URLs and curse words are all removed by the algorithm (Non Customer Need Word Removal). As in [306], phrases that are stop words, according to spaCy's list of stop words are removed. URLs are removed using regex rules. Curse or profanity words are removed with the better-profanity library, which uses a simple word list to detect profane words.[10] Finally, when finding non-discriminative phrases (Removal of Non-Discriminate Phrases), the Python library wordfreq (representative of a normal distribution of words) is used to act as the external reference corpus.[11]

For the remaining parameters, an exhaustive grid search is performed, to determine what combinations of hyperparameter values resulted in the best ranking of keyphrases for the social media algorithm i.e. that correspond with top keyphrases in future new-to-market product descriptions (ground truth). The hyper-parameter values used for each parameter

[7] This specific future time period of 1-36 months ahead was previously discussed in Chapter 3.
[8] https://www.reddit.com/r/Dentistry/ - last accessed 07/06/2024
[9] https://radimrehurek.com/gensim/models/phrases.html - last accessed 07/06/2024
[10] https://github.com/snguyenthanh/better_profanity – last accessed 07/06/2024
[11] https://pypi.org/project/wordfreq – last accessed 07/06/2024

in the experiment are recorded in Table 4.2. The table shows the minimum value, the maximum value and the step size from the minimum to the maximum values used for the remaining values (e.g. the *Min Chi Square P-value* parameter uses the values 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1). It is determined that the values used for the *% Most Similar to Gold Standard Subreddit* and the *Social Media Min Document Frequency* are sufficient to find acceptable results (100 and 20 values are used for each parameter retrospectively). For the *Min Chi Square P-value*, these values are searched between as the values of 0.01, 0.05 and 0.1 are all commonly accepted significance values used in the literature [307]. As this grid search is performed on the test set it is understood that this can result in an over-optimized performance which would not necessarily generalise with the same level of performance. To mitigate any such concerns, it is shown how a sequential range of values for each parameter yields similar results with little deviation, to demonstrate that it is not a particular set of parameter values that achieves a high performance. During experimentation, it was found that the parameter value range combinations recorded in the "Range Values" column of Table 4.2 showed "good performance". The results produced from these parameter ranges will be detailed later in this evaluation.

Table 4.2. Grid Search Hyper-Parameter Values

Parameter Name	Min Value	Max Value	Step Size	Range Values
% Most Similar to Gold Standard Subreddit	0.01	1.0	0.01	0.05-0.20
Social Media Min Document Frequency	0.00005	0.00025	0.00001	0.00005-0.00020
Min Chi Square P-value	0.01	0.1	0.01	0.01-0.03

This exhaustive grid search searches over each parameter for the values ranging from the "*Min Value*" column to the "*Max Value*" column with a step size specified by the "*Step Size*" column. The value ranges for each parameter used in the experiments are contained in the "*Range Values*" column.

4.2.2 Experimental Results

This experiment aims to evaluate hypothesis **H** that customer needs produced by the social media algorithm are early indicators of needs as expressed in future new-to-market product descriptions (i.e. ground truth). As stated at the start of this chapter, in the context of toothpaste products, these needs consist of ingredients, flavours, health problems, health claims or other non-related health benefits. It is of note that because the algorithm in this chapter generates lists of keyphrases, only the List Evaluation (detailed in Section 3.5) is used.

The experiment proposes two approaches for finding keyphrases on social media, the approach described in this chapter (A) and a baseline approach (B). As shown in Table 4.2, the approach in this chapter uses a range of consecutive values for three different parameters. This is done to show that a wide range of parameter values can be used for the approach and still achieve good performance rather than any one set of parameter values (as discussed in Section 4.2.1). For the baseline approach, each month the algorithm produces the most frequent lemmatized unigrams and bigrams that occur in that month. As in

Section 4.2.1, the same restriction that the chosen unigrams and bigrams can only contain nouns, adjectives, verbs and adjectives is followed. This is carried out as the entities tagged from the product descriptions within the ground truth only contain these POS tags (as shown in Table 3.4) and thus results in a more fair experimental comparison. This baseline is used to provide context (in the absence of any other available baseline in the literature) for the results described later in this chapter.

This baseline approach is also used to illustrate how difficult it is to find future customer needs and thus achieve any non-zero results in the defined metrics for the List Evaluation. The task itself is very challenging as users rarely discuss needs relating to products on social media. The task thus really tests the algorithm's performance at finding keyphrase customer needs in the noise of many more irrelevant keyphrases i.e. a needle in a haystack problem. What further compounds the difficulty of the task is how the List Evaluation must be carried out to match keyphrase needs i.e. match keyphrase strings within a Levenshtein distance of 0.8 (Section 3.5.1). Moreover, users on social media sometimes don't use the same vocabulary for a customer need as is detected in new-to-market product descriptions i.e. ground truth. For example, the ground truth detecting the toothpaste ingredient "Sodium Lauryl Sulfate" which might correspond to the abbreviated name of "SLS" detected by the social media algorithm. Therefore, keyphrase needs can go undetected if they carry a similar meaning, however, aren't spelled similarly. The specific way in which List Recall is defined in the List Evaluation along with how the social media algorithm works makes it inherently difficult to achieve high results.[12] List Recall only gets increased results for finding future keyphrases at a time where they haven't been detected before by the social media algorithm, thus preferring a diverse set of needs to be produced. However, the social media algorithm produces highly similar keyphrases between adjacent Fixed Time Windows. This is because it works on data from Previous Time Windows (36 months in this experiment) meaning adjacent time windows work on highly similar data subsets. Therefore, the keyphrases it produces are similar. Furthermore, the fact that the social media algorithm only produces customer needs from 2015-2017 while the ground truth produces needs from 2015-2020 further increases the difficulty of the metric i.e. social media algorithm must detect keyphrase needs far into the future.

Table 4.3 shows the List Mean Precision and List Recall results from the List Evaluation for the approach described in this chapter (A) and the baseline approach (B) over increasing values of the number of produced keyphrases K (rounded to 3 decimal places). As the approach is recorded over a range of different parameter values, the table shows the mean results from these parameter values. Thus, the results presented are not stylised (or best-case) results, but rather give a general trend or indication of performance. The List Mean Precision results show that the approach in this chapter is considerably better than the baseline approach, which is to be expected. However, the List Mean Precision in which needs are identified across the parameter values is impressive, with future needs being found with a mean result between 10%-15% across all values of K. The fact that List Mean Precision for this approach gets better for lower values of K shows that there is success in the ranking of these keyphrases i.e. on average the matched keyphrases appear near the

[12] As stated in Section 3.5, List Recall is defined as the total number of unique keyphrases the algorithm run over Reddit is able to match in the ground truth data within the future time period divided by the total unique number of keyphrases K the ground truth produces in the entirety of the analysis.

Table 4.3. List Mean Precision and List Recall Results for the this chapter's approach (A) vs Baseline (B)

	List Mean Precision				List Recall			
	$K = 5$	$K = 10$	$K = 15$	$K = 20$	$K = 5$	$K = 10$	$K = 15$	$K = 20$
A	0.158	0.141	0.118	0.103	0.021	0.024	0.034	0.046
B	0.000	0.000	0.000	0.026	0.000	0.000	0.000	0.008

The approach described in this chapter (A) shows the mean performance when searched within the parameter values in the "Range Values" Column of Table 4.2 (768 records). The Baseline Approach (B) shows the performance when the most frequent lemmatized unigrams and bigrams which occur in a given months are produced.

top of the produced lists. The List Recall results for this approach are also better than the baseline across all values of *K*. A possible reason the List Recall results are not as impressive (compared to the List Mean Precision results) may be because similar results are often produced across adjacent time windows when the Mann-Kendall slope approach is used (Phrase Importance Ranking). This factor may contribute to the low List Recall results in general. That being said, how List Recall is defined in the evaluation makes it very difficult to achieve high results in this metric. As the results in Table 4.3 only show the mean results across a subset of parameter values, the distribution of these results across the chosen parameter values is shown to demonstrate that the results don't deviate much from each other (see Appendix E).

4.2.3 Case Study Evaluation: Toothpaste Customer Needs

Before carrying out any experimentation, a product discovery team from a large undisclosed MNC specializing in the oral-care sector provided a list of the top customer needs from the time period 2015-2020. These needs along with selected keyphrases associated with them (contained in brackets) are: 1) charcoal toothpaste (charcoal, activated charcoal); 2) coconut toothpaste (coconut, coconut oil); 3) enzyme-boosted products (enzyme, enzymatic); 4) bamboo toothbrushes (bamboo); 5) eco-friendly products (eco, eco-cleaner, eco-friendly, biodegradable, sustainable, recyclable, natural, waste, plastic); and 6) vegan-based products (vegan). A retrospective analysis is carried out to observe whether these needs can be detected by the social media algorithm earlier than when they occur in the ground truth. Here, a need from the list is considered to be found if any of its associated keyphrases match the keyphrases produced by the social media algorithm each month e.g. the need "enzyme-boosted" is considered to be found if a keyphrase from the algorithm contains either "enzyme" or "enzymatic".

For an analysis which is carried out between 2015-2017 (36 months), the social media algorithm is tested to observe whether it can detect the needs in the list of the top needs within the hyper-parameter value ranges (i.e. "Range Values" column) recorded in Table 4.2. Table 4.4 goes on to show the total mean number of times these needs are detected (Mean Num Times Detected), the mean first date the need is detected by the social media algorithm (Mean 1st Date Social Media Detected) and the first date the need is contained within the ground truth (1st Date Ground Truth Detected) when the number of produced

Table 4.4. Evaluation of Social Media Algorithm on Top Customer Needs (K=20)

Need	Mean Num. Times Detected	Mean 1st Date Social Media Detected	1st Date Ground Truth Detected
charcoal	25.238	2015-01-31	2017-10-31
coconut	16.357	2015-03-31	2017-08-31
enzyme	0	n/a	2018-07-31
bamboo	0.151	2017-08-31	2017-11-30
eco-friendly	1.368	2015-10-31	2015-02-28
vegan	22.103	2016-02-29	2015-04-30

Columns "Mean Num Times Detected" (0-36 months) and "Mean 1st Date Social Media Detected" (2015-01-31 - 2017-12-31) show the mean performance when searched within the parameter values in the "Range Values" Column of Table 4.2 (768 records)

keyphrases K is 20 for both the ground truth and social media algorithm. As this experiment records over a range of hyper-parameters, the table shows the mean values for the social media algorithm's recorded columns (i.e. "Mean Num Times Detected" and "Mean 1st Date Social Media Detected"). The "Mean Num Times Detected" column is rounded to 3 decimal places.

The precision that the algorithm can detect the needs is good with 4 out of the 6 needs being detected with an average value of at least 1 time in the analysis (charcoal, coconut, eco-friendly and vegan), while 5 of the needs are detected at least once within the defined hyper-parameter range (charcoal, coconut, eco-friendly, vegan and bamboo). The number of times the algorithm can detect some of the needs is also good with charcoal, coconut and vegan being detected on average 25, 16 and 22 times in a total of 36-time windows (i.e. months). Some of the lead times for the needs are impressive with charcoal and coconut having average lead times of 58 and 53 months ahead before they are detected in the ground truth. This realization is significant as it would allow companies to identify needs long before they become mainstream, thus giving them several advantages in the marketplace. The bamboo and enzyme needs are rarely or never detected by the social media algorithm. The need for bamboo may not have often been found as it is a need for toothbrushes rather than toothpaste products (this analysis searches for toothpaste needs). In the case of enzymes, this may have been because users simply didn't discuss this need on social media and in search engines, even if it was identified by large corporations through other methods (e.g. user interviews or questionnaires). As the results in Table 4.4 only show the mean results across a subset of parameter values, the distribution of these results across the chosen parameter values to demonstrate that the results don't deviate much from each other is shown (see Appendix F).

4.2.4 Importance of Steps of the Methodology

In this section, the reasons for certain steps being taken in the rule-based social media algorithm are explained. This is done by detailing why certain methods are used or why specific data sources are employed to inspect the impact they have on the task of predicting future customer needs. Specifically, two steps of the methodology are looked at: a) Removal

Table 4.5. Mean Performance of the Mean Precision and Recall Results Over Multiple Values of *% Most Similar to Gold Standard Subreddit* (%MSGSS) Parameter

% MSGSS	List Mean Precision				List Recall			
	$K = 5$	$K = 10$	$K = 15$	$K = 20$	$K = 5$	$K = 10$	$K = 15$	$K = 20$
0.01 - 0.1	0.148	0.135	0.114	0.104	0.017	0.025	0.040	0.054
0.11 - 0.2	0.161	0.139	0.117	0.102	0.023	0.024	0.034	0.042
0.21 - 0.3	0.149	0.132	0.112	0.101	0.020	0.024	0.030	0.044
0.31 - 0.4	0.160	0.140	0.112	0.104	0.020	0.023	0.029	0.048
0.41 - 0.5	0.150	0.135	0.119	0.104	0.019	0.021	0.033	0.044
0.51 - 0.6	0.147	0.129	0.112	0.102	0.017	0.020	0.029	0.038
0.61 - 0.7	0.148	0.135	0.120	0.109	0.016	0.020	0.029	0.039
0.71 - 0.8	0.151	0.136	0.117	0.106	0.016	0.018	0.028	0.039
0.81 - 0.9	0.146	0.131	0.114	0.101	0.016	0.016	0.028	0.037
0.91 - 0.99	0.145	0.127	0.112	0.100	0.015	0.016	0.028	0.038
1	0.144	0.127	0.112	0.100	0.015	0.016	0.028	0.039

Results show mean performance when searched within complete ranges of the remaining two parameter values in Table 4.2 i.e. *Social Media Min Document Frequency*=0.00005-0.00025 and *Min Chi Square P-value*=0.01-0.1. There mean for each row is calculated based upon 2600 records.

of Posts Based on Irrelevant Subreddit(s), and b) Phrase Importance Ranking From Reddit & Google Trends.

4.2.4.1 Removal of Posts Based on Irrelevant Subreddit(s)

A key parameter in the *Removal of Posts Based on Irrelevant Subreddit(s)* step of the social media algorithm is the *% Most Similar to Gold Standard Subreddit* parameter (Table 4.1). This parameter controls the number of posts to be included/excluded in the remainder of the analysis. To show how this step of the methodology impacts the results, the results are shown across various consecutive ranges of values for this parameter. These results are recorded in Table 4.5 (rounded to 3 decimal places), which uses the same experimental setup and metrics as in Table 4.3. The results are also shown when this approach isn't used and thus no posts are removed from the analysis i.e. *% Most Similar to Gold Standard Subreddit*=1. To observe how this parameter solely impacts the results, the complete ranges for the remaining two grid searched parameters are considered i.e. *Social Media Min Document Frequency*=0.00005-0.00025 and *Min Chi Square P-value*=0.01-0.1. As each row consists of a range of different parameter values, the mean results for each column are recorded (as in Table 4.3).

There seems to generally be an increase in results when this step of the methodology is performed compared to when it isn't. To show this, a set of significance tests is run comparing the results when *% Most Similar to Gold Standard Subreddit* is between 0.01-0.99 (i.e. step is performed) compared to when *% Most Similar to Gold Standard Subreddit*=1 (i.e. step isn't performed). Specifically, the samples of the List Mean Precision and List Recall results generated by both approaches are compared over the same values of produced keyphrases K used in Table 4.5 (i.e. 5, 10, 15 and 20). For each result set, a Mann-Whitney

U Rank Test [308] is run. This is used instead of a t-test because the result data is not normally distributed nor contains the same sample sizes. From the set of tests, the p-values comparing each of the results all have a value less than 0.01. After the tests are run, it is also found that the mean results of all the samples are all greater - in favour of using the approach (i.e. *% Most Similar to Gold Standard Subreddit* between 0.01-0.99). It can therefore be concluded that using the data reduction approach provides a positive statistical difference in performance. This parameter can thus be significant in the process of finding future customer needs and can be seen as a sub-contribution of this thesis.

4.2.4.2 Phrase Importance Ranking (Reddit & Google Trends)

In this section, the method of keyphrase ranking for the social media algorithm is verified using data from both Reddit and Google Trends (as discussed in Section 4.1.2). In the approach described in this chapter, data reduction and keyphrase extraction are performed using data from only Reddit and then the slope values from a regression-fitted MK trend test are considered from both Reddit and Google Trends for a keyphrase to rank it in comparison with other keyphrases. Here, the reason for taking this keyphrase ranking approach is detailed while at the same time illustrating that either one of the data sources on their own can effectively rank the keyphrases and obtain reasonable results. This is done by showing the performance of keyphrase ranking using the slope values from a) both data sources as in the methodology i.e. *Reddit + Both Ranking*; b) only Reddit i.e. *Reddit + Reddit Ranking*; and c) only Google Trends i.e. *Reddit + Google Trends Ranking*.

For each of these ranking methods, the frequency-based time series associated with each keyphrase is normalized using unit vector normalization (as in described Section 4.1.2). This is done because these time series are based on raw keyphrase frequencies, meaning mainly highly frequent keyphrases appear nearer to the top of the final output lists for each Fixed Time Window if no normalization occurs, which would best be avoided. Such highly frequent keyphrases could be ones commonly in Toothpaste posts such as "brush" or "clean". This happens as the method of keyphrase ranking ranks the keyphrases based on the slope value returned from the MK trend test for their retrospective time series. If the time series are between 0-1 (normalized) then keyphrases that show the most relative increase in keyphrase usage are picked. However, if the time series is between 0-keyphrase frequency (no normalization) then keyphrases that show an overall increase in usage are picked (mostly highly frequent time series).

The entire range of hyper-parameters recorded in the grid search experiment is used (Table 4.2) rather than the hyper-parameter value ranges used in the experiment ("Range Values" column in Table 4.2). This is done as the hyper-parameter value ranges used in the experiment are optimized to work on the *"Reddit + Both Ranking"* method. For each of these methods across a range of different hyper-parameter value ranges, the mean performance of the List Mean Precision and List Recall scores are recorded across the same range of number of produced keyphrases K as in Table 4.3. These results are recorded in Table 4.6 (rounded to 3 decimal places).

The Reddit ranking approach performs better than the Google Trends ranking for the List Recall metric due to how it is calculated. In the experiments List Recall is calculated over

Table 4.6. Mean Performance of the List Mean Precision and List Recall Results for Ranking Methods

	Mean Precision				Recall			
	$K = 5$	$K = 10$	$K = 15$	$K = 20$	$K = 5$	$K = 10$	$K = 15$	$K = 20$
Reddit + Both Ranking	0.150	0.134	0.115	0.103	0.018	0.021	0.031	0.042
Reddit + Reddit Ranking	0.088	0.102	0.103	0.100	0.028	0.039	0.058	0.073
Reddit + Google Trends Ranking	0.114	0.104	0.089	0.086	0.010	0.020	0.028	0.030

Results show mean performance when searched within complete ranges of the parameter values in Table 4.2 i.e. *% Most Similar toGold Standard Subreddit*=0.01-1, *Social Media Min Document Frequency*=0.00005-0.00025 and *Min Chi Square P-value*=0.01-0.1. The mean for each row in calculated based upon 26000 records.

the entire time period in which the analysis is done, in which the social media algorithm tries to detect every phrase ever produced by the ground truth. This thus would benefit from a larger unique number of phrases being produced by the social media algorithm, as there is then a higher chance of it being found as a keyphrase produced by the ground truth. It is the volatility in the keyphrase frequencies of the Reddit data that leads to the Reddit ranking approach performing better than the Google Trends ranking approach, obtaining better results in the List Recall metric across every value of *K*. Google Trends ranking on the other hand leads to a better performance compared to the Reddit ranking for some values of *K* for List Mean Precision. This could be because Google Trends represents more people than Reddit and thus the slope value returned for its ranking could be more precise for future customer needs prediction. The reason why this analysis uses the "Both" method of ranking is that it is a good trade-off between the List Mean Precision and List Recall scores produced by the Reddit and Google Trends ranking methods. The "Both" ranking method gets better List Mean Precision on every iteration compared to the Google Trends ranking method and on some iterations, it is close to the List Recall results produced by the Reddit data.

4.2.5 Main Findings

In this section, the steps of the methodology for finding future customer needs in the domain of Toothpaste were detailed (Section 4.2.1). The algorithm's performance at finding future customer needs was then evaluated using the List Evaluation strategy defined in Chapter 3 (Section 4.2.2). Compared to a random classifier it was shown that the algorithm was able to obtain significantly better performance in all baseline tests. Thirdly, the algorithm's performance was evaluated in a real-world scenario by testing whether it can detect 6 of the most important customer needs identified by a large MNC specializing in oral care (Section 4.2.3). Here, it was shown that 5 of these needs were detected ahead of the marketplace by the social media algorithm. Furthermore, 3 of the needs (i.e. charcoal, coconut and vegan) were detected with significant lead times ahead of the marketplace, therefore potentially giving companies an advantage if a system of this kind is used. Finally, the impact of certain steps in the methodology is highlighted for the task of predicting future customer

needs (Section 4.2.4). Specifically, the positive impact of the "Removal of Posts Based on Irrelevant Subreddit(s)" step in the methodology is detailed and the positive effect of ranking phrases using Reddit and Google Trends is shown.

This chapter also partially addressed RQ 1 that future customer needs can be predicted using social media data. This was shown when presenting the results, which showed that needs could be predicted with metrics significantly better than a defined baseline. To assist in the validation of these findings, a large MNC in the oral-care sector was approached to provide a set of the "biggest" historical new product trends in the market. Taking these as a case study, it was shown that the approach was capable of finding such "high-impact" needs, often with large lead times. The following two steps show that the approach addresses RQ 1. It is noteworthy that this RQ is also addressed in Chapter 5 by using ML to predict future customer needs.

It would, however, be easy to argue that with such large amounts of data, anything can be correlated with a set of keywords in product descriptions i.e. ground truth. The following notes a few details of the approach that specifically attempts to hinder such eventualities. Firstly, it uses ranked keywords and this ranking is incorporated into the derivation of evaluation metrics. Secondly, the performance metrics are quite restrictive: it is difficult to score "high". Thirdly, a variety of hyperparameter settings are explored and illustrate that the approach is not sensitive to optimally picked values for these parameters. Finally, the approach to data reduction seeks to eject arbitrary (but still potentially relevant) data from consideration.

4.3 Summary and Discussion

As pointed out in Chapter 2, there is a lack of approaches in the literature which predict customer needs in lists of keyphrases. Therefore, this chapter outlines a rule-based approach to the problem by using social media data from Reddit to predict keyphrases in Fixed Time Windows which appear as trending needs in future product descriptions i.e. ground truth. Needs were found for a given product type by only considering posts that contained the presence of a predefined keyword and were also included in particular subreddits that were likely to discuss topics relating to the needs of the product type. To extract keyphrases from social media, various techniques from keyphrase extraction were proposed. These keyphrases were then ranked using knowledge from UGC (Reddit and Google Trends) and applying a Mann-Kendall trend test in order to discover whether a keyphrase's frequency was increasing over time. The task was then to observe whether the keyphrases produced by the algorithm could predict the top keyphrases in the ground truth (i.e. needs addressed in real market products) at a future time period This therefore answers whether there is a future time-lagged relationship between the needs generated by the algorithm and the ground truth. To evaluate the approach, the domain of "toothpaste needs" was studied. It was shown that social media could detect needs occurring in future toothpaste product descriptions (i.e. ground truth) with significantly better precision and recall results than a defined baseline. In addition, it was able to detect 3 of these needs with significant

lead times ahead of the marketplace. The impact of this work is that it is evidenced that discussions on social media may give companies significant margins to outpace their competitors in new product development. Even doing so by a number of weeks, can be significant in terms of potential profit and market acquisition.

In light of the following, there are also various limitations of this approach. During the evaluation, the methodology was only applied to the oral-care sector. Thus, an obvious avenue for future work is to explore other product types (e.g. smartphones). As the techniques used in this approach were based on general data analysis it would be assumed that it would have all the right indicators of generalizing to new product types, however as experiments were not conducted, it is a limitation of the study. The approach in Chapter 5 addresses this concern by being applied across multiple CPG product types e.g. toothpaste, popcorn, wine, shampoo etc. Also, with many previous studies employing some knowledge of sentiment when extracting customer needs [24–30, 32, 34, 45–47, 54, 64, 211–213, 309–311], it would be interesting to observe whether it is a predictive factor in discovering needs in future product descriptions. This problem is addressed in Chapter 5, with sentiment being used as a feature in an ML-based approach. Finally, this approach has a limitation in the fact that customer needs are being predicted retrospectively. Careful consideration is taken in the treatment of future data as a ground truth when performing this analysis so that future information does not bias the analysis. The results materializing from this experiment could thus have similar criticisms to that of predicting election results post facto [312]. That being said, this analysis is of interest nonetheless as it was able to correlate past customer needs on social media to future needs in ground truth product descriptions (which is a novel contribution of this work).

Chapter 5: Machine Learning Approach

This chapter describes the ML approach used in this thesis for predicting future customer needs from social media. It works by framing the prediction task of finding future customer need keyphrases as a binary classification task. As stated in Chapter 4, it is included because there is a lack of such valuable approaches in the literature.

First, the methodology of the approach in this chapter is described which uses supervised ML to predict keyphrases as being future customer needs (Section 5.1). Here, keyphrases are extracted from Reddit posts and predicted as future customer needs based on 10 families of features generated for this specific task. A MTSC model is then built on the features that try to predict whether the keyphrase will appear as a trending need in future new-to-market products. During model building, an approach that incorporates MTL by being trained on multiple product categories (e.g. toothpaste, cereal, and beer) rather than just one category (e.g. toothpaste) is also proposed. Secondly, the approach is evaluated across 15 CPG product categories (Section 5.2). Here, the second ground truth dataset described in Chapter 3 is used i.e. TCN. The impact MTL has on the task is also assessed. Additionally, a further examination of the results is carried out to see where certain optimizations can be made to improve the model's capability. Finally, a summary of the chapter is provided (Section 5.3). Here, directions for future work are also discussed.

As detailed in Chapter 1, the goal of this chapter is to evaluate RC 2 (which explores if future customer needs can be predicted on Reddit using ML) and RC 4 (which uses MTL to learn what a general future need looks like across multiple categories). Respectively, these contributions address RQ 1 and RQ 3 which question whether future customer needs can be detected using social media and whether the use of MTL can be employed to train a generalizable model for which future customer needs can be predicted. During the evaluation (Section 5.2), these questions are addressed.

It is of note that a large portion of the work in this chapter is taken from [82]. Additionally, the following GitHub repository details the resources that are mentioned throughout this chapter https://github.com/davidkilroy/Multi-Task-Future-Customer-Needs-Model.

5.1 Methodology

The problem addressed in this chapter is similar to the one outlined in Chapter 4 i.e. Figure 4.1 (Section 4.1). Similarly, it predicts keyphrases in Fixed Time Windows using data from a number of time windows in the past (i.e. Previous Time Windows). That being said, the approach in this chapter differs in two main aspects. Firstly, it uses a different ground truth dataset (i.e. the TCN dataset), which consists of 37 product categories within CPG,

as discussed in Chapter 3. Secondly, the ground truth is used in the training and evaluation of the model, whereas in the rule-based approach (Chapter 4), it is just used during the evaluation. It is used in the training of the model in this chapter as this approach uses an ML model that requires past data to make inferences, while the rule-based approach doesn't require past data to make inferences.

As stated, during experimentation, for each product category analyzed in this chapter, the ground truth dataset is used to train and evaluate a supervised keyphrase extraction model run over social media to allow it to identify future customer needs. This is done by framing the problem as a binary classification task by checking whether a keyphrase on social media appears in the ground truth dataset in some future time period for each product category in the analysis i.e. keyphrase does/doesn't appear in the ground truth in the future. The significance of this is that the ground truth contains top keyphrases addressed in products (as detailed in Chapter 3) and therefore by predicting what will occur in it, this approach effectively forecasts what new customer needs will be addressed in future products e.g. predict the top needs for breakfast cereal items.

Figure 5.1 outlines the algorithm's approach when predicting future customer need keyphrases at each Fixed Time Window on social media for a specified product category. First social media data is collected from Reddit (e.g. collect a corpus of posts for the category "soup"). Secondly, the posts are cleaned using text preprocessing techniques (e.g. tokenization, lemmatization, etc.) and candidate keyphrases are selected for the classification task. Thirdly, a large number of univariate time series are generated for each candidate keyphrase for the MTSC task. These series range from linguistic-based (e.g. dependency or part-of-speech tags) to sentiment-based information series. Finally, a MTSC model is trained on the univariate time series associated with the candidate keyphrases which contain the binary label determined by whether it appears in the TCN ground truth dataset in the future i.e. is a future customer need. This general framework of classifying keyphrases from social media using MTSC techniques has been addressed before, with [194] generating similar families of features to the approach in this chapter (e.g. sentiment and content-based) to distinguish between "organic" and "promoted" hashtags on Twitter. A lot of the core concepts employed in the methodology of [194] are also used here, especially when generating univariate time series for each candidate keyphrase (or hashtag in their study). However, the types of univariate time series generated in this chapter are tailored for the task of finding future customer needs e.g. time series encoding whether a keyphrase occurs in posts discussing products to help the model find keyphrases that are customer needs.

All of the numbered components in Figure 5.1 make up the subsections of this section with the addition of a further section to explain the types of univariate time series created to solve the classification task: 1) Data Collection - scraping Reddit data for each product category analyzed (Section 5.1.1); 2) Text Preprocessing & Keyphrase Selection - preprocessing posts and selecting candidate keyphrases (Section 5.1.2); 3) Generate Multiple Time Series For Each Keyphrase - discussing how time series are generated for each keyphrase for the classification task (Section 5.1.3); 4) Families of Univariate Time Series Generated - examining the families of univariate time series computed for the task of identifying future customer needs (Section 5.1.4); and 5) Time Series Classification - detailing how the ground truth label is added in the binary classification set-up and reviewing the MTSC algorithm

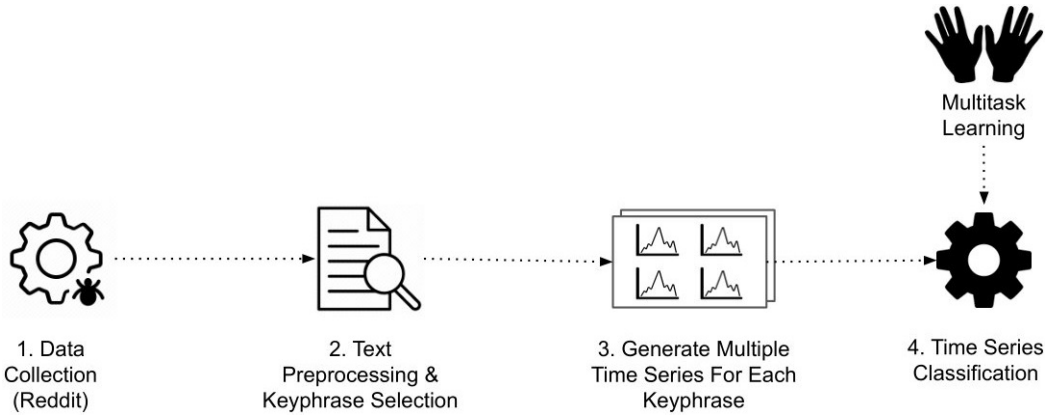

Figure 5.1. Overview of methodology used to predict keyphrases representing future customer needs

applied (Section 5.1.5). As shown in Figure 5.1, the use of MTL is also detailed which is one of the contributions of this thesis (detailed in Section 5.1.5). Here, the approach of building a single model capable of accurately predicting future customer needs in any product category is detailed.

5.1.1 Data Used

Social Media Data: As discussed in Chapter 3, the social media data used in this chapter is from Reddit. As seen in Table 3.6, it represents 15 product categories in the area of CPG from 2011-01-01 to 2018-12-31.

Ground Truth Data: The ground truth data used in this chapter is curated from the second dataset described in Chapter 3 (i.e. TCN dataset), which consists of ranked lists of keyphrases representing customer needs. These ranked lists are formulated from a database of new-to-market product descriptions.

5.1.2 Text Preprocessing & Keyphrase Selection

For each product category analyzed, text preprocessing and keyphrase selection techniques are applied to automatically choose keyphrases from the social media post data - a prerequisite to perform the future customer need keyphrase classification task. For all the main preprocessing tasks implemented in this section, the Python library spaCy is used [313] i.e. sentence boundary detection, lemmatization and POS tagging. Specifically, the *en_core_web_lg* model from spaCy trained on OntoNotes 5 [314] is used to perform these tasks, which achieves high performance across many general NLP problems.[1]

Figure 5.2 outlines the steps performed to select candidate keyphrases from social media posts. The first step carried out is sentence boundary detection [315] i.e. splitting a post into an array of sentences. This is done as only the sentence where the searched "Target Keyphrase(s)" is extracted (discussed in Chapter 3) is mentioned e.g. when mining for the

[1] https://spacy.io/models/en#en_core_web_lg - last accessed 07/06/2024

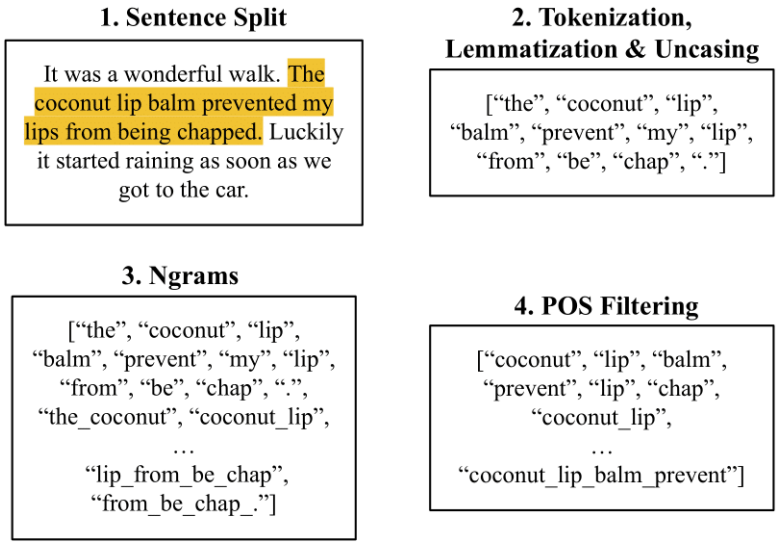

Figure 5.2. Overview of Text Preprocessing & Keyphrase Selection in order to extract candidate keyphrases

Lip Balm product category only sentences containing "lipbalm", "lip balm" or "chapstick" are searched for (as in Table 3.6). This is done as posts on Reddit can be quite large, with much of the discussion unrelated to the product category being analyzed.

Secondly, the sentence is tokenized, lemmatized and uncased, as performed in many other studies using keyphrase extraction [316–318]. Tokenization is needed as it is the first step required to separate candidate keyphrases for the classification task, while lemmatization and uncasing are carried out to group inflected phrases together.

Thirdly, multi-word phrases are formed by taking the set of all possible consecutive n-grams in the range of 1 to 4 grams [319, 320]. This process is seen in Figure 5.2 by forming multi-word phrases from unigrams e.g. "coconut_lip" from the consecutive words of "coconut" and "lip".

Finally, only n-grams with specific POS tag combinations are kept for the next stages of the analysis, as in [275, 321]. For the task, tag combinations that correspond to customer needs are required. To do this, tag combinations of phrases that are already labelled as customer needs are extracted from the TCN dataset i.e. ground truth. Specifically, all the keyphrases recorded across 5 product categories in the TCN dataset are extracted (Table 3.1) i.e. Vitamins & Dietary Supplements, Cat Food, Pasta Sauce, Tea and Potato Snacks. It is noteworthy that these categories are not used in the primary analysis (i.e. in Table 3.6) to avoid any potential bias in the experimental evaluation. They are also diverse in category classes including Health & Beauty (i.e. Vitamins & Dietary Supplements), Pet (i.e. Cat Food), Food (i.e. Pasta Sauce and Potato Snacks) and Drink (i.e. Tea). This diversity is necessary as POS tag combinations associated with keyphrase customer needs are different across category class types e.g. the POS tag combinations in Health & Beauty are different to Food tag combinations. To extract these tag combinations from the keyphrases in the TCN dataset, the same *en_core_web_lg* model is run over them to identify their POS. In total, 31 tag combinations are identified from the 5 product categories. These are made up of single-word combinations (e.g. nouns like chicken or adjectives like energetic) as well as multi-word combinations (e.g. adjective phrases like micro-cleaning). All the POS tag

combinations identified are contained within the single POS tags of nouns, verbs, adjectives, adverbs or proper nouns. This is expected as the ground truth dataset instructs annotators to only label customer needs with these POS tags (i.e. Section 3.4.3). A complete list of these POS tags can be found in the GitHub repository recorded at the start of this chapter. It is important to note that there is similar work for generating task-specific POS tag combinations for other keyphrase extraction tasks e.g. [275] generates a list of tags for the extraction of computational linguistic terms.

5.1.3 Generating Multiple Time Series For Each Keyphrase

This section details how the collection of preprocessed posts for each product category (discussed in Section 5.1.2) is transformed into a form suitable for the prediction of keyphrases using techniques from MTSC. Figure 5.3 shows an example output of the data produced. As with classical ML, a number of features for each candidate keyphrase is generated. However, the value in the fields generated for each feature is not an individual number but a univariate time series. For each candidate keyphrase instance, each of these univariate series makes a multivariate series (required for the task of predicting future customer needs using MTSC techniques). In this section, the process of going from a collection of preprocessed posts (i.e. output of Section 5.1.2) to multivariate time series data for each keyphrase (in Figure 5.3) is detailed. In the next sections, the types of time series features generated (i.e. Section 5.1.4) are discussed before detailing how the ground truth label from the TCN dataset is added to each candidate keyphrase instance along with the MTSC techniques used to classify them (i.e. Section 5.1.5).

Keyphrase	Feature 1	Feature 2	...	Feature 1000
charcoal	[0.02 ... 0.05]	[0.6 ... 0.8]	...	[1 ... 1]
bread	[0.02 ... 0.01]	[0.3 ... 0.2]	...	[0.85 ... 0.9]
milk	[0.05 ... 0.06]	[0.7 ... 0.9]	...	[1 ... 0.95]
orange	[0.001 ... 0.0002	[0.3 ... 0.25]	...	[0.9 ... 0.95]

Figure 5.3. Example output of data suitable for Multivariate Time Series Classification

Figure 5.4 gives a top-down view of the transformations performed in order to move from a collection of preprocessed posts to a set of candidate keyphrases with multiple associated features in the form of univariate time series. Each of the steps in the figure will be described in this section. At a high level, the following steps are performed: 1) Add Additional Features - text-based models are run over the post data e.g. running a text classification "Buy Intent" model from HuggingFace over posts; 2) Group Keyphrases & Summarize Features - features are summarized for each candidate keyphrase at each Fixed Time Window (i.e. month) e.g. for the keyphrase charcoal calculate the mean "Buy Intent"; and 3) Turn Into Time Series - for each keyphrase at a given Fixed Time Window (i.e. month) the features are turned into individual univariate time series by obtaining the values each month for the keyphrase of interest 36 months into the past (i.e. number of Previous Time Windows) and sorting them by time e.g. for charcoal on the 2014-01-01 find the "Mean Purchase Intent"

each month from 2011-01-01 to 2014-01-01. As previously discussed, this entire generation process is performed the same way as [194] by constructing time series for keyphrases (or hashtags in their case) based on calculating summary statistics from the posts it occurs in at each Fixed Time Window. However, different time series are generated in this chapter to reflect the task of recognizing future customer needs (discussed in Section 5.1.4), which is different from classifying between "organic" and "promoted" hashtags [194].

The first step of the approach is to generate a number of additional features about the post or candidate keyphrases of interest in the post. When calculating features about the post, various text-based models are applied to the sentence with the target keyphrase. As seen in Figure 5.4, the models chosen may record document-level information such as Buy Intent from the library Hugging Face - distinguishing between posts containing "buy intent". These features are included as there is a justification for them improving the task of predicting future customer needs. A detailed list of all the features used along with reasons why they are added is addressed in the next section (Section 5.1.4).

In total, summary statistics for four different types of features are calculated (three of which are data types): 1) continuous features, 2) boolean features, 3) string features and 4) keyphrase-level features. Continuous features are columns where the fields contain numerical values (e.g. 4, 5.1, etc). For these features the following summary statistics are computed a) mean, b) maximum, c) minimum and d) sum. Boolean features are columns where the fields contain True or False values. For these features, only the percentage of posts that are True is computed as a summary statistic of the feature column. True is only recorded, as False can be inferred from it, and hence does not add any additional information to the ML model. In some situations, however, False is recorded as well as the value Not a Number (NaN) appears in some fields provided by the Reddit Pushshift API (e.g. "is_video" field). String features are columns where the fields contain a sequence of characters e.g. "submission" for the "Post type (sub/com)" in Figure 5.4. For string features, type matching of a user-defined string is carried out and the percent of posts that contain the matched string as a summary statistic is reported e.g. percent of posts that are "submissions" based on the "Post type (sub/com)" feature (as in Figure 5.4). The process for choosing the types of strings to search for changes depending on the feature summarized. In some cases, it is an exhaustive list of all possible strings in the string-based feature column (e.g. "submission" and "comment" for the "Post type (sub/com)" feature) and in other cases only specific strings are searched for due to their being too many values in the feature (e.g. subreddit feature column). Keyphrase-level statistics are obtained by retrieving an attribute from the keyphrase, therefore accounting for just one summarized statistic. As seen in Figure 5.4, an example of a keyphrase-level feature is the Document Frequency at a given Fixed Time Window. Another example may include a dimension of a word embedding for a keyphrase. Table 5.1 outlines the discussed summary statistics computed in this study. It is important to note that it is possible to include additional statistics (e.g. median for continuous types), however, for computational reasons it was decided to keep this number low. The current statistics are just put in place in order to test whether the general framework works i.e. to see whether it can learn what future customer need trends look like.

The third and final step of the approach is to turn each summary statistic feature into an

Date	Post	Post Preprocessed	Post Type (sub/ com)	Score Attr- ibute
2011- 01-29	"...my homemade charcoal toothpaste.."	["homemade", "charcoal", "toothpaste" ... "charcoal_ toothpaste"]	sub	5
2015- 08-14	"...coconut toothpaste from.."	["coconut", "toothpaste", "coconut_ toothpaste"]	com	-4

Step 1: Add Additional Features

Date	Post	Post Preprocessed	Post Type (sub/ com)	Score Attr- ibute	Buy Intent - Hug Face	Doc Embed Dim 1 - sbert	Doc Len - spaCy
2011- 01-29	"...my homemade charcoal toothpaste.."	["homemade", "charcoal", "toothpaste" ... "charcoal_ toothpaste"]	sub	5	0.8	0.45455	25
2015- 08-14	"...coconut toothpaste from.."	["coconut", "toothpaste", "coconut_ toothpaste"]	com	-4	0.8	0.2342	50

Step 2: Group Keyphrases & Summarize Features

Date	Keyphrase	Doc Freq	Sub %	Mean Buy Intent	Max Buy Intent	Min Buy Intent	Sum Buy Intent
2011- 01-01	charcoal	0.02	0.6	0.7	1	0	10
2011- 01-01	bread	0.02	0.3	0.2	0.85	0	2
2018- 12-01	charcoal	0.05	0.7	0.08	1	0	20
2018- 12-01	bread	0.001	0.3	0.05	0.9	0	5

Step 3: Turn Into Time Series

Date	Keyphrase	Doc Freq	Sub %	Mean Buy Intent	Max Buy Intent	Trend in TCN 1-3 Years in the Future?
2014- 01-01	charcoal	[0.02 ... 0.05]	[0.6 ... 0.8]	[0.7 ... 0.8]	[1 ... 1]	Yes
2014- 01-01	bread	[0.02 ... 0.01]	[0.3 ... 0.2]	[0.2 ... 0.1]	[0.85 ... 0.9]	No
2018- 12-01	charcoal	[0.05 ... 0.06]	[0.7 ... 0.9]	[0.08 ... 0.75]	[1 ... 0.95]	Yes
2018- 12-01	bread	[0.001 ... 0.0002]	[0.3 ... 0.25]	[0.05 ... 0.03]	[0.9 ... 0.95]	No

Figure 5.4. Overview of preprocessing in order to extract candidate keyphrases

Table 5.1. Overview of Summary Statistics Used

Type of Feature	Summary Statistic(s)	Num Series
continuous	1) mean 2) maximum 3) minimum 4) sum	4
boolean	1) % True 2) % False (optional) 3) % Not a Number (optional)	2 or 1
string	% Searched String	1
keyphrase-level	keyphrase-level statistic	1

individual univariate time series. This is done by obtaining the values each month for the candidate keyphrase of interest 36 months into the past (i.e. number of Previous Time Windows) for a specific feature and ordering these values by time e.g. charcoal on the 2014-01-01 find the "Mean Purchase Intent" each month from 2011-01-01 to 2013-12-31. As seen in Figure 5.4, the final output of this is multiple univariate time series being created for each candidate keyphrase instance for a given month (i.e. Fixed Time Window).

As time series are built based on data from 36 months into the past (i.e. 36 Previous Time Windows), the time frame in which multivariate time series data is available is reduced. When collecting Reddit data, posts are scraped between 2011-01-01 to 2018-12-31 for each product category. However, due to this construction step, time series data is only available from 2014-01-01 to 2018-12-31. Another consequence of the construction step is that instances with the same keyphrase for the same product category between Fixed Time Windows (i.e. months) are highly similar. This is a result of the rolling window nature of the approach when constructing the time series e.g. the two instances of "charcoal" in the product category Toothpaste for the months 2014-05-01 and 2014-06-01 are nearly the same. As described later in the evaluation (i.e. Section 5.2), due to this the training and testing data must be separated by at least 36 Previous Time Windows for each instance i.e. the time period required for the instances to no longer share any time series data. Therefore, during the evaluation, the training period for each category is between 2014-01-01 to 2014-12-31 and the testing period is between 2018-01-01 to 2018-12-31.[2] This is done as 2014-12-31 and 2018-01-01 are separated by 36 months (avoids any potential train/test overlap that would occur). Because the training period is between 2014-01-01 to 2014-12-31, only 7 product categories can be trained on in this analysis as there is only ground truth available for these 7 for those dates (as seen in Table 3.6). Although the remaining 8 categories are not used in the training process they do not go unused as they are used in the testing process. This is discussed more in the evaluation (Section 5.2).

[2]Due to the fact that ground truth data is only available until 2021-12-31, the testing period used in this chapter is limited to be 2018-12-31, which is 36 months (or 3 years) before the ground truth. It's noteworthy that this chapter predicts future customer needs 12-36 months into the future.

5.1.4 Families of Univariate Time Series Generated

For the analysis in this chapter, the total number of unique univariate time series can be split into families of series, as shown in Table 5.2. In total, there are 1263 univariate time series from 10 families of series. The idea behind including this large feature set is to learn what a future customer need instance looks like on the social media platform Reddit. In this section, an overview of these families of features is given, with a rationale behind the selection process for their inclusion. The appendices (i.e. G-P) give a more in-depth description of how these features are generated.

Table 5.2. Overview of Features Used

Family	Appendix	Number of Time Series
Reddit Information Based	G	51
Frequency Based	H	4
Product Information Based	I	24
Sentiment Based	J	112
Question Detection Based	K	20
Embedding Based	L	350
Subreddit Based	M	100
Kansei Engineering	N	32
Linguistic Based	O	456
User Based	P	114
		1263

The series from the **Reddit Information Based Series** are generated from attributes that are provided with each post from Pushshift i.e. historical Reddit API [164].[3] These collected attributes range from the score (i.e. number of upvotes minus number of downvotes on a post) to whether the post contains a video. The majority of the time series generated here (e.g. a time series generated from an attribute about a post containing a video) may not necessarily be directly useful in the multivariate problem of detecting whether a keyphrase will become a future customer need addressed in real products i.e. occur at a future date in the ground truth dataset (i.e. TCN dataset). However, some features are directly useful, such as series derived from the score or the number of comments, with other studies on Twitter using retweet and like attributes to identify future product trends [322]. Refer to Appendix G for a more detailed description of the exact features created for the Reddit information-based series.

The time series from the **Frequency Based Series** are generated from different statistics about each candidate keyphrase's occurrence in each Fixed Time Window (i.e. month). All of these types of features are keyphrase level statistics (as described in Section 5.1.3) e.g. document frequency. Measures of keyphrase frequency have been used in previous tasks identifying customer needs from social media [22] and are therefore useful in this classification task. Refer to Appendix H for a more detailed description of the exact features generated for the frequency-based series.

[3] https://pushshift.io/api-parameters/ - last accessed 07/06/2024

For the **Product Information Based Series**, features from pre-trained models which are run over Reddit posts are generated. These models all try to capture whether a post is "product-related" in some sense e.g. post contains purchase intent [67].[4] These types of features are good at identifying customer needs in other studies [53], hence are included in this analysis. Refer to Appendix I for a more detailed description of the exact features generated for the product information-based series.

For the **Sentiment Based Features**, as with Product Information Based Features, features are generated from a pre-trained model that is run over Reddit posts. Specifically, the outputs of a model run over the GoEmotions dataset [323] are summarized, which contains 28 output class labels representing emotions (e.g. anger, caring, disappointment, excitement, etc.). Sentiment has been widely used in the customer needs mining literature [54, 178], hence its inclusion as a feature. Refer to Appendix J for a more detailed description of the exact features generated for the sentiment information-based series.

For the **Question Detection Based Series**, as with some other features discussed in this section, features from models that are run over Reddit posts are generated. These models try to detect whether a post is asking a question or stating an answer. These features are included with the hypothesis of future customer need keyphrases being in more posts that contain questions or statements e.g. people asking what charcoal toothpaste was before it became a popular customer need in toothpaste products.[5] Refer to Appendix K for a more detailed description of the exact features generated for the question detection-based series.

For the **Embedding Based Series**, as with some other features discussed in this section, features from models that are run over Reddit posts are generated. This is done by using pre-trained document and phrase embedding models with the Python libraries SBERT [324], spaCy [313] and fasttext [325]. Embedding information has already been used to identify customer needs in other studies [53, 326] (although used for document classification). It is feasible to say that it will provide predictive information for this classification task as past trending phrases likely share a similar vector space by having similar meaning (i.e. phrase embeddings) while customer need keyphrases may be in similar documents to past trending ones (i.e. document embeddings). Refer to Appendix L for a more detailed description of the exact features generated for the embedding-information based series.

For the **Subreddit Based Series**, subreddit (discussion form on Reddit) information associated with each post is summarized. As the subreddit feature on Reddit is a string (e.g. r/AskReddit, r/Music), defined strings are searched for to generate a statistic for each candidate keyphrase (as described in Section 5.1.3). The defined strings searched for come from 100 of the most subscribed subreddits at the time of experimentation, resulting in 100 new univariate time series features in the classification problem. The use of subreddit information was also applied in Chapter 4 to identify future customer needs. It is useful as certain subreddits may be indicative of places where new trends are discussed e.g. in the subreddit r/eli5 people may ask questions about queries they want solved, such as best

[4] https://huggingface.co/j-hartmann/purchase-intention-english-roberta-large - last accessed 07/06/2024

[5] https://www.reddit.com/r/NaturalBeauty/comments/2s6h2u/best_homemade_whitening_toothpaste/ - last accessed 07/06/2024

ingredients to use for smoother lips (lip balm) or whiter teeth (toothpaste). Refer to Appendix M for a more detailed description of the exact features generated for the subreddit information-based series.

For the **Kansei Engineering Based Series**, posts are classified based on whether they contain one of the words in a Kansei group. Kansei engineering has been described as "translating technology of a consumer's feeling and image for a product into design elements" [111]. Recently, it has become an important topic in the customer needs mining literature for product development, with many computational approaches for it being built [25, 26, 213–215]. Traditional non-computational approaches to Kansei engineering work on questionnaires to measure a user's feelings towards a customer need, where groups of words called Kansei attributes are used to measure their emotions. Kansei attributes consist of a pair/groups of bipolar words in which respondents indicate their feeling towards a product e.g. 1) unique-personalized-rare vs common-general; 2) quality-reliable-sturdy-safe vs unreliable or 3) novel-fresh-interesting vs boring. Posts are classified based on whether they contain one of the words in a Kansei group. To retrieve a list of these Kansei attributes, the work in [213] is followed, which first identifies 16 groups of bipolar Kansei attributes from 10 previous Kansei engineering studies (primarily in the last decade) and then expands on these attributes using an automated method. Refer to Appendix N for a more detailed description of the exact features generated for the Kansei Engineering information-based series.

For the **Linguistic Based Series**, as with some other features discussed in this section, features from models that are run over Reddit posts are summarized. Keyphrase-level statistics are also analyzed. All the univariate series generated either represent 1) tagging information (e.g. POS, dependency labels, etc.), 2) document information (e.g. post length) or 3) phrase-level information (e.g. the number of vowels, whether it contains an @ symbol, etc.). Refer to Appendix O for a more detailed description of the exact features generated for the linguistic information-based series.

For **User Based Series**, features based on authors are generated (i.e. users on Reddit). The use of author information has been seen in many of the social media forecasting topics already discussed in this thesis e.g. predicting customer needs [135, 207] and detecting future occurrences using MTSC [194]. Refer to Appendix P for a more detailed description of the exact features generated for the user information-based series.

5.1.5 Time Series Classification

In this section, the time series techniques used to address the future customer needs keyphrase classification problem are detailed. Specifically, the following is discussed: 1) how the ground truth label is added to each candidate keyphrase from the TCN dataset; 2) the MTSC algorithm used for the task (i.e. Supervised ML); and 3) the use of MTL to build a single model capable of identifying future customer needs in any product category.

As previously discussed, the ground truth dataset (i.e. TCN) used in this chapter consists of the top 20 most addressed customer needs in products each month from 2014-01-01

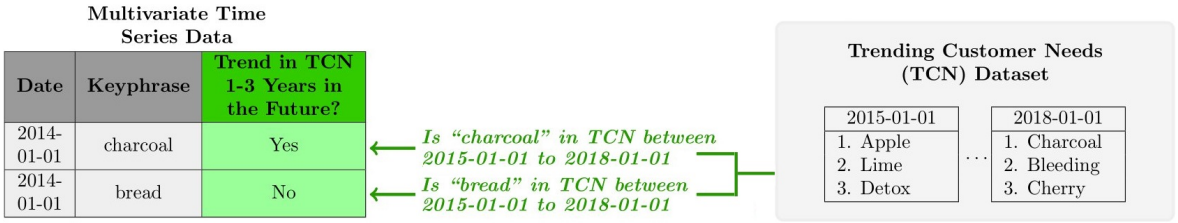

Figure 5.5. How ground truth label is added for classification problem from the TCN dataset

across multiple product categories. Figure 5.5 shows how this data is added as the ground truth label for each multivariate time series instance, where each instance is a candidate keyphrase in a given Fixed Time Window (i.e. month) for a particular product category (i.e. Toothpaste in Figure 5.5). A binary output label indicates whether the phrase appears in the TCN dataset for the product category being analyzed 1-3 years in the future. This is seen in the figure for the term "charcoal" on 2014-01-01 which has a positive output label (i.e. Yes) given that it appears as a top customer need in the TCN dataset 1-3 years in the future i.e. in 2018-01-01. The next instance "bread" has a negative output label (i.e. No) as it doesn't appear in the TCN dataset during that time period. It is important to understand that the main objective of adding the ground truth label to the instances is to train and evaluate a MTSC algorithm that predicts customer needs ahead of time before they hit the marketplace (specifically 1-3 years ahead). The main premise behind this is that future customer needs in a product dataset (i.e. TCN) represent needs that are currently unmet and are therefore valuable for businesses to identify. Although not seen in Figure 5.5, it is also valuable to point out that there is a high degree of data imbalance between the positive and negative label in the final ground truth column (i.e. "Trend in TCN 1-3 Years in the Future?" column in Figure 5.5). This data imbalance is detailed in Section 5.2.1 i.e. there is a ∼260:1 ratio of negative instances to positive instances. To clarify, there are a lot more negative instances than positive ones because multiple thousand instances are being analyzed in each Fixed Time Window (i.e. month) and only 20 customer needs 1-3 years ahead each month in the TCN dataset. In Section 5.2, the techniques used to handle this data imbalance are discussed.

As discussed in Section 2.6, the algorithm MINIROCKET is used to solve the MTSC problem addressed in this chapter. To recap, MINIROCKET is a fast process used to transform a 2D multivariate time series into a 1D vector using random convolutional kernels. This 1D vector is then used as an instance to train a linear classifier such as Ridge/Logistic Regression to solve the classification task. During experiments, the multivariate version of MINIROCKET from sktime is used.[6] To train the linear classifier on the embeddings produced by MINIROCKET, the cross-validated version of Ridge Regression from sklearn is used (which is one of the recommended algorithms to use with MINIROCKET [246]).[7] The following implementations of models from the mentioned libraries are used as they are the recommended ones used in the linked coding repository of MINIROCKET.[8] The

[6] http://www.sktime.net/en/v0.13.0/api_reference/auto_generated/sktime.transformations.panel.rocket.MiniRocketMultivariate.html - last accessed 07/06/2024

[7] https://scikit-learn.org/1.0/modules/generated/sklearn.linear_model.RidgeClassifierCV.html - last accessed 07/06/2024

[8] https://github.com/angus924/minirocket - last accessed 07/06/2024

same default hyper-parameter values for the two models as in the repository are also used. Two of these important hyper-parameter values include: 1) 10,000 for the *num_kernels* parameter of MINIROCKET - producing an embedding space of 10,000 dimensions that the linear model is trained on; and 2) True for the *normalize* parameter of Ridge Regression - standardizes the embeddings before training/testing the classifier.

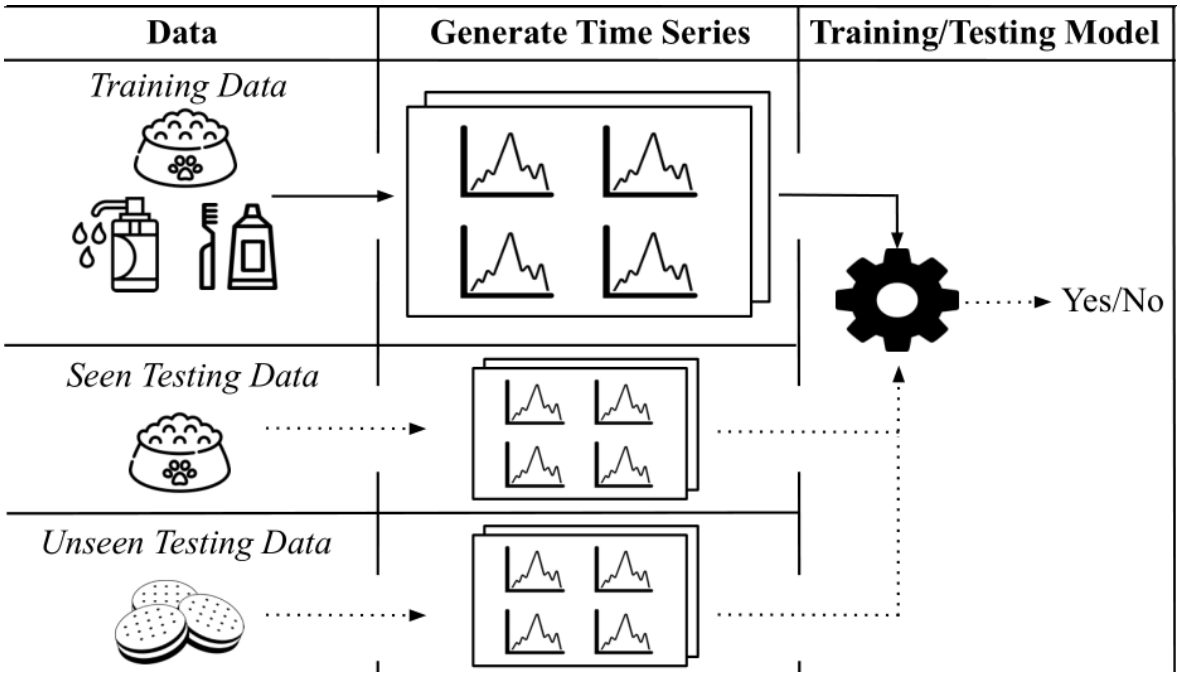

Figure 5.6. Multi-Task Learning: A generalizable model is built from multiple product categories (e.g. Dog Food, Shampoo and Toothpaste) and tested on categories it has seen (e.g. Dog Food) and not seen during training (e.g. Cookies).

As discussed, the use of MTL is a key contribution to this thesis. How it is used is described in Figure 5.6. During training, time series features from the instances of the available training product categories are generated (e.g. Dog Food, Shampoo and Toothpaste in Figure 5.6). A model is then built from these instances which contains the ground truth label that allows the prediction of future customer needs. During testing, time series features are generated using the same process during training, however, only for one category. The described trained model is then used to classify these instances. Two types of product categories are used when testing the model during evaluation: 1) *Seen Testing Category* - a category the model has seen during training (e.g. Dog Food in Figure 5.6); and 2) *Unseen Testing Category* - a category the model has not seen during training (e.g. Cookies in Figure 5.6).[9] As discussed in Chapter 3, the categories in Table 3.1 that have ground truth data on/before 2014-01-01 are used to train the model (i.e. 7 categories) and thus also make up the Seen Testing Categories. This is due to the fact the training time period in the evaluation is between 2014-01-01 to 2014-12-31 (for the reasons described in Section 5.1.3). All other categories are not used in model training and therefore make up the Unseen Testing Categories (i.e. 8 categories). During the evaluation, it is shown that the model which is produced from this described MTL approach leads to similar performance compared to training and testing on the same product category e.g. train on Dog Food to predict Dog Food. This is important as this model can be used on categories the model has

[9]For the Seen Testing Category, although the model has used the category in the training process the same data is not used for training and testing - described at more detail in the evaluation (Section 5.2)

not seen during training. It does this by learning what general future customer needs look like on Reddit rather than one for a particular product category. The reason this model performs better is due to the MTL characteristic of *Task-Relatedness* [80] i.e. tasks are similar. In this task, this characteristic is seen due to the signals of future customer needs on Reddit for different categories being similar e.g. Toothpaste and Cookies. This is also the logic behind most of the MTL approaches working better throughout the ML literature e.g. [327] made a better-performing classification model which learns higher-level features by using MTL to train on images from multiple object categories. This characteristic of *Task-Relatedness* for this problem is helped by how task-agnostic features are generated. This is seen in Section 5.1.4 where the features generated are not specific to any one product category but rather general to multiple product categories e.g. user/frequency/sentiment features.

5.2 Evaluation

This section addresses the following two RQs: 1) To what extent can future customer needs be detected with performance useful for the purposes of product development (i.e. RQ 1)?; and 2) how can MTL (described in Section 5.1.5) be used to train a generalizable model for which future customer needs can be predicted for a product category the model has seen and not seen during training (i.e. RQ 3)? To do this, the approach is first compared to the rule-based approach described in Chapter 4 by assessing how the model compares to it when trained and tested on the same product category e.g. train and test on toothpaste. Here, it is shown that the approach in this chapter builds on that of the previous chapter by using techniques from ML to obtain significantly better performance. The performance of MTL is then shown when compared to the approach of training and testing on the same category.

This section first details the two different training strategies used during experiments i.e. training on one versus multiple categories for prediction (Section 5.2.1). When describing the strategies, an in-depth explanation of the specific model training and validation details used during experiments is given. The approach is then compared to a baseline (i.e. approach in Chapter 4), which carries out the same future prediction task as in this chapter (Section 5.2.2). The impact of MTL is then assessed (Section 5.2.3). A further investigation into the results is then performed which highlights the benefits of the approach in general and looks at where future improvements can be made e.g. viewing misclassifications (Section 5.2.4). Finally, a summary and discussion of the evaluation is given (Section 5.2.5).

5.2.1 Evaluation Methodology

During the evaluation, two training strategies are employed: 1) **One Category** training approach and 2) **Multiple Category** training approach i.e. MTL. These different ap-

proaches are used at various stages in the evaluation and are finally compared at the end of this section. When generating results for the One Category approach, the same category data is used to train and test the model e.g. train and test on dog food. In the Multiple Category approach, one model is trained using a large number of product categories and then tested individually on categories it has seen and not seen during training (as described in Section 5.1.5) e.g. train on dog food, nail polish, shampoo, etc. and test on a Seen Training Category like dog food as well as an Unseen Category like cookies. Although the same product category data for the One Category and Multiple Category approaches is used, this data is still split to remove any train/test overlap.

As discussed in Section 5.1.3, as a result of the 36-month (i.e. length of Previous Time Window) rolling window nature of the approach in this chapter, instances with the same keyphrase for the same product category between Fixed Time Windows (i.e. months) are highly similar e.g. the two instances of "charcoal" in the product category Toothpaste for the months 2014-05-01 and 2014-06-01 are nearly the same. As a result of this data overlap, the training and testing data for the One Category approach are required to be separated by at least 36 Previous Time Windows (i.e. 36 months) because 36 Previous Time Windows of data are used for each instance (to avoid potential train/test contamination). Multivariate time series data is available from 2014-01-01 to 2018-12-31. Due to this train/test overlap issue, the model is trained on data from 2014-01-01 to 2014-12-31 and tested on data from 2018-01-01 to 2018-12-31. For similar reasons, the same train/test split is followed for the Multiple Category approach. As discussed previously (Section 5.1.5), this is the reason why 7 (and not 15) categories are used in the model training process. As seen in Table 3.1, this is because only 7 categories have ground truth data at/before the dates between 2014-01-01 to 2014-12-31 i.e. training period. These are the only categories used in the One Category approach in the experiments, as the One Category uses the same category to train and test the model i.e. if no training data is available for a category then it cannot be tested. These are also the only categories used to train the Multiple Category MTL approach, therefore making up the Seen Testing Categories (described in Section 5.1.5). The remaining 8 categories are solely used to test the Multiple Category MTL model to see if it generalizes on categories it has not seen during training, thus making up the Unseen Testing Categories (described in Section 5.1.5).

As discussed in Section 5.1.5, the MINIROCKET model followed by a Linear Ridge Regression classifier is used when detecting future customer needs. During the model-building process, it was discovered that these two models are infeasible to use on the entirety of the training data from a time complexity standpoint. This is because on average each category has ∼34,000 instances every month with 1,263 univariate time series, which are each 36 months in length. This results in ∼18,550,944,000 data points for each category over 12 months i.e. from 2014-01-01 to 2014-12-31. The time or computing power to deal with this data was not available, even with the fast speed of MINIROCKET. Instead, the data is undersampled, regardless of whether the One Category or the Multiple Category approach is run. This significantly reduces the number of instances used for training as there is a massive data imbalance between the positive and negative ground truth labels, as discussed in Section 5.1.5. Specifically, there is a ∼260:1 ratio of negative instances to positive instances before undersampling across the categories used in the analysis from

2014-01-01 to 2014-12-31. When undersampling, random undersampling is employed.[10]

For the task in this chapter, undersampling allows a significant reduction in training time, however, it comes with the drawback of not representing the true distribution of the output class(es) that significantly affects predictions produced by classifiers [328]. For example, a model trained on all the training data versus a model trained on an undersampled version of the same data could easily predict a different output class for the same instance. The models in this chapter likely predict far too many instances as the positive class (i.e. future customer needs) during testing, given that they are trained on a fabricated version of the data which highly undersamples the negative class. To mitigate this undesired consequence, the information from the predicted probability output of the model is used rather than asking it what class an instance belongs to. This probability output is obtained using the *decision_function*[11] and *softmax* function[12] available in sklearn. This predicted probability output is used to find an optimal threshold that yields the highest performance on a held-out validation set at predicting future customer needs e.g. all instances that have a probability output greater than 0.8 (threshold) for the positive class are predicted as positive instances. Specifically, the probability output of the Linear Ridge Regression classifier is validated (which is first trained on the 2D embeddings produced by the MINIROCKET algorithm run over the 3D multivariate time series data). When choosing the threshold, an exhaustive grid search is performed for the probability value that yields the best F1 score between the values 0 to 1 with a step size of 0.01 (the choice of the F1 metric for model performance is elaborated later in this section).[13] It is noteworthy that finding this probability threshold for the classification of imbalanced data is an area that has been thoroughly explored in the ML literature [329].

When splitting the training data from 2014-01-01 to 2014-12-31 into training and validation sets, the traditional approach of randomly splitting the data on a per-instance basis is not used, as instances with the same keyphrase could have highly similar multivariate time series (as previously discussed in this section). This would not represent the real relationship between the training and testing sets, which have no overlap. To bypass this issue, the data is split on the unique keyphrase level instead which ensures no training/validation overlap. Specifically, 90% of the unique keyphrases are randomly sampled from the training data and the instances associated with these keyphrases are used to train the model. The remaining 10% of keyphrases representing instances are then used to validate it. When splitting the keyphrases, stratified random sampling is used - where unique keyphrases are divided based on if they ever become a future customer need i.e. contained in the ground truth TCN dataset 1-3 years in the future. When training the model, the data is then downsampled to a 1:1 ratio of positive and negative instances (allowing for faster training times). The validation data is then only downsampled to the ratio that represents the initial class distribution between positive and negative in the original training data e.g. 260:1

[10] https://imbalanced-learn.org/stable/references/generated/imblearn.under_sampling.RandomUnderSampler.html - last accessed 07/06/2024

[11] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.RidgeClassifierCV.html#sklearn.linear_model.RidgeClassifierCV.decision_function - last accessed 07/06/2024

[12] https://github.com/scikit-learn/scikit-learn/blob/188267212cb5459bfba947c9ece083c0b5f63518/sklearn/utils/extmath.py#L863 - last accessed 07/06/2024

[13] No other parameters are validated for computational/time purposes. Such parameters may include the inputs into MINIROCKET (e.g. number of kernels) or Linear Ridge Regression (e.g. alpha parameter).

ratio. This is done as a probability threshold that represents the class distribution in the real training data is required to be estimated. By removing 90% of the unique keyphrases that represent future customer need instances this distribution is misrepresented, hence the need to undersample again.

Data: $catgoryArr \leftarrow [' shampoo',' toothpaste',' eyeliner']$
for $category \in categoryArr$ **do**
  $allData = \textbf{getDataForCategory}(category)$
  $allTrainData$, $allTestData = \textbf{splitDataByDate}(allData)$
  $trainData, validData = \textbf{splitDataByValidProcedure}(allTrainData)$
  $trainDataUndersampled = \textbf{undersample}(trainData)$
  $model = \textbf{trainModel}(trainDataUndersampled)$
  $probabilityThreshold = \textbf{validate}(model, validData)$
  $prediction = \textbf{test}(allTestData, model, probabilityThreshold)$
end

Algorithm 1: How the One Category approach is trained, validated and tested

Algorithm 1 shows the pseudocode for how the model is validated, trained and tested. It draws upon all of the topics discussed in this section e.g. how the data is undersampled, how it is split into training and validation sets etc. Algorithm 1 specifically shows how this is done for the One Category approach (i.e. train and test using the same product category), however, this example can easily be extended to the Multiple Category approach by instead using multiple categories to train/validate/test the model. To start, the product categories analyzed are defined i.e. *categoryArr*. In the algorithm, these categories are shampoo, toothpaste and eyeliner. However, during experiments, more than just these categories are analyzed (as discussed multiple times throughout the thesis). These categories are looped through with various processes being applied in each iteration which are responsible for training, validating and testing the model. For the first process applied in each iteration, all the data for the category being analyzed is retrieved i.e. **getDataForCategory**. It is assumed all processing has been applied in this step to turn the Reddit posts into candidate keyphrases - which each consist of multiple features in the form of univariate time series (as described in Section 5.1.3). Secondly, the multivariate time series data is split by date into training and testing splits i.e. **splitDataByDate**. Due to the train/test overlap issue described in this section, for each category, data is reserved from 2014-01-01 to 2014-12-31 for training and 2018-01-01 to 2018-12-31 for testing. Thirdly, the training data is further split into training and validation sets i.e. **splitDataByValidProcedure**. Here, it is split on the unique keyphrase level (for the reasons described in this section). After, the training data is undersampled, using random undersampling, due to it being computationally infeasible to run the model on the entirety of the data (as described in this section) i.e. **undersample**. The model (MINIROCKET followed by a Linear Ridge Regression classifier) is then trained on the undersampled data i.e. **trainModel**. It is then validated on the held-out validation set to obtain the probability threshold that optimises the F1 score i.e. **validate**. As explained, this is done as the output class distribution represented by the undersampled training data does not estimate the class distribution of the testing data (which is not sampled). When the model is finally applied to the testing data (i.e. **test**), the probability threshold is used when classifying instances e.g. if the probability output of the instance is above the 0.8 threshold it is classified as a positive

instance.

When running the experiments on the test data, it was noticed that different versions of the same model type were sometimes not producing similar results e.g. the One Category model for lip balm was not producing the same results. This is due to the various stochastic processes performed when transforming the data. Such processes include applying the MINIROCKET algorithm which produces random 2D kernel embeddings for each run or undersampling the training data randomly. To obtain a realistic result set, each approach is run 10 times and the mean results in the experiments are reported. This increases the time complexity of the experiment, however, is necessary to reduce experimental bias.

5.2.2 Baseline Approach Comparison

In this section, the model's performance against the approach in Chapter 4 (i.e. baseline) is examined using the two evaluation approaches defined in Section 3.5 i.e. Binary Classification Evaluation and List Evaluation. This is done to show that using ML leads to improvements in performance compared to when a rule-based approach is employed (i.e. Chapter 4). In this section, the rule-based approach is only compared against the One Category approach described in this chapter i.e. training/testing on the same product category (detailed in Section 5.2.1). This is done to clearly show that the ML approach is better than the rule-based baseline, without the use of MTL i.e. Multiple Category approach. In Section 5.2.3, the performance of the Multiple Category approach (which uses MTL) is shown, which shows further improvements over the One Category approach.

As discussed in Chapter 4, the baseline used in this section is a rule-based approach that predicts future customer needs from Reddit that are of interest to businesses. It addresses the same overall task as in this chapter. However, the keyphrases in Chapter 4 are predicted in ranked lists compared to a binary output (as in this chapter). The baseline is also only evaluated on the task of identifying future customer needs in Toothpaste products. However, this chapter is assessed for three categories as it is about a multi-category analysis. These categories are 1) Toothpaste, 2) Perfume and 3) Dog Food. More categories are not included due to the time it takes to collect Google Trends data for each category to run the baseline (i.e. rule-based approach from 4). When picking categories to compare, Toothpaste is chosen as it is the one used in the baseline. The two other categories are selected as they make up a diverse range of categories for the comparison experiment. There are several parameters required to run the baseline, including category-specific input parameters. It is intended to act as favourably as possible to the baseline when providing it with these parameters for categories it didn't use in its evaluation i.e. Dog Food and Perfume. Appendix Q further details the parameters and parameter values used for each category.

5.2.2.1 Baseline Approach Comparison - Binary Classification Evaluation

In this section, both the One Category ML approach and the rule-based baseline are assessed using the Binary Classification Evaluation (i.e. Section 3.5.2). These approaches are

evaluated on the instance level by checking whether the predicted keyphrase instances are in the ground truth TCN dataset. As the output of the rule-based baseline is a ranking of keyphrases for each Fixed Time Window (i.e. month), some manipulations are required for it to be able to be assessed using this evaluation approach. To turn this ranking approach into a form suitable for binary classification, the natural ordering of the keyphrases is used when classifying. Specifically, a threshold is introduced to set the first number of ranked keyphrases (sorted in ascending order of rank) to be true (i.e. are future customer needs) while the others remain false (i.e. are not future customer needs). During experimentation, 15 of these thresholds were explored to find which one yields the best F1 score. Precisely, the following thresholds are used: 5, 10, 15, 20, 50, 100, 250, 500, 750, 1000, 1250, 1500, 1750, 2000, and 2500. These thresholds are tried as they represent a large and widespread range of values used to find a near-optimal F1 for the baseline. Smaller increment values are used at the start (e.g. 5, 10, 15, 20) due to there being only a small number of positive instances in the dataset (as discussed in Section 5.2.1).

The results of this evaluation for the One Category approach and baseline approach are seen in Table 5.3. As seen in the table, the One Category approach outperforms the best baseline approach across all 3 categories. Furthermore, it outperforms the baseline on the categories that aren't addressed in its experiments by a large margin i.e. Dog Food and Perfume. This is probably because the baseline's parameter values aren't tailored to these categories. In the broader picture, however, it does better because it uses supervised ML. It therefore can learn from past instances of future customer needs, which generalizes better than rules encoded by a human in the baseline approach e.g. is the *Min Chi Square P-value* parameter < 0.02?[14]

In this section, the F1 scores of the One Category and baseline approaches are also compared using a statistical test for each of the product categories used in the baseline comparison i.e. Toothpaste, Dog Food and Perfume. This can be run as both the One Category approach and the baseline approach are run multiple times, as detailed in Section 5.2.1 and Section 5.2.2. Specifically, a Mann-Whitney U test [308] is run comparing the F1 scores of the One Category approach and the "best" baseline approach. The "best" baseline approach for each category is represented by the threshold which has the highest mean performance seen in Table 5.3 i.e. 500 for Toothpaste, 1250 for Dog Food and 1000 for Perfume. As with previous studies comparing result data from the output of different ML models [330], the reason this test is used instead of a t-test is that the F1 scores for each approach are not normally distributed [308].[15] It is important to note, that for the same reasons, this test is used throughout this evaluation to compare different samples of results. Table 5.4 shows the p-value (rounded to 3 decimal places) of this test for each product category. If the test is in favour of the baseline by the median scores being greater than the One Category scores, a + is post-fixed to the result (as in [330]). The reason why the median scores are compared, in this case, is because the Mann-Whitney U test is a "test of medians" [331] - therefore the test is in favour of the baseline if the median scores

[14] The *Min Chi Square P-value* is a parameter used in the baseline approach to find non-discriminate phrases (Section 4.1.2)

[15] The t-test compares the means of the two samples and assumes they are normally distributed while the Mann-Whitney U test compares the rank sum of the two samples and doesn't assume they are normally distributed [308].

Table 5.3. Binary Classification Evaluation showing the mean F1 Scores (rounded to 3 decimal places) for the One Category and baseline approaches. The best result for the baseline across each threshold for each category is in bold

Category	Method					
	One Category	*Baseline*				
Toothpaste	0.099	*5*	*10*	*15*	*20*	*50*
		0.012	0.016	0.023	0.025	0.036
		100	*250*	***500***	*750*	*1000*
		0.049	0.069	**0.073**	0.072	0.071
		1250	*1500*	*1750*	*2000*	*2500*
		0.068	0.064	0.059	0.057	0.055
Dog Food	0.120	*5*	*10*	*15*	*20*	*50*
		0.000	0.000	0.003	0.006	0.019
		100	*250*	*500*	*750*	*1000*
		0.028	0.057	0.065	0.068	0.069
		1250	*1500*	*1750*	*2000*	*2500*
		0.070	0.068	0.066	0.061	0.059
Perfume	0.130	*5*	*10*	*15*	*20*	*50*
		0.000	0.001	0.002	0.006	0.035
		100	*250*	*500*	*750*	***1000***
		0.052	0.063	0.071	0.080	**0.085**
		1250	*1500*	*1750*	*2000*	*2500*
		0.085	0.080	0.076	0.070	0.059

The mean F1 scores (rounded to 4 decimal places) for Perfume at the thresholds 1000 and 1250 are 0.0852 and 0.0850 respectively, hence why the threshold at 1000 is the best baseline.

are greater than the One Category scores. The results in the table solidify the fact that the One Category approach is better than the baseline for the F1 metric using the Binary Classification Evaluation. This is because for all the categories analyzed, the results from the two samples are significantly different (i.e. all p-values are <0.001) and the median results for the One Category approach are all greater than the baseline.

5.2.2.2 Baseline Approach Comparison - List Evaluation

Similar to the output of the ranking algorithm, the proposed ML approach addressed in this chapter needs to have its output transformed for it to be evaluated by the List Evaluation approach i.e. to calculate List Mean Precision and List Recall (described in Section 3.5.1). Specifically, this involves changing the binary prediction output to a ranked list of keyphrases each Fixed Time Window (i.e. month). This is done by using the predicted probability score outputted by the Linear Ridge Regression classifier (i.e. ML model used in this study) to rank the terms of each Fixed Time Window in descending order of confidence. By ranking this way, the instances the model estimates are most likely to become future customer needs will be at the top of the list, while the ones it least estimates will become future customer

Table 5.4. P-value (rounded to 3 decimal places) for the Mann-Whitney U Test of F1 scores from the One Category approach vs the best baseline approach for Binary Classification Evaluation

Category	F1
Toothpaste	<0.001
Dog Food	<0.001
Perfume	<0.001

+ test is in favour of the Baseline

needs will be at the bottom.

The results of the evaluation for the One Category and baseline approach are seen in Table 5.5. The One Category approach is better than the baseline across all the results for the Dog Food and Perfume categories. However, the baseline performs better for the Toothpaste category by obtaining higher performance on all the List Mean Precision results and one of the List Recall results. As discussed previously in this evaluation, the baseline is specifically tuned for the Toothpaste category across the metrics used in the List Evaluation, so it is not surprising that it performs better here.

Table 5.5. List Evaluation showing the mean results (rounded to 3 decimal places) for the One Category and baseline approaches. For each category the result from the best approach is in bold

Category	Method	Metric							
		Mean Precision				*Recall*			
		K=5	*K=10*	*K=15*	*K=20*	*K=5*	*K=10*	*K=15*	*K=20*
Toothpaste	One Category	0.057	0.109	0.124	0.155	0.009	**0.026**	**0.038**	**0.054**
	Baseline	**0.221**	**0.205**	**0.19**	**0.161**	**0.011**	0.015	0.019	0.023
Dog Food	One Category	**0.11**	**0.159**	**0.203**	**0.225**	**0.011**	**0.029**	**0.042**	**0.057**
	Baseline	0.0	0.0	0.033	0.049	0.0	0.0	0.007	0.01
Perfume	One Category	**0.102**	**0.179**	**0.205**	**0.244**	**0.026**	**0.05**	**0.076**	**0.104**
	Baseline	0.0	0.005	0.009	0.018	0.0	0.006	0.012	0.023

As in Section 5.2.2.1, the results of the One Category and baseline approaches are also compared using a Mann-Whitney U test, as they are run multiple times. This test is chosen for the same reasons as described in Section 5.2.2.1 i.e. result data is not normally distributed. Table 5.6 shows the p-value (rounded to 3 decimal places) for each metric in the List Evaluation over every product category used in the baseline comparison. As in Section 5.2.2.1, if the median scores of the baseline are greater than the median scores of the One Category approach, a + is post-fixed to the result. For all the results in the table, the One Category approach is significantly better 18/24 times when the p-value level is either 0.1, 0.05 or 0.01 (i.e. test is in favor of the One Category approach and the p-value is under the mentioned levels). The baseline is significantly better 4/24 times when the p-value level is 0.1 and 3/24 times when the level is 0.05 or 0.01. The levels (i.e. 0.1,

0.05, 0.01) are reported here as they have been commonly used in other studies to test for statistical significance [332]. The results in this table demonstrate that the One Category approach is better than the baseline for the List Evaluation barring the List Mean Precision metric for the Toothpaste category.

Table 5.6. P-value (rounded to 3 decimal places) for the Mann-Whitney U Test of results from the One Category approach vs the baseline approach for List Evaluation

Category	Metric							
	Mean Precision				*Recall*			
	K=5	*K=10*	*K=15*	*K=20*	*K=5*	*K=10*	*K=15*	*K=20*
Toothpaste	<0.001+	<0.001+	0.001+	0.877+	0.086+	0.249	0.002	<0.001
Dog Food	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001
Perfume	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001

+ test is in favour of the baseline

5.2.2.3 Baseline Approach Comparison - Summary

To summarize, the One Category approach outperforms the baseline entirely in the Binary Classification Evaluation (Section 5.2.2.1) and mostly in the List Evaluation (Section 5.2.2.2), with the exception of the List Mean Precision metric for the Toothpaste category. Considering these results, it is shown that future customer needs can be predicted with better performance than the previously proposed approach. By doing this, further developments are made in addressing RQ 1 which states that future customer needs can be predicted using Reddit.

Additionally, it can be concluded that the effort to obtain predictions from the One Category approach is much lower than the rule-based approach. This is because the rule-based approach requires input parameters to run it, such as the "Gold Standard Subreddit" and the "Google Trends Category" (as discussed in Section 5.2.2). Conversely, the One Category approach only requires the raw posts when making predictions.

5.2.3 Impact of Multi-Task Learning

This section outlines how the Multiple Category approach (which uses MTL) achieves similar performance to the One Category approach i.e. MTL model trained on all product categories is similar to using the same category data to solely train/test the model. This is important as an MTL model can be used to predict categories not seen in the training process, therefore generalizing to unseen categories without having to be retrained. By doing this, RQ 3 is addressed, which questions whether MTL can be employed to train a generalizable model for which future customer needs can be predicted for a product category the model has and has not seen during training.

As discussed in Section 5.1.5, there are two types of categories used to test the MTL model: 1) Seen Testing Categories; and 2) Unseen Testing Categories. The Seen Testing Categories are categories used in testing but are also used by the model in the training process. Conversely, the Unseen Testing Categories are categories used in testing but are not in training. As discussed earlier in the evaluation (i.e. Section 5.2.1), only the Seen Testing Categories are used to train the Multiple Category MTL approach as they are the only categories that have ground truth data from the TCN dataset (i.e. ground truth) in the specified training period in the experiments i.e. between 2014-01-01 to 2014-12-31. These Seen Testing Categories are also the only categories used in the One Category approach as they have data to train and test on the same category. As seen in Table 3.1, the Seen Testing Categories have ground truth data from TCN at/before 2014-01-01. Specifically, these 7 categories are Dog Food, Eyeliner, Lip Balm, Nail Polish, Perfume, Shampoo and Toothpaste. The Unseen Testing Categories make up the 8 remaining categories in Table 3.1 i.e. Beer, Cereal, Coffee, Cookie, Pizza, Popcorn, Soda and Soup. In this section, two separate evaluations for assessing the Seen and Unseen Testing Categories are carried out. The main reason for this is that a comparison analysis between the One Category approach and the Multiple Category approach can only be performed on these categories. After all, the One Category approach can only make predictions for the Seen Testing categories. Although the Unseen Testing Categories are not used in the comparison analysis (i.e. to examine whether the MTL approach is better than using the same category to train/test a model), they still contribute to the evaluation as they test if the MTL model is capable of detecting future customer needs on categories it hasn't seen during training e.g. can a model trained on Eyeliner, Toothpaste and Perfume predict an unseen category such as Cookies. This is important as it truly tests to see if the model can learn what a general future customer need keyphrase looks like on Reddit (as discussed in Section 5.1.5), as opposed to the One Category approach, which learns what it looks like for a specific category.

5.2.3.1 Multi-Task Learning Approach Comparison For Seen Categories - Binary Evaluation

This section compares the One Category approach to the Multiple Category MTL approach for the Seen Testing Categories using the Binary Classification Evaluation (discussed in Section 3.5.2). The results of this evaluation are shown in Table 5.7. As seen in the table, the Multiple Category approach outperforms the One Category approach across 5 of the 7 categories. The One Category approach obtains higher performance in 1 category (i.e. Perfume) and they both achieve the same performance for 1 category (i.e. Shampoo).

As in Section 5.2.2, the F1 scores of the One Category and Multiple Category approaches for the Seen Testing Categories are also compared using a Mann-Whitney U test (as they are run multiple times). Table 5.8 shows the p-value (rounded to 3 decimal places) of this test for each product category analyzed. As in Section 5.2.2, if the median scores of the One Category approach are greater than the median scores of the Multiple Category approach, a + is post-fixed to the result. Although the Multiple Category approach performs better across 5 of the 7 categories (as shown in Table 5.7), it only performs significantly better

Table 5.7. Binary Classification Evaluation showing the mean F1 Scores (rounded to 3 decimal places) for the One Category and Multiple Category approaches across the Seen Testing Categories. For each category the result from the best approach is in bold

Category	Method	
	One Category	*Multiple Category*
Dog Food	0.120	**0.121**
Eyeliner	0.069	**0.077**
Lip Balm	0.099	**0.136**
Nail Polish	0.096	**0.116**
Perfume	**0.129**	0.085
Shampoo	**0.115**	**0.115**
Toothpaste	0.099	**0.115**

Table 5.8. P-value for the Mann-Whitney U Rank Test of F1 scores (rounded to 3 decimal places) from One Category approach vs Multiple Category approach for Binary Classification Evaluation across the Seen Testing Categories

Category	F1
Dog Food	0.971
Eyeliner	0.247
Lip Balm	0.015
Nail Polish	0.19
Perfume	0.019+
Shampoo	0.853+
Toothpaste	0.247

+ test was in favour of the One Category approach

1/7 times when the p-value level is 0.1 or 0.05 and never when the level is 0.01.[16] The baseline is also better 1/7 times when the p-value level is 0.1 or 0.05 and never when the level is 0.01. These levels (i.e. 0.1, 0.05, 0.001) are used for the same reasons as discussed in Section 5.2.2 i.e. commonly used in other studies. The results in the table show that the Multiple Category approach performs similarly to the One Category approach for the Binary Classification Evaluation.

5.2.3.2 Multi-Task Learning Approach Comparison For Seen Categories - List Evaluation

This section compares the One Category approach to the Multiple Category MTL approach for the Seen Testing Categories using the List Evaluation (discussed in Section 3.5.1). As performed in the baseline comparison (i.e. Section 5.2.2), the output of both the One Category and Multiple Category approaches is changed for them to be evaluated using the List Evaluation approach i.e. by using the predicted probability output. The results of this evaluation are seen in Table 5.9. The One Category performs better than the Multiple Category approach here, with it obtaining 33 of the best results from a total of 56 in the table. The Multiple Category approach obtains 21 of the best results while they both obtain the same result twice (i.e. Recall when K is 10 for Nail Polish and Recall when K is 5 for Toothpaste).

As in Section 5.2.2, the results of the One Category and Multiple Category approaches are also compared using a Mann-Whitney U test (as they are run multiple times). Table 5.10 shows the p-value (rounded to 3 decimal places) of this test for each product category. As in Section 5.2.2, if the median scores of the One Category approach are greater than the median scores of the Multiple Category approach, a + is post-fixed to the result.

[16] The Multiple Category approach for Shampoo is the same as the One Category approach in Table 5.7 , however, the One Category approach slightly outperforms it in Table 5.8. This is because the mean result is recorded in Table 5.7 and the median is recorded in Table 5.8

Table 5.9. List Evaluation showing the mean results (rounded to 3 decimal places) for the One Category and Multiple Category approaches across the Seen Testing Categories. For each category the best approach is in bold

Category	Method	Metric							
		Mean Precision				*Recall*			
		K=5	*K=10*	*K=15*	*K=20*	*K=5*	*K=10*	*K=15*	*K=20*
Dog Food	One Category	**0.110**	**0.159**	**0.203**	**0.225**	**0.011**	**0.029**	**0.042**	**0.057**
	Multiple Category	0.055	0.122	0.161	0.178	0.008	0.028	0.039	0.052
Eye-liner	One Category	**0.058**	**0.085**	**0.108**	0.111	**0.015**	**0.028**	**0.041**	0.053
	Multiple Category	0.037	0.059	0.099	**0.117**	0.007	0.019	0.037	**0.054**
Lip Balm	One Category	0.092	0.153	0.174	0.194	0.014	0.031	0.040	0.052
	Multiple Category	**0.140**	**0.167**	**0.217**	**0.230**	**0.025**	**0.038**	**0.065**	**0.078**
Nail Polish	One Category	0.120	0.165	0.186	0.183	0.021	**0.044**	0.060	0.074
	Multiple Category	**0.143**	**0.185**	**0.197**	**0.190**	**0.024**	**0.044**	**0.063**	**0.083**
Perf-ume	One Category	**0.102**	**0.179**	**0.205**	**0.244**	**0.026**	**0.050**	**0.076**	**0.104**
	Multiple Category	0.042	0.076	0.107	0.126	0.016	0.031	0.045	0.063
Sham-poo	One Category	**0.043**	**0.096**	**0.170**	**0.187**	**0.014**	**0.028**	**0.054**	**0.066**
	Multiple Category	0.033	0.058	0.142	0.148	0.009	0.018	0.047	0.059
Tooth-paste	One Category	0.057	0.109	0.124	0.155	**0.009**	**0.026**	**0.038**	**0.054**
	Multiple Category	**0.073**	**0.132**	**0.156**	**0.183**	**0.009**	0.021	0.033	0.053

Although the results in Table 5.9 may portray that many of the results are better for the One Category approach, in fact, it only performs significantly better 6/56 times and 4/56 times when the p-value level is 0.1 and 0.05 and never when the level is 0.01. The Multiple Category approach also only performs significantly better 3/56 times and 2/56 times when the p-value level is 0.1 and 0.05 and never when the level is 0.01. Furthermore, for both the approaches, statistical significance across the mentioned values is only ever achieved in 2 categories: Perfume for the One Category approach and Lip Balm for the Multiple Category approach. The results in the table show that the Multiple Category approach performs similarly to the One Category approach for the List Evaluation. To summarize, the Multiple Category approach performs similarly to the One Category approach in the Binary and List Evaluation approaches. Due to this, RQ 3 is partially addressed, which asks whether MTL can detect future customer needs for product categories the model has seen and not seen during training.

Table 5.10. P-value (rounded to 3 decimal places) for the Mann-Whitney U Test of results from One Category approach vs Multiple Category approach for List Evaluation

Category	Metric							
	Mean Precision				*Recall*			
	K=5	*K=10*	*K=15*	*K=20*	*K=5*	*K=10*	*K=15*	*K=20*
Dog Food	0.123+	0.283+	0.393+	0.878+	0.393+	0.82	0.436+	0.939+
Eye-liner	0.739+	0.49+	0.315+	0.444+	1.0	0.939+	0.796	0.82+
Lip Balm	0.123	0.082	0.912+	0.401+	0.393	0.012	0.353	0.014
Nail Polish	0.684	0.785+	0.739	0.789+	0.796	0.789+	0.971	0.85+
Per-fume	0.123+	0.229+	0.043+	0.046+	0.052+	0.047+	0.029+	0.073+
Sham-poo	0.796+	0.599+	0.315+	0.222+	0.436+	0.47+	0.436+	1.0
Tooth-paste	0.529	0.656	0.684	0.909+	0.529	0.703+	0.631+	0.88+

+ test was in favour of the One Category approach

5.2.3.3 Multi-Task Learning Approach Comparison For Unseen Categories - Binary Evaluation

This section shows the results of the Multiple Category MTL approach for the Unseen Categories using the Binary Evaluation (discussed in Section 3.5.2). The results of this evaluation are seen in Table 5.11. It would be an unfair test to compare the results from the Seen and Unseen Testing Categories using a statistical test because some categories are predicted with better performance than others - therefore making the test unfair. That being said, the results in the table are not too different from the Seen Testing Category results in Table 5.7. This shows that (according to the Binary Evaluation) the MTL model can still

predict future customer needs for a category it has not seen during training with relatively similar performance to ones it has seen during training. This is very useful because even if there is no ground truth category data available for a product category, future customer needs for it can still be predicted on Reddit. To further emphasise the fact that the results from the Seen and Unseen Testing Categories don't differ much from each other, the distribution of F1 scores for these categories types are plotted using the Multiple Category approach in Appendix R. The appendix shows that the two distributions of the F1 scores (i.e. for the Seen and the Unseen Testing Categories) don't deviate much from each other.

Table 5.11. Binary Classification Evaluation showing the mean F1 Scores (rounded to 3 decimal places) for the Multiple Category approach across the Unseen Testing Categories.

Category	Method
	Multiple Category
Beer	0.086
Cereal	0.107
Coffee	0.086
Cookie	0.084
Pizza	0.118
Popcorn	0.109
Soda	0.070
Soup	0.083

5.2.3.4 Multi-Task Learning Approach Comparison For Unseen Categories - List Evaluation

This section shows the results of the Multiple Category MTL approach for the Unseen Categories using the List Evaluation (discussed in Section 3.5.1). The results of this evaluation are seen in Table 5.12. As with the previous section (i.e. Section 5.2.3.3), it would not be fair to compare the results from the Seen and Unseen Testing categories using a statistical test, however, the results are not too dissimilar from the Seen Testing Category results in Table 5.9. Because the information seen in tables 5.9 and 5.12 can be difficult to summarize, the mean result across all the categories of the Multiple Category approach is also shown for each of the 80 Unseen Testing Categories and the 70 Seen Testing Categories results for each metric across each value of K in Table 5.13. As seen in the tables, the performance for the Seen and Unseen Testing Categories are very similar. This shows that (according to the List Evaluation) the MTL model can still predict future customer needs on a category it has not seen during training with relatively similar performance to ones it has seen during training. To further visualize the fact that the results from the Seen and Unseen Testing Categories don't differ much from each other, the distribution of Mean Precision and Recall scores are plotted across all the mentioned values of K for these categories types using the Multiple Category approach in Appendix S. Again, this clearly shows the results are similar.

Table 5.12. List Evaluation showing the mean results (rounded to 3 decimal places) for the Multiple Category approach across the Unseen Testing Categories

Category	Method	Metric							
		Mean Precision				*Recall*			
		K=5	*K=10*	*K=15*	*K=20*	*K=5*	*K=10*	*K=15*	*K=20*
Beer	Multiple Category	0.075	0.097	0.120	0.125	0.016	0.032	0.052	0.068
Cereal	Multiple Category	0.052	0.082	0.125	0.131	0.007	0.020	0.036	0.043
Coffee	Multiple Category	0.107	0.120	0.136	0.141	0.017	0.038	0.052	0.064
Cookie	Multiple Category	0.040	0.119	0.149	0.163	0.014	0.040	0.071	0.091
Pizza	Multiple Category	0.075	0.110	0.143	0.159	0.013	0.027	0.045	0.057
Popcorn	Multiple Category	0.088	0.133	0.126	0.142	0.014	0.033	0.047	0.066
Soda	Multiple Category	0.090	0.118	0.128	0.126	0.022	0.040	0.062	0.075
Soup	Multiple Category	0.072	0.155	0.172	0.180	0.012	0.031	0.043	0.060

5.2.3.5 Multi-Task Learning Approach Comparison - Summary

To summarize, the Multiple Category approach and One Category approach perform similarly when assessed on Seen and Unseen Testing categories. This addresses RQ 3, that MTL can be employed to train a generalizable model for which future customer needs can be predicted for a product category the model has seen and not seen during training. As seen in this section, although the two approaches perform similarly (i.e. One Category and Multiple Category), it is recommended to use the Multiple Category MTL approach. This is because it can provide predictions for categories it hasn't seen during training. It also only needs to be trained once in comparison to training multiple individual One Category models (leading to lower computational costs).

5.2.4 Further Examination of Results

This section performs a deeper analysis of the results to see where certain optimizations can be made to improve the model's capability. Specifically, the following are assessed: 1) lead times the keyphrases are predicted to be future customer needs on Reddit before they trend in the TCN ground truth dataset (Section 5.2.4.1); 2) analyze the potential for future optimizations of the model by examining the effectiveness of the validation approach (Section 5.2.4.2); 3) give a deeper and alternative insight into the capability of the model by displaying different performance graphs e.g. precision-recall curve (Section 5.2.4.3); and 4)

Table 5.13. Seen and Unseen Testing Category mean results across all the categories used in the analysis for List Evaluation (rounded to 3 decimal places). For each metric the result from the best approach is in bold

Category	Method	Metric							
		Mean Precision				*Recall*			
		K=5	*K=10*	*K=15*	*K=20*	*K=5*	*K=10*	*K=15*	*K=20*
Seen	Multiple Category	**0.075**	**0.014**	0.114	0.029	**0.154**	0.047	**0.167**	0.063
Unseen	Multiple Category	**0.075**	**0.014**	**0.117**	**0.033**	0.137	**0.051**	0.146	**0.066**

investigate the misclassifications made by the model to see if there are any recurring mistakes so that changes can be made to mitigate them e.g. adding new ML features to address an error (Section 5.2.4.4). It is of note that in this section, the Multiple Category MTL approach is examined based on the findings of Section 5.2.3.5. Therefore, the discussion in this section assumes that this model is applied.

5.2.4.1 Lead Times of Future Customer Needs

This section observes the lead times the social media model detects future customer needs in the TCN ground truth dataset. To do this, all the keyphrases the MTL model correctly predicts as future customer needs are obtained i.e. in the TCN dataset 1-3 years in the future. These keyphrases are then checked to see how far out they are from the date of prediction to the date they are in the TCN dataset. This is done across all the 15 product categories tested in the analysis and across all 10 runs of each category - each category tested in the analysis is run 10 times (as described in Section 5.2.1). In the analysis, only the date in the TCN dataset nearest to when the keyphrase from the MTL model was predicted is recorded e.g. if the model predicted "charcoal" on 2018-01-01 and "charcoal" appeared in the TCN dataset on 2019-01-01, 2020-06-01 and 2021-03-01 the gap is considered to only be 1 year as 2018-01-01 is 1 year away from 2019-01-01. When checking how far out the keyphrases are from the month of prediction to the date they are in the TCN dataset, dates that are closer than 1 year away are included even if they are detected as future customer needs by the model e.g. if the model predicted "coconut" on 2018-01-01 and "coconut" appeared in the TCN dataset on 2018-06-01, 2019-01-01, 2020-06-01 and 2021-03-01 the gap is considered to only be 5 months.

Figure 5.7 shows a kernel density estimation plot of these lead times.[17] Although a lot of these future customer needs are detected before 5 months there are a lot found past 2 years in advance. Such lead times would be highly beneficial for companies to identify before these needs start to become popular in the marketplace.

[17] This is shown instead of a histogram for the same reasons as detailed in Appendix R and S (more visually intuitive). The same library as in Appendix R and S is also used to generate the plots i.e seaborn.

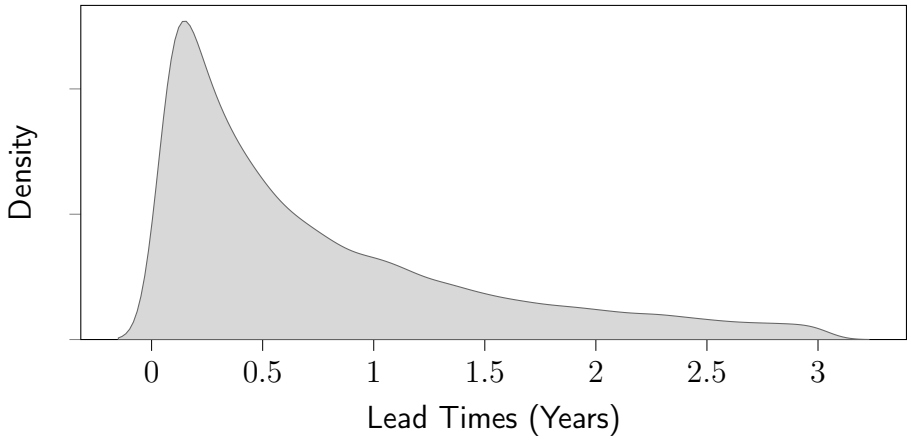

Figure 5.7. Lead times (years) of detecting future keyphrase customer needs before they are addressed in the marketplace i.e. TCN dataset

5.2.4.2 Future Optimizations

This section discusses the limitations of the validation procedure, which hinders the approach at finding future customer needs with increased performance.

As discussed earlier in the evaluation (i.e. Section 5.2.1), the output probability threshold parameter of the model is validated by finding its value which optimizes the F1 score for predicting future customer needs. The data used to train and validate the model comes from 2014-01-01 to 2014-12-31, while the data used to test the model comes from 2018-01-01 to 2018-12-31. These splits in the training and testing times are required so that no instance overlap occurs (discussed in Section 5.2.1). When applying the validated probability threshold parameter value (estimated from 2014) to the test data (in 2018), the model can be investigated based on how well it performs for each category by comparing the F1 score it generates to the F1 score represented by the optimal probability threshold parameter value which can be found in the test data. This optimal probability threshold parameter value can be found the same way the validated threshold parameter value is found, however, it requires the ground truth label which is not provided in real-world scenarios, hence why it is not used already during testing. To show how well the assumptions hold, the difference between the optimal F1 score found in the test data compared to the F1 score estimated during validation for each category is recorded.

These differences are shown in Table 5.14, which tracks the mean difference (rounded to 3 decimal places) for each category each month between 2018-01-01 to 2018-12-01 across the 10 runs of the algorithm. There is quite a lot of room for improvement in the validation parameter estimation approach used in this analysis, with some mean differences being as high as 5% F1. An area where this could be improved is in the way in which the data is split into training and validation sets, which is not conventional given the nuances of the training data having overlapping time series (described in Section 5.2.1). Improving this process is an area of future work. As discussed earlier in the evaluation (Section 5.2.1), the probability threshold is the only hyper-parameter tuned in this chapter for computational/-time purposes. Tuning other hyper-parameters (e.g. inputs into MINIROCKET) is also an area of future work that would allow for increases in model performance.

Table 5.14. Mean difference (rounded to 3 decimal places) between the optimal F1 score on the test set and the chosen F1 score in the validation set for the Probability Threshold parameter

Category	Mean F1 Difference
Beer	0.032
Cereal	0.029
Coffee	0.034
Cookie	0.042
Dog Food	0.024
Eyeliner	0.035
Lip Balm	0.036
Nail Polish	0.042
Perfume	0.046
Pizza	0.027
Popcorn	0.036
Shampoo	0.034
Soda	0.054
Soup	0.050
Toothpaste	0.021

5.2.4.3 Performance Graphs

This section plots the results of the Binary Classification Evaluation using some alternative visualizations/metrics other than the F1 score (used in Section 5.2.2 and Section 5.2.3). Specifically, the PR Curve and the ROC Curve are plotted - both are typically used to show the performance of binary classification at a range of probability thresholds [333]. The PR Curve is built by plotting precision-recall pairs obtained using different thresholds from a probabilistic (or other continuous-output) classifier [333]. The ROC Curve is built the same way using thresholds from a classifier, however, by plotting true-false positive rate pairs instead. When plotting, the AUC for both the PR Curve (i.e. AUC PR [334]) and ROC Curve (i.e. AUC ROC [334]) is also reported. This is done as these metrics have been used a lot to compare models [334] while providing a different view of the results other than the F1, which has been used extensively in this chapter. Although both the PR Curve and the ROC Curve are plotted, the PR Curve is known to be better for binary classification scenarios where the data is imbalanced [335] (as in this chapter). The main reason for this is due to the inclusion of True Negatives in the False Positive Rate for the ROC Curve and the mindful avoidance of this in the PR Curve. By including the True Negatives in the calculation, importance is given to situations where the classifier predicts an instance to be the negative class when it is in fact the negative class, which is highly common in imbalanced classification tasks where the focus should instead be on predicting the positive class. This leads to a random baseline classifier achieving an AUC ROC score of 0.5 regardless of the distribution of the output class, whereas the AUC PR for the random classifier moves with the distribution of the output class [335].

Figure 5.8 shows the PR Curve for the MTL model and a random classifier. The AUC for

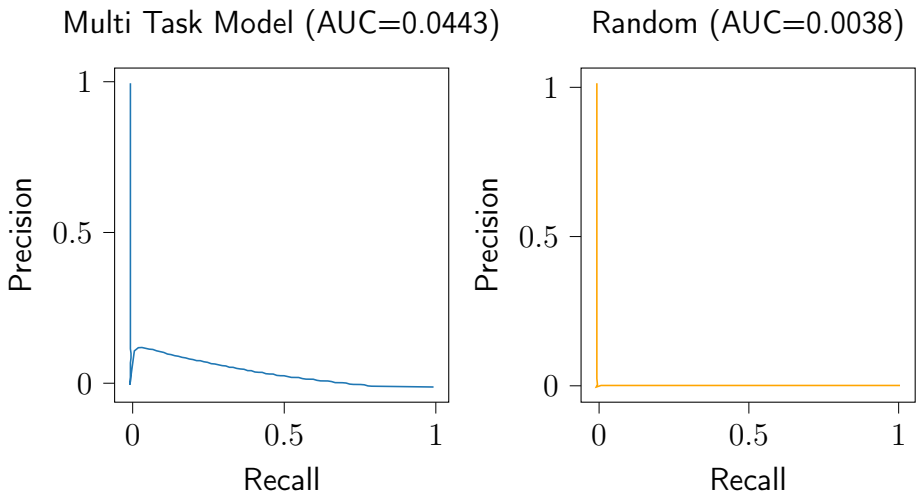

Figure 5.8. Precision Recall Curve for the MTL Model and a Random Classifier. The PR AUC score (rounded to 4 decimal places) is also shown for each classifier.

each of the PR curves is also shown in the figure (rounded to 4 decimal places). Here the random classifier generates probability values from a continuous uniform distribution in the range of 0 to 1.[18] Specifically, it generates these values from the output of the keyphrase selection process (i.e. Section 5.2) and therefore assumes no MTSC takes place. This random classifier generates an AUC PR score of 0.0038 which approximates the ratio of the initial class distribution i.e. 260:1 (detailed in Section 5.2.1). As discussed in the AUC PR literature, this approximation is expected [335]. The MTL model has an AUC PR score of 0.0443 which outperforms the random classifier by some margin (≈11.5 times better). This shows that a significant amount of learning is taking place during MTSC (i.e. Section 5.1.3, Section 5.1.4 and Section 5.1.5), as the random classifier aimlessly picks keyphrases the MTL approach learns features on Reddit from. Of additional note in the figure is that the precision and recall both drop to zero across some probability thresholds in the PR Curve for the MTL model. This is because there are a number of keyphrases which are in the TCN dataset (i.e. ground truth), however, are never selected as candidate keyphrases during the Text Processing & Keyphrase Selection process (i.e. Section 5.1.2). These are accounted for when performing the evaluation, however, are given a predicted probability score of 0 as they never have a chance of being predicted as the positive label. Although barely visible in the figure, the same occurs for the random classifier as these keyphrases are not chosen during keyphrase selection.

Figure 5.9 shows the ROC Curve for the MTL model and a random classifier. The AUC for each of the ROC curves is also shown in the figure (rounded to 4 decimal places). The same random classifier used for the PR curve is used in this figure. The random classifier achieves an AUC ROC score of 0.4159, which is contradictory to the literature that random achieves a score of 0.5 [335]. This occurs as there are a certain number of ground truth keyphrases that are never found in the keyphrase selection process (as discussed earlier in this section). This is the reason why the ROC curves for both the MTL and random classifiers dramatically shoot up at the threshold 0. As with the PR Curve, the ROC curve for the MTL model (0.723) is better than the random classifier (0.4159). However, as

[18] https://docs.scipy.org/doc/numpy-1.15.0/reference/generated/numpy.random.random.html - last accessed 07/06/2024

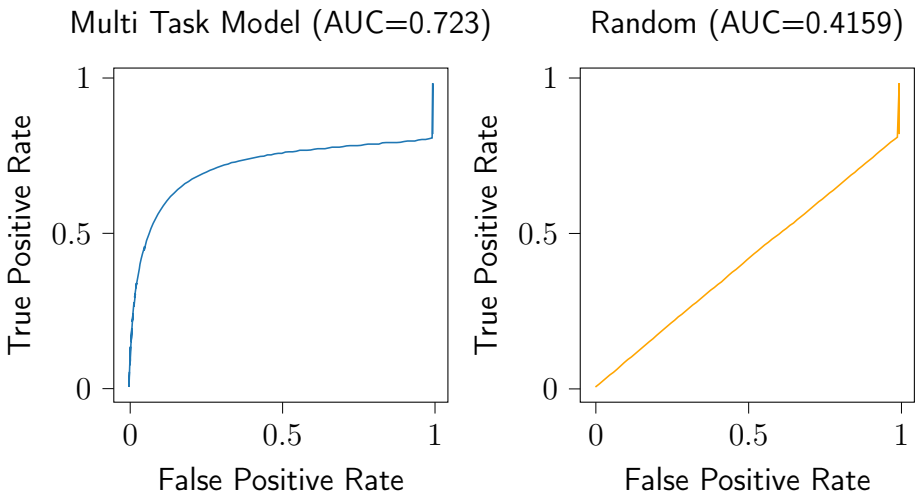

Figure 5.9. Receiver Operating Characteristic Curve for the MTL Model and a Random Classifier. The ROC AUC score (rounded to 4 decimal places) is also shown for each classifier.

discussed, the figure does not paint the full picture of the real difference between the two classifiers.

5.2.4.4 Misclassifications

This section aims to highlight instances where the model made the largest prediction errors. This is done to illustrate where there are some inherent problems associated with it or where it can be improved to better understand future work directions to improve it. Specifically, instances where the model is wrong in misclassifying a keyphrase as a "future customer need" when it isn't one and vice versa (i.e. not a "future customer need" when it is one) are highlighted. To find these instances, the probability output omitted by the MTL model for each keyphrase and its associated ground truth label in the TCN dataset are recorded. Table 5.15 records the most probable keyphrases the MTL model predicts as future customer needs (i.e. highest prediction probability output) which are not future needs (i.e. did not appear in the TCN dataset 1-3 years in the future). This therefore shows the keyphrase instances the MTL model most thinks are future customer needs, however aren't. Table 5.16 then records the most probable keyphrases the MTL model predicts are not future customer needs (i.e. lowest prediction probability output) which are future needs (i.e. appear in the TCN dataset 1-3 years in the future). This shows the keyphrase instances the MTL model most thinks are not future customer needs, however are. For this table, to show the most probable keyphrases the MTL model predicts are not future needs, the keyphrases that have the lowest probability output omitted by the model are shown not just ones given a probability score of zero for this evaluation (detailed in Section 5.2.4.3). For the information in both Table 5.15 and 5.16, the keyphrases with the highest mean predicted probability output across all runs of the model are reported i.e. the MTL approach is run 10 times for all product categories (as described in Section 5.2.1).

It is important to reiterate that this model predicts future customer need keyphrases in the TCN dataset. As detailed in Appendix C, the TCN dataset groups customer needs into two main definitions: 1) direct and 2) indirect. These two definitions may provide context

Table 5.15. The top 10 predicted keyphrases the MTL model thinks are future customer needs (ranked by prediction probability output), however aren't (i.e. are not in the TCN dataset 1-3 years in the future)

Category	Keyphrases
Beer	garlic, onion, corn, grape, tomato, fried, ginger, baked, liquid, funky
Cereal	flax, olive, mashed, fatty, whey, dressing, nutritional, walnuts, melon, cheesecake
Coffee	unsweetened, avocado, dairy, tomato, sour, blueberry, extract, veggie, olive, pepper
Cookie	roasted, toasted, grape, marshmallow, beef, tart, garlic, crust, glaze, buttery
Eyeliner	precision, voluminous, amazonian, sweet, cosmetic, pomade, enhance, facial, glossy, clay
Lip Balm	moisturising, smooth, hydrate, healthy, neutral, antioxidant, soft, facial, agave, delicious
Nail Polish	acrylic, cream, enamel, finish, lemon, solvent, berry, neutral, acetate, wine
Perfume	aromatic, amber, apricot, bergamot, oily, hydrating, sandalwood, grapefruit, intense, frankincense
Pizza	almond, coconut, pudding, roasted, shredded, vanilla, pretzel, biscuit, fatty, balsamic
Popcorn	roasted, savory, hummus, cucumber, baked, nuts, tortilla, broccoli, watermelon, nutritional
Soda	almond, creamy, potassium, optional, spicy, cocoa, coconut, unsweetened, maple, stevia
Soup	fatty, nutritional, cashew, radish, eggplant, rosemary, toasted, steamed, rotisserie, crispy

to why a certain customer need is in either Table 5.15 or Table 5.16. For example, Table 5.16 states that "fatigue" is a customer need for Cereal products. This may seem strange, however, Appendix C states that direct needs can also be benefits a user overcomes from using the product and not just from using the product (e.g. energy).

In Table 5.15 (i.e. keyphrases wrongly predicted as future customer needs), there appears to be only one dominant issue where the model is making a mistake i.e. misclassifying. This is that the model is predicting irrelevant keyphrases for a product category e.g. "garlic" for the Beer category, "beef" for the Cookie category or "voluminous" for the Eyeliner category. This occurs due to how the MTL model is trained on a wide variety of product categories. Through training this way, it learns the characteristics of a future customer need rather than needs for its own category. Therefore, if a keyphrase meets the criteria of being a future customer need according to the model, it will be predicted in the positive class e.g. mentioned often enough on social media (document frequency), has a phrase embedding which is shared in the same embedding space as previous positive instances (phrase embedding). This is thus the reason why there appears to be seemingly strange keyphrases for a particular category predicted with high confidence by the model.

Table 5.16. The top 10 predicted keyphrases the model thinks are not future customer needs (ranked by prediction probability output), however are (i.e. are in the TCN dataset 1-3 years in the future)

Category	Keyphrases
Beer	india pale, recycled, barley malt, top ferment, alcohol free, pilsen, kosher, light, alcohol, preservative
Cereal	freeze dry, artificial color, red blood, palm oil, sustainable, recycle, straight, refined sugar, saturated fat, fatigue
Coffee	freeze dry, rounded, halal, low acidity, compostable, single origin, protect, aluminium, full bodied, silky
Cookie	trans fat, wafers, sustainable, palm oil, trans, halal, colouring, sticks, corn syrup, egg
Eyeliner	vitamin e, animal testing, precise application, jojoba oil, sensitive eye, resin, felt tip, water resistant, mineral oil, drying
Lip Balm	dead skin cell, moisture loss, animal testing, gently exfoliate, sweet almond oil, regenerate, revive, stimulate, sweet almond, penetrate
Nail Polish	high shine, full coverage, animal testing, flat brush, easy application, dibutyl phthalate, highly pigmented, high gloss, phthalate, nail enamel
Perfume	perfumed body, halal, animal testing, softness, recycled, vegetarian, cedar wood, raw material, sustainable, femininity
Pizza	stuffed crust, recyclable, hard, wood fire, palm, gluten free, dry tomato, sustainable, fry, pepperoni
Popcorn	palm oil, handmade, halal, air pop, sunflower oil, trans fat, hot air, add sugar, gmo, cholesterol
Soda	halal, fat, calorie, cream, sugar, cola, red, zero, alcohol, fruit
Soup	pea, add sugar, chicken bone broth, palm, bpa, halal, extra virgin olive oil, stew, chicken, egg

In Table 5.16 (i.e. keyphrases not predicted as future customer needs), there similarly appears to only be one issue occurring. It is that certain keyphrases don't have the expected predicted probability output of being a future customer need. This can said for almost every keyphrase in the table e.g. "alcohol free" for Beer, "high shine" for Nail Polish, "cream" for Soda or "pea" for Soup. This is almost the opposite problem of the first issue mentioned in Table 5.15, which is that certain irrelevant keyphrases are predicted for a product category. However, part of this problem arises from the same first issue in Table 5.15, which is the fact that the model is finding it difficult to learn keyphrases associated with its product category due to how the model is trained during the MTL process.

There are most likely a lot of ways that the discussed issue of predicting keyphrases that are not relevant to a category can be remedied. One way could be to add more ML features that relate a keyphrase to its associated category. Some of these existing types of features include comparing a keyphrase's frequency to a background corpus (detailed in Appendix H). However, some additional ways of guiding the model to get to know that a keyphrase is related to a domain would be useful in mitigating these errors [73, 336, 337]. This could make the approach resistant to these types of issues even when employing the MTL

technique of learning customer need keyphrases from other categories i.e. as each keyphrase would have a relatedness score to its respective category.

5.2.5 Main Findings

To help guide this evaluation, two research questions were proposed at the beginning of this section: 1) To what extent can future customer needs be detected with performance useful for the purposes of product development (i.e. RQ 1)?; and 2) how can the use of MTL be employed to train a generalizable model for which future customer needs can be predicted for a product category the model has seen and not seen during training (i.e. RQ 3)? To address these questions, details on how the approach was implemented were provided along with the two training strategies used in the evaluation (Section 5.2.1): 1) One Category training - which uses the same category data to train and test the model to find future customer needs; and 2) Multiple Category training - which incorporates MTL by using multiple categories to train a model. Using the One Category approach, the model was compared against a baseline - which consisted of the approach in Chapter 4 (Section 5.2.2). The One Category approach was only used here as it was desired to test if the general ML approach is better than the baseline. During experiments, it was shown that the One Category model significantly outperformed the baseline in two of the described evaluation approaches across multiple categories used in the baseline analysis. By illustrating this, it was shown that the approach described in this chapter can predict with better performance than the baseline - thus showing improvements from Chapter 4. Along with Chapter 4, this contributes to addressing RQ 1 that future customer needs can be predicted using Reddit. This also directly addresses RC 2 that keyphrases can be predicted using an ML-based approach. The Multiple Category MTL approach was then evaluated against the One Category approach to test if the approach of training on multiple categories is similar to the category used to test the model. This was shown to be the case as the Multiple Category approach performed highly similarly to the One Category approach for both Seen and Unseen testing categories across two of the evaluation approaches for multiple categories used in the experiment. By doing this, RQ 3 was addressed. This also addresses RC 4, which states that a model that incorporates MTL would be made to accurately predict for a category the model did and didn't use in the training process.

Throughout the evaluation some of the reported results may seem underwhelming drawing criticism e.g. F1 scores ranging from 7-14% for the best-performing MTL model across 15 product categories used to evaluate it. Similar criticisms are given to tasks with a high data imbalance that are also difficult to predict. Tasks of this kind are seen in a various range of topics in the ML literature, including hashtag prediction [338], intrusion detection [339], image classification [340], predicting responses to intensive Post-Traumatic Stress Disorder (PTSD) treatment [341], predicting treatment discontinuation in patients with diabetes [342], classification of unstructured medical notes [343], etc. Depending on the level of imbalance, these tasks can achieve similar performance to the one addressed in this chapter. In a particular study addressing the task of virality prediction of hashtags [338] with a class imbalance of 15:1, the best two models achieved an F1 score of 36.28% and an AUC PR score of 30%. In a study on intrusion detection [339], a model achieved an

AUC PR score of 20.51% at detecting blacklist intrusions with a label distribution ratio of ≈166:1. A classification model identifying images on Wikipedia achieved a mean F1 score of 26.7% across 31 labels which had a positive label percentage of 5.71-7.55% (depending on the dataset used). A model predicting Treatment Discontinuation (TD) for diabetes patients achieved an AUC PR score of 8.1%, 22.8% and 29% a respective 2, 3 and 4 months into treatment with a positive to negative label ratio of ≈30:1 in the training set and ≈26:1 in the test set. Finally, a text classification model predicting medical notes into 16 different classes obtained a mean AUC PR score between ≈10% and ≈90% depending on the prevalence of the disease (i.e. label distribution), with lower disease prevalence obtaining lower AUC PR scores. These studies show that models that perform difficult tasks with a high data imbalance generally achieve low performance (if evaluated correctly with high-quality ground truth data and suitable metrics). Although these studies build low-performing models, they are still useful as they perform necessary tasks e.g. predicting responses to intensive PTSD treatment [341]. The same can be said for the area of research addressed in this chapter and in Chapter 4 (i.e. predicting future customer needs), which has been talked about for many years in various studies in the business literature [76, 344–346].

5.3 Summary & Discussion

As pointed out in Chapter 2, there is a lack of approaches in the literature that predict future customer need keyphrases. Therefore, this chapter outlines an approach to the problem that improves on Chapter 4 by using supervised ML over Reddit data to predict keyphrase instances as future customer needs. Specifically, this is done by framing the problem of extracting customer needs from Reddit as a binary keyphrase classification problem where candidate keyphrases are classified at each Fixed Time Window. 15 individual corpora each representing product categories were collected by only considering posts that contain the presence of defined keyphrase(s) likely to discuss the category of analysis e.g. the defined keyphrases "'cookie" and "biscuit" make up the Cookie category. The posts from each of the categories were then preprocessed and candidate keyphrases from them were selected for the classification task. 1263 features for each of the candidate keyphrases in each product category were then generated - each coming from 10 families of features e.g. frequency-based, product-based, sentiment-based, user-based, etc. Each feature is in the form of a univariate time series, therefore associating each candidate keyphrase instance with a multivariate time series data type. The process of adding the ground truth label to each candidate keyphrase instance across each of the 15 product categories is then described. To do this, the TCN dataset was utilized - a list of trending keyphrase needs occurring in products each month from 2011-2021 that spans multiple product categories in the area of CPG. This dataset was used to indicate whether a candidate keyphrase will appear as a top customer need to be addressed in real products 1-3 years in the future. Without the use of this dataset, supervised ML would not have been able to be performed for the task of classifying future customer need keyphrases. Finally, the MTSC algorithm was detailed (i.e. Mini Rocket followed by Linear Ridge Regression), which was used to learn the relationship

between the candidate keyphrases and the binary output label from TCN. To evaluate the approach, 15 product categories were analyzed. In the main evaluation investigation, it was shown that the approach could detect future customer needs significantly better than the approach described in Chapter 4 and that the MTL model could detect future customer needs in categories it hadn't seen during training with performance similar to categories it had seen during training. In a further examination of the model, it was shown that it could also detect customer needs with lead times up to 2-3 years in advance of them occurring in products and be improved by a large margin by changing the validation procedure.

By showing that the ML approach can detect future customer needs better than the approach detailed in Chapter 4, further progress in addressing RQ 1 was made i.e. can future customer need keyphrases be detected? Also, RQ 3 was addressed, by demonstrating that the proposed MTL model can detect future customer needs for categories the model had seen and not seen during training.

In light of these contributions, there were also various limitations, indicating areas of future work. Although the experiments were run on powerful machines there was a lack a degree of resources to perform some highly intensive tasks. As a workaround, the use of undersampling in the training process was included and the validation of important algorithm hyper-parameters was not performed e.g. the *num_kernels* parameter to the MINIROCKET algorithm or the *alpha* parameter to the Linear Ridge Regression algorithm (as discussed in Section 5.2.1). An obvious area of future work would thus be to explore more ways to reduce computational demands to run the experiment on all the training data and optimize additional hyper-parameters, although other techniques could also be used to efficiently perform hyper-parameter optimization for large datasets [347]. After generating features for the task, no feature selection was performed. Hence, no exploration on which features most impacted the model or could be excluded to allow for potential increases in model performance or improvements in training times was carried out i.e. by reducing the input space. Such benefits may have allowed other important limitations of this study to be addressed e.g. the validation of model hyper-parameters due to the decrease in the initial feature input space. Recently, there has been an increase in the number of algorithms performing feature selection for multivariate time series data [348–351], so there's no reason this can't be performed in future studies. The work in this chapter also has the limitation of not providing an Explainable Artificial Intelligence (XAI) analysis of the classification task, which has been a growing area of research for studies using ML on social media [352]. This could be useful for providing feature-level explanations for why a particular attribute is important for the task i.e. feature importance. This would also allow for other benefits, including answering questions on why a particular feature (e.g. admiration sentiment) or feature family (e.g. sentiment) is important e.g. the feature family "sentiment" does provide a crucial role in the prediction of future customer needs. This could also help provide instance-level explanations. This would allow for analysts using the prediction model to understand why a particular customer need is being predicted e.g. "vegan" for Dog Food products is predicted to be popular in products in the future due to its rising frequency, high sentiment and diverse user base discussing it. As with feature selection, there has been an increase in the number of algorithms allowing for the explainability of MTSC tasks [353–356], so there's no reason this analysis can't be performed in future studies. Similar to the limitation of not performing an XAI analysis of the model, an examination into what

product categories most impacted the performance of the MTL model built in this chapter could be carried out. Many findings could arise from performing an analysis of this kind. These may include discovering that only a small number of product categories produce a similar performing model (in comparison to using all available categories) or finding out that some categories negatively impact model performance.

Another limitation of the work in this chapter is that although it is performed on multiple product categories, these categories are all in the area of CPG. It could be the case that the model only works in this area. A further analysis into this would need to be performed to test whether this is the case, requiring another ground truth dataset (which would be a very expensive task). Finally, although careful consideration is taken in the treatment of not allowing any bias in the experimental set-up (e.g. correctly splitting the data into training and testing sets), this analysis has the limitation in the fact that customer needs are being predicted retrospectively. This has also been the stated limitations of the results from the previous chapter i.e. Chapter 4. That being said, as in Chapter 4, this analysis is of interest nonetheless as it was able to map customer needs on Reddit to future needs in a dataset of real products i.e. TCN.

Chapter 6: Comparing ML-based and Human Product Development Approaches at Predicting Future Customer Needs

This chapter describes a user study carried out in this thesis, which evaluates whether the output produced by the approach in Chapter 5 (that predicts future customer needs) is useful to a multi-billion dollar (USD) firm developing real products. As stated in Chapter 2, the rationale for its execution is that there is a lack of such evaluations in the literature. Performing a study of this nature is a valuable contribution to this thesis as product developers are the end users of systems mining customer needs, hence carrying out a study like this which incorporates their opinions of the algorithm output is useful.

First, the methodology of the evaluation is explained and two sub-assessments are detailed (Section 6.1). The first sub-evaluation has participants from a large MNC generate lists of customer needs that they think will become required by real customers in the future (Section 6.1.3). These lists are then compared to the output predicted by the algorithm-generated list. It is of note that this sub-evaluation is similar to [53], which compares detected customer needs (in the form of document clusters) obtained from experiential interviews in a product development firm with real end users to ones predicted by an algorithm. The second sub-evaluation then has the same participants fill out a questionnaire, that asks participants to reflect on their experiences and whether they think the output predicted by the algorithm is useful (Section 6.1.4). Secondly, the results of the two sub-evaluations are provided (Section 6.2). The results from the first sub-evaluation exhibit that the algorithm and the participant predicted future customer needs have a substantial overlap (Section 6.2.1). The second sub-evaluation shows that the participants think the algorithm predicts novel and highly useful content, however, tends to produce some unuseful predictions for the purposes of new product development (Section 6.2.2).

As detailed in Chapter 1, the goal of this chapter is to address RQ 4, which questions whether future customer needs predicted by an algorithm in this thesis can be useful to companies developing real products. This question in turn addresses RC 5 which evaluates whether future customer needs predicted by an algorithm can be useful to a firm developing products.

6.1 Evaluation Methodology

Figure 6.1 outlines the evaluation methodology used to assess whether the list of future customer need keyphrases predicted by the approach in Chapter 5 is useful to a product

development team in a large MNC. The evaluation itself consists of two separate sub-evaluations (list comparison evaluation and questionnaire evaluation). The list comparison evaluation compares lists of future customer needs individually predicted by human participants to the algorithm in Chapter 5. As seen in Figure 6.1, this consists of two main steps: a) Generate List - where each participant generates lists of keyphrases that they think will become future customer needs; b) Compare Lists - where the keyphrases generated by each participant are compared to the algorithm-generated list (similarly predicting future customer needs). As a whole, this evaluation tries to determine the extent to which the algorithm-generated list can detect future customer needs identified by participants working in the MNC, who spend a lot of their time researching new product ideas and trends. The questionnaire evaluation has participants fill out a survey that asks questions on whether they found the algorithm output useful in general and to reflect on their experiences. It tries to capture any information not found in the initial experiment. For example, perhaps there are future needs predicted by the algorithm that were not generated by participants that warrant future exploration.

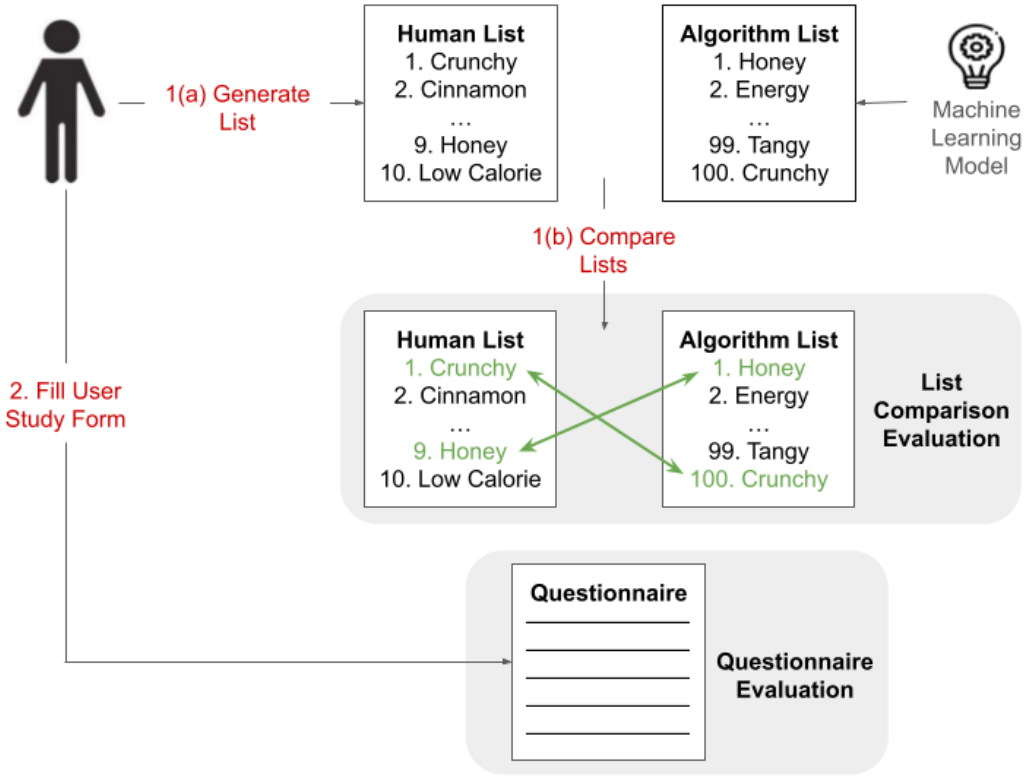

Figure 6.1. Evaluation methodology - how participants assess the algorithm predicting future customer needs

It is noteworthy that in this chapter, the evaluation design is influenced by the authors of this thesis not being allowed access to the participant's predicted future customer needs i.e. what people from a large MNC think will be popular in the future. This is because these needs may be integrated into real products made by the MNC in the future and therefore are considered proprietary information. This brought about challenges, however, these were addressed by developing a robust methodology that inhibits the organisers from accessing/knowing these needs. This is useful as this same evaluation methodology can be used in similar situations where this knowledge cannot be known by researchers. Such evidence of changes in the methodology to account for this can particularly be seen when

comparing the participant-generated list to the algorithm-generated list.

This section first provides details about the employees from the MNC who completed the experiment (Section 6.1.1). After, minor details of the algorithm used in this experiment (initially described in Chapter 5) are briefly noted (Section 6.1.2). Thirdly, the first evaluation is detailed (i.e. list comparison evaluation), as seen in Figure 6.1 (Section 6.1.3). Here, participants are asked to predict their own list of future customer need keyphrases before comparing them to the algorithm-generated list. Fourthly, the second evaluation is described (i.e. questionnaire evaluation) which has participants fill out a questionnaire form that tries to capture additional information on whether they found the algorithm output useful for the purposes of detecting future customer needs (Section 6.1.4). Finally, a pilot study is described, which was run before the main experiment (Section 6.1.5).

6.1.1 Participants

In this section, a high-level overview of the mentioned company is described along with its employees who took part in the study. The company itself is a large MNC selling CPG products, which as of 2023 has a valuation significantly above $50 billion according to Yahoo Finance.[1] As per companiesmarketcap.com, it also is within the top 20 most valuable consumer goods companies globally.[2] The area the company works in (i.e. CPG and Fast Moving Consumer Goods (FMCG)) requires a high degree of focus on New Product Development (NPD) due to the short product life cycles associated with consumer goods [358, 359]. Some of the products it makes which also include the products predicted in Chapter 5 include Dog Food, Shampoo and Toothpaste. Due to this overlap, these are the categories used in the evaluation of this chapter (detailed in Figure 6.1).

There are two groups of participants used in this evaluation: 1) experts; and 2) non-experts. Experts are the people within the company who primarily hold job titles in the fields of marketing, insights and product development. It was communicated by the MNC that although some of these participants did not specifically hold product development job titles, they did strongly participate in the devising of new product ideas that would later be developed. On the other hand, non-experts are people within the company whose primary work is not related to product development, however, still contribute towards the company coming up with new product ideas (albeit on a minor scale). These two participant groups are analyzed to understand whether the algorithm benefits one group more than the other.

A total of nine Experts were recruited for this evaluation. From these participants, there was an equal number carrying out product development on Dog Food, Shampoo and Toothpaste i.e. a total of three in each group. These participants individually tested the performance of the output predicted by the algorithm across their respective categories of interest. It is of note that all of these participants also have management roles and are therefore key in the company's decision-making process when it comes to developing new products. The product

[1] This valuation is based on the Market Capitalization which multiplies the share price by the number of shares outstanding, and also takes into account the amount of debt the company has taken on [357].

[2] https://companiesmarketcap.com/consumer-goods/largest-consumer-goods-companies-by-market-cap/ - last accessed 07/06/2024

development process used in the company can be described at a high level as a three-pronged approach: 1) Identify; 2) Qualify; and 3) Enhance. First, ideas are identified from various sources such as social media, web search queries and other competitors' launches. After, these ideas are further qualified using a deeper analysis of social media posts and web search queries, along with an analysis of the sales of an idea for which a product was centered around. Finally, an idea is enhanced with other methods, such as further exploration using the named data sources (i.e. social media and web data) along with other techniques like user interviews and surveys. These techniques are similar to the ones in the general product development literature used to generate ideas (as detailed in Section 2.1).

Three Non-Experts were used in the experiment. These participants are not key to the product development process, however, hold other business roles such as marketing and data analytics. To compensate for their lack of knowledge, these participants read consumer and product development reports about future customer needs before making predictions. Specifically, each participant read reports created by Mintel before predicting future customer needs, which have also been used in the product development literature to identify ideas [360–362]. These reports contain detailed information about customer needs for a certain product category, with some even containing information about future needs.[3] Each of the three participants was assigned a different category to generate future customer needs along with a corresponding Mintel report e.g. one participant was assigned Dog Food, one was assigned Toothpaste etc. Each of the Mintel reports had the following titles: 1) The Future of Haircare Styling & Colour 2023 (shampoo); 2) A Year of Innovation in Pet Food and Products 2023 (dog food); and 3) The Future of Oral Care 2023 (toothpaste). From the names of the titles of the reports, it is clear that the Toothpaste and Shampoo reports contain exact predictions about future customer needs, however, the Dog Food title does not. For the case of the Dog Food report the participant was asked to carefully look out for suggestions of future dog food customer needs in the report when generating their list. It is of note that by asking these participants to generate needs after reading Mintel reports, shared ideas by experts from a marketing agency are effectively captured to be compared against the algorithm.

6.1.2 ML Approach

This section details the algorithm used in this evaluation along with the data used to train and test it. As discussed, the algorithm detailed in Chapter 5 is used in this evaluation, which predicts future customer needs 1-3 years in the future from Reddit data across multiple different product categories. Specifically, the MTL version of it is used, which is trained on a wide variety of product categories. This version of the model was chosen instead of the One Category approach as it is the recommended approach to use in Chapter 5. The same version of it is used in this evaluation, which is trained on Reddit data from 2014-01-01 to 2014-12-31.

The model itself predicts future customer needs for each of the three categories used in this evaluation i.e. Dog Food, Shampoo and Toothpaste. A once-off prediction for each category

[3] https://www.mintel.com/products/reports/ - last accessed 07/06/2024

is generated so to compare to the participants' output. This prediction was generated on April 2023, which as discussed in Chapter 5 requires 3 years (or 36 months) of past Reddit data.

It is of note that a drawback of this evaluation is that the algorithm generates future customer needs in April 2023 while the participants were asked to do the exercises in November 2023. This leaves a seven-month gap between the algorithm generating needs and the participants generating and comparing the needs from the algorithm. The reason for this was due to the shutdown of the Reddit APIs in April 2023 [363], not allowing researchers to access this data beyond this point. Although this is a weakness of the overall evaluation, it is noteworthy that both the algorithm and participants are predicting future customer needs 1-3 years in the future. Therefore, this seven-month gap doesn't compromise the main aim of the experiment and in fact, makes it more difficult for the algorithm to identify needs.

6.1.3 List Comparison Evaluation: Generate & Compare Keyphrase List

This section details the first evaluation completed by participants. As shown in Figure 6.1, the first task of this evaluation is to have participants generate future customer needs. Here, each participant predicts a list of customer needs in the form of keyphrases that they think will be popular in the future. After, these lists are compared to the algorithm output using a matching process. It is noteworthy that each participant completes this list generation and comparison task individually i.e. this task is not completed as part of a group.

The purpose of this evaluation is to observe the overlap between the future customer needs predicted by the participants and the algorithm. Having an algorithm detect needs predicted by participants is of obvious interest as it could screen needs without product development teams having to spend a large amount of time finding them.

Generate

In this section, the instruction participants were provided when predicting future customer needs was to: *"generate a list of the top 10 phrases which most represent customer needs which will trend in the marketplace 1-3 years in the future"*. To provide further detail on what this question was asking, a 3-page guideline document was provided to participants before generating their lists (detailed in Appendix T).

It is of note that the output each participant generates is compared to the algorithm output in the comparison step. Due to this, when directing participants to generate future customer needs, the guidelines are written with the intention of comparing to the algorithm output in an easy and fair manner. This entails having similar boundaries for what a future customer need is to the algorithm in Chapter 5. The ground truth data used by the algorithm is the TCN dataset, therefore much of the guidelines in this section are similar to the guidelines provided by the TCN dataset (detailed in Appendix C). It is of note that the guidelines

written for the TCN dataset are based on prior research from the product development literature (detailed in Section 3.4.3).

The main areas these guidelines address are in defining: 1) what a phrase is; 2) what a customer need is; 3) what a customer need is not; and 4) what a future customer need is. When defining a phrase, the guidelines request participants to keep the total number of words in the phrase to be around 1-3 words, as these were the length of the phrases used in the TCN dataset (detailed in Section 3.4.1). When defining a customer need, as in the TCN dataset needs are split into two broad categories: 1) direct needs or benefiting descriptions (e.g. antiseptic for soap products); and 2) indirect needs or features of products (e.g. chicken for dog food products). When defining what a customer need is not, the same guidelines from the TCN dataset are provided e.g. a customer need is not a deal that comes as part of the product being sold such as "buy one get one free". Finally, participants are asked to keep in mind the scope of 1-3 years for future prediction as used by the algorithm i.e. do not predict a future customer need that can not be feasibly made in that scope of time (e.g. teeth-healing for Toothpaste).

Compare

As discussed earlier in this section, participants predict 10 future customer needs. These 10 predictions are compared to a list of 100 future customer needs outputted by the algorithm.[4] 100 needs were outputted by the algorithm (and not more) due to the time it takes participants to complete the comparison task (detailed later in this section).

A simple matching process was used to compare the participant's output to the algorithm output. For each phrase predicted by the participant, every phrase outputted by the algorithm was scanned manually to observe whether it was a match. To define what a match is, a brief half-page guideline document was provided to participants before completing the task (detailed in Appendix U). In summary, a match is defined as: 1) an exact match - the two phrases are the same; 2) a synonym match - the two phrases have the same meaning (e.g. "baking soda" and "sodium bicarbonate") or highly similar meanings (e.g. "cocoa" and "chocolate"); and 3) a match baring an insignificant word in the phrase - the two phrases are the same baring an insignificant word in the phrase which isn't necessary e.g. "paraben" and "free from paraben" would be considered a match as parabens are chemicals found in products which are widely considered undesirable (hence "free from" is inferred and therefore unnecessary).[5]

Figure 6.2 details the matching process used to compare the participant-predicted list and the algorithm-predicted list. If phrases match according to the guidelines, the location of the phrase in the algorithm-generated list is recorded (e.g. "Honey" is in the top 5 phrases in Figure 6.2 while a match is not recorded for Cinnamon"). The algorithm output is ranked by how likely a phrase is to be a future customer need, hence the reason why the ranking location is recorded. It is noteworthy that the general location of the phrase in the algorithm-generated list is recorded instead of the exact location. As in Figure 6.2, this is

[4] As detailed in Section 6.1.2, the predictions outputted by the algorithm are ranked by descending prediction probability (i.e. by how likely it is to be a future customer need.)

[5] https://thedermreview.com/what-are-parabens/ - last accessed 07/06/2024

in increments of 5 in the evaluation e.g. 1-5, 6-10, etc. This is done as the participant's predictions are considered proprietary information that could be integrated into real market products. Hence, the exact locations of the phrases in the algorithm-generated list could not be known (as it would reveal their predicted phrase). By doing this, this study proposes a way to collect valuable evaluation information about the algorithm without having to view the participant's predictions.

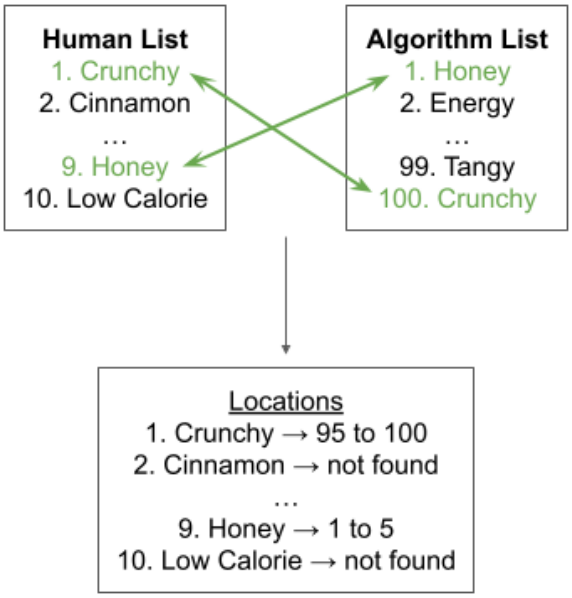

Figure 6.2. Comparison Task - how participants predictions are compared to the algorithm output

When completing the comparison task, instead of each participant completing the task themselves two independent employees did it. This was done due to the heavy time constraints of the participants in the main evaluation. These independent employees work in the company and had familiarized themselves with the comparison guidelines (i.e. Appendix U). First, each employee individually completed the matching task according to the compare output guidelines. After, both of them compared their location-matching output. Any location disagreements between each of them were discussed before a final agreement was settled, even though these disagreements were uncommon. This two-stepped process was carried out by two employees instead of one to bring more validity to the comparison evaluation. It is of note that the main participants from the Dog Food category (totalling 3 participants) did the comparison task themselves however, due to not wanting any private prediction information to leave their direct team i.e. not even to the independent employees in the same company.

6.1.4 Questionnaire Evaluation

This section details the second evaluation completed by participants. As seen in Figure 6.1, this entails having each participant complete a questionnaire/survey. The purpose of this is to ask questions about the utility of the algorithm at predicting future customer needs that wasn't captured in the first evaluation. To record participants' input, Google Forms was used, which has also been used in many other user studies [364–366].

Prior to asking questions, a brief set of primer questions was asked to familiarize the par-

Table 6.1. Questions asked to participants before completing the second evaluation exercise

It was expected (and highly relevant for the purposes of product development) to see the following 3 keyphrases ..
It was unexpected (but highly relevant for the purposes of product development) to see the following 3 keyphrases ...
It was expected (but highly relevant for the purposes of product development) to see the following 3 keyphrases ...
It was unexpected (and highly relevant for the purposes of product development) to see the following 3 keyphrases ...

ticipants with the algorithm output. For the majority of participants, this was the first time they had seen the algorithm's predictions, hence the reason why these questions were asked. In addition, participants were asked to familiarize themselves with the algorithm output for 3-5 minutes. Improving a participant's knowledge of instructions before completing a task has been shown to increase their performance [367], thus why these steps were taken. The questions asked to participants were meant to have them interactively scan through the output predicted by the algorithm. Table 6.1 shows the exact questions asked to participants. As seen in the table, each question requires the participants to view and process the output. By doing this, the participants gain knowledge about the algorithm, which can be used to answer the main questions better.

In total, 10 questions were asked to participants about the output predicted by the algorithm. These questions came from 4 families of questions, with each family having the role of collecting additional useful information about the algorithm's performance. These four families of questions are: 1) Novelty Questions; 2) List Changed Questions; 3) Unuseful Content Questions; and 4) System-Useful Questions. The **Novelty Questions** aim to understand whether the algorithm predicted output that the participant hadn't thought of. Doing this is useful as it provides the participants with recommendations that they hadn't thought of before. The **List Changed Questions** aims to ask participants if their lists would change after having read the algorithm-generated list. The answers to these questions would likely receive negative responses, as the participants thoroughly research new and emerging customer needs on a daily basis. Therefore, any non-negative responses to these questions display a positive sentiment to the algorithm. The **Unuseful Content Questions** aim to ask if there is much irrelevant output predicted by the algorithm. Unuseful content is produced by the algorithm due to the high number of irrelevant posts on social media [368–370]. The number of irrelevant posts on Reddit is high, so any non-negative answer to this question is deemed to be positive. Finally, **System-Useful Questions** try to determine the overall usefulness of the algorithm to the participants. It asks high-level questions about whether they found the output useful for the purposes of product development. Because the questions asked here try to understand whether the algorithm would have usefulness in the company's entire idea generation process, any non-negative answer is considered positive. Refer to Appendix V for all of the 10 questions asked to participants.

6.1.5 Pilot Experiment

Before carrying out the evaluation with the MNC, a pilot study was conducted with 2 independent participants not working in the MNC. As with many pilot studies [371, 372], this was done to get feedback on the evaluation methodology to assess its ease of completion so that the participants in the main experiment could more easily navigate through it with minimal intervention. Specifically, these participants were asked to generate a list of the top 10 future customer needs for Toothpaste products with the help of online web search tools such as Google. This was suitable as they had no prior experience in product development roles, unlike the participants in the main experiment. These participants then compared their predicted output to a list of 100 future customer need keyphrases generated by ChatGPT [373, 374]. Specifically, ChatGPT was asked to "generate a list of the top 100 future customer need keyphrases in the toothpaste market" - which is the same as what the algorithm in the main experiment predicted. Finally, these participants completed the questionnaire (Section 6.1.4).

As mentioned, the main purpose of this pilot study was to get feedback to assess the ease of completion of the two evaluations. At each major step of the two evaluations, these participants were asked to record any difficulties in understanding what was asked of them. Accordingly, changes were made to the guidelines and the questions asked in the questionnaire. An example of one of these changes included adding an example list of future customer need keyphrases for participants to understand what is being asked.[6] Many changes were also made to the phrasing of the questions in the questionnaire. Another purpose of the pilot study was to estimate the time it took to complete the two tasks. This was done to ensure the expert group didn't end up spending a lot of time to complete the task (due to other business priorities). Each of the two participants in the pilot study took ≈45 minutes to generate and compare the phrases and a further ≈20 minutes to complete the survey.

6.2 Results & Findings

This section aims to address RQ 4 which questions if the algorithm in Chapter 5 (which predicts future customer needs) is useful for firms developing real products. To do this, it presents the results from the described evaluations in Section 6.1. Firstly, the results according to the evaluation approach in Section 6.1.3 are detailed, which compares the participants and algorithm lists over three different product categories i.e. Shampoo, Dog Food and Toothpaste (Section 6.2.1). Secondly, the results of the questionnaire according to Section 6.1.4 are provided which captures respondents' input not addressed in the first evaluation (Section 6.2.2). It is worth mentioning that Expert and Non-Expert participant results are separated in this section to understand whether one set of participants found the algorithm more/less useful. It is also worth mentioning, however, that no statistical tests are performed comparing these two groups due to the very small sample sizes (9 Expert

[6] These examples were provided for product categories not used in the main experiment i.e. Popcorn and Eyeliner.

and 3 Non-Expert participants) [375]. More participants could not be recruited due to the level of expertise of the people involved (highly specific for the task).

6.2.1 List Comparison Evaluation

In this section, the results of the comparison between the participant and algorithm-generated phrases are detailed. First, the raw number of phrases found is provided, and after the locations of the found phrases in the algorithm-generated list are shown.

Figure 6.3 shows the distribution of phrases that are "found" and "not found" by the algorithm. Each participant generated 10 phrases, hence why there are 90 total phrases for the Expert group (9 participants) and 30 phrases for the Non-Expert group (3 participants). The figure shows that 31.11% of phrases across the Expert group are identified in the algorithm-generated list and 46.67% of phrases are identified for the Non-Expert group.[7] The fact that the algorithm can detect the participant's phrases from an output of a relatively short list (i.e. 100 phrases as stated in Section 6.1.3) is significant. This is because a lot of research was conducted by the participants in the study, especially the Experts e.g. through competitor launches, social media monitoring, web search query investigation, user interviews, etc. (as discussed in Section 6.1.1). Therefore, by detecting these predicted future customer needs, a lot of time spent researching may be reduced if the algorithm is used. It is also worth pointing out that this type of evaluation doesn't account for useful phrases predicted by the algorithm that weren't present in the participants lists (answered in Section 6.2.2). Therefore, there are likely more useful phrases than the ones that are detected as "found" in Figure 6.3 i.e. there are phrases that the participants didn't consider (and therefore are labelled as "not found"), however, are still useful to them.

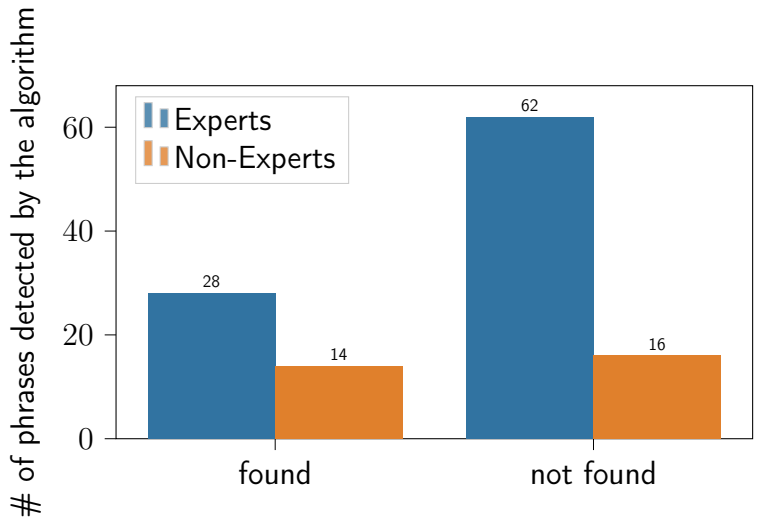

Figure 6.3. Number of participant generated phrases found by the algorithm

The rate at which phrases are identified in the Non-Expert group is higher than in the Expert group. The Non-Expert group read Mintel reports prior to predicting their lists of future customer needs and the algorithm's ground truth is formed from Mintel's product

[7] It's of note the algorithm has does not have the following raw accuracy metrics however, as it generates 100 phrases (stated in Section 6.1.2) which are compared to the participant list of 10 phrases.

database (Chapter 3), which may explain this higher overlap.

Figure 6.4 shows the locations of the detected participant's phrases in the algorithm-predicted list. As stated in Section 6.1.3, the general location is reported (in increments of 5) as the participant's predictions are considered proprietary information that could not be made available. In total, 28 and 14 of these phrases are respectively reported for the Expert and the Non-Expert groups in Figure 6.4, as these are the number of detected phrases in Figure 6.3.

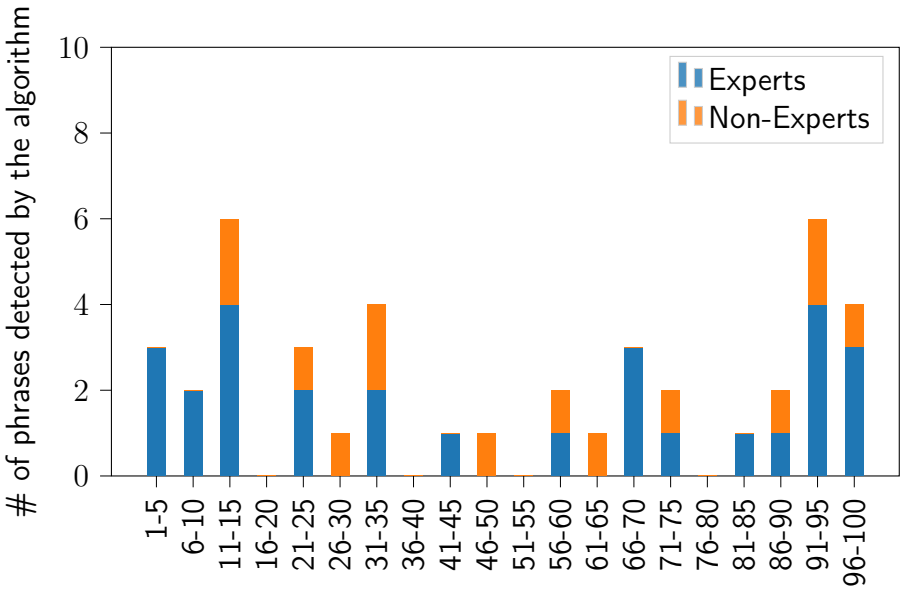

Figure 6.4. Locations of participant generated phrases found by the algorithm

As seen in Figure 6.4, there appears to be little relationship with the participant phrases being detected early in the algorithm predicted list, although there is a somewhat of a high number of detected phrases early i.e. 9 Expert predicted phrases detected between 1-15 phrases predicted by the algorithm. This is unexpected as the algorithm output is ranked by how likely it thinks a phrase will become a future customer need (detailed in Section 6.1.2). Although unexpected, this shows that the ranking of these phrases doesn't matter as much as previously thought, which indicates that the number of detected phrases may be larger if the algorithm-generated list had more phrases for comparison to the participant's lists e.g. 200 phrases. Therefore, if this evaluation was to be redone a larger algorithm-generated list would be used. The total number of phrases in the algorithm-generated list was kept at 100 to reduce the time spent by the MNC comparing the algorithm and participant lists (as stated in Section 6.1.3).

Appendix W shows the raw data of whether each phrase generated by the participants was found and what location it was found in the algorithm-generated list. It also details the product category each participant works in (i.e. Shampoo, Dog Food and Toothpaste), along with if they are an Expert or a Non-Expert. This data allows readers to explore/verify further questions about this evaluation that were not addressed e.g. is there a significant difference in the number of phrases detected between categories or are these phrases only being detected by a small number of participants?

6.2.2 Questionnaire Evaluation

In this section, the results of the questionnaire evaluation are presented. It aims to capture the information about the algorithm not addressed in the initial comparison evaluation. Specifically, it seeks to identify whether: 1) the algorithm predicts novel content (Section 6.2.2.1); 2) the participant's list would change after reading the algorithm list (Section 6.2.2.2); 3) the algorithm predicts much non-useful content (Section 6.2.2.3); and 4) the algorithm as a whole is useful to product development teams (Section 6.2.2.4). In this evaluation, some questions asked to respondents that are not focal points have been moved to appendices.

Appendix X shows the raw data of each participant's answers to the questionnaire. It details the product category each participant works in along with if they are an Expert or a Non-Expert. As in Appendix W (Section 6.2.1), this data allows readers to explore/verify further questions about this evaluation that were not addressed.

6.2.2.1 Novelty Questions

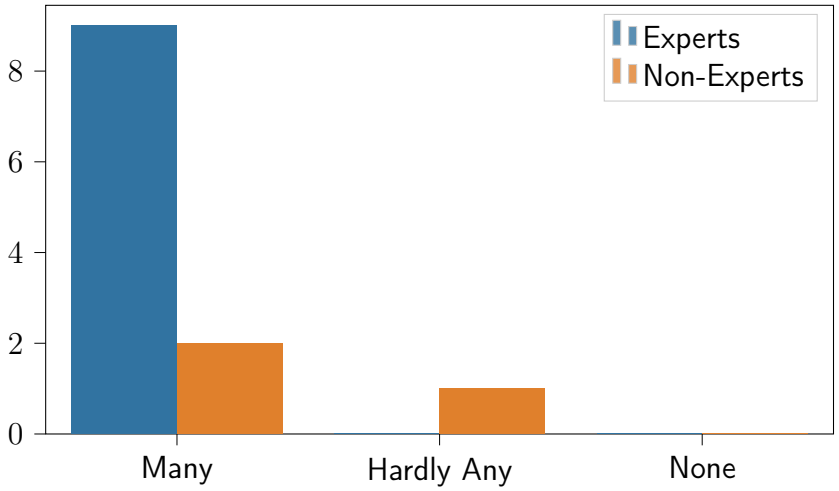

Figure 6.5. This algorithm generated list contains _____ keyphrases which weren't considered when my list was made

This family of questions seeks to discover if the algorithm produces much novel content. Figure 6.5 and 6.6 shows the main questions respondents were asked about this topic. The results show that the algorithm predicts a lot of phrases that were not considered by participants (Figure 6.5). This is quite unanimous between the two groups of participants. However, the results also show that the participants think that there are "Hardly Any" that would be considered for further investigation (Figure 6.6).[8] There are many ways participants could have deemed the keyphrases be to irrelevant and hence not worthy of further investigation. For example, three reasonable exclusions could have been: 1) the phrase is not a customer need but is a general phrase; 2) the keyphrase is a customer need but not relevant to the product category; and 3) the keyphrase is a current customer need and not a future one and is therefore irrelevant. Therefore, having almost a third of the participants consider many phrases for further investigation is deemed as a positive

[8] It is of note that the number of choices for respondents to pick in Figures 6.5 and 6.6 may depict the two sets of responses to look more contradictory than they actually are i.e. there are only three choices

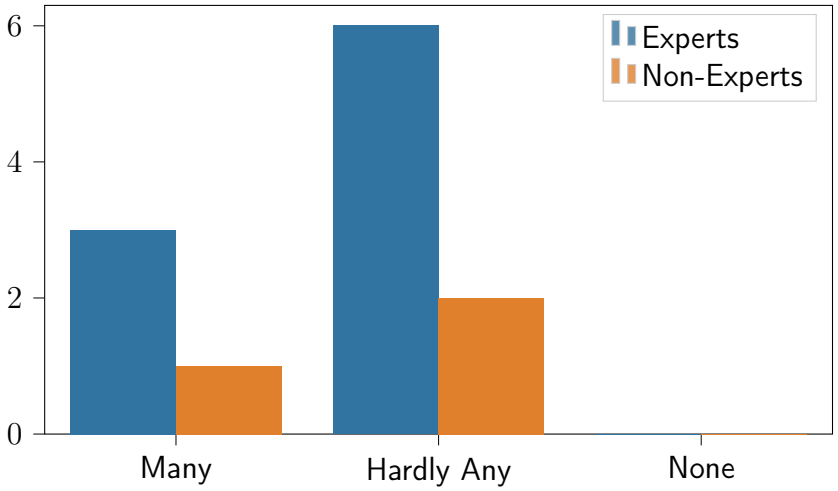

Figure 6.6. From the new keyphrases there are _____ which would be considered for further investigation

response. Appendix Y shows respondents' follow-up estimations on the number of phrases that weren't considered (i.e. follow-up to Figure 6.5) and would be considered for further investigation (i.e. follow-up to Figure 6.6).

6.2.2.2 List Changed Questions

Figure 6.7 shows the only question asked to participants about whether they would modify their list significantly after reading the algorithm output. Surprisingly, many of the respondents agreed with this statement. A total of four Experts agreed with this statement, with three being neutral, one disagreeing and one strongly disagreeing. The Non-Experts had an even more positive response to the statement, with one strongly agreeing and two agreeing. It was expected that many of the responses to this statement would have been disagree or strongly disagree, especially for the Experts who spent a lot of time researching future customer needs. The responses to this statement are therefore very positive as it shows some of the participants think that the predicted output by the algorithm is so useful that they are willing to change their list significantly.

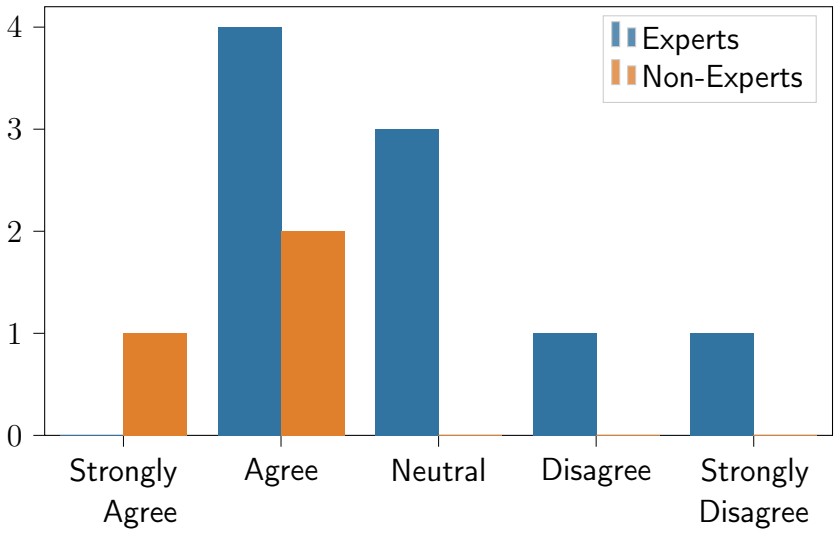

Figure 6.7. My generated list would now change significantly having read the algorithm generated list

6.2.2.3 Unuseful Questions

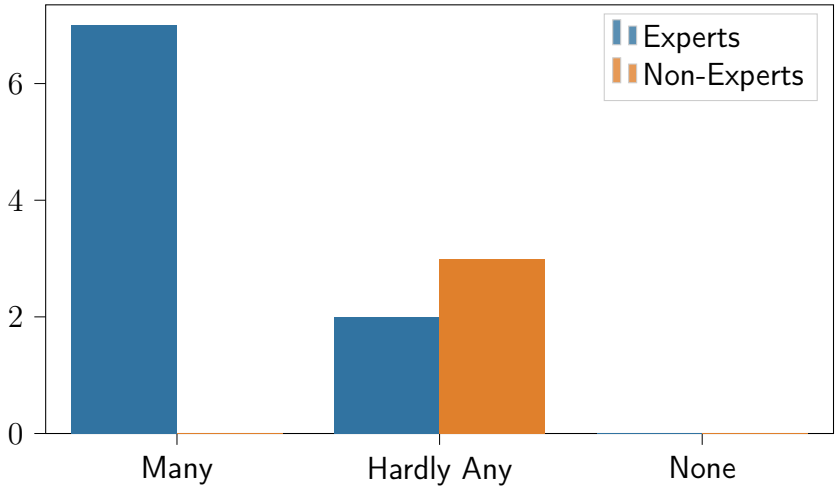

Figure 6.8. There are _____ keyphrases in the algorithm generated list which are definitely not useful

Figure 6.8 shows the question asked to participants about whether the algorithm predicted much unuseful content. As discussed in Section 6.2.2.1, unuseful content could be considered a multitude of things e.g. not customer need phrases, phrases not deemed to be future customer needs, etc. As seen in the figure, the vast majority of Experts think there are many unuseful phrases predicted by the algorithm. This is expected as there are many reasons this could be chosen. It is interesting to note that all the Non-Experts chose "Hardly Any" for their response. This therefore shows that the output predicted by the algorithm is not so irrelevant that it is evident to a non-expert, which would indicate that little of the output contains seemingly random content. Appendix Z shows the respondent follow-up estimation for the number of unuseful phrases predicted by the algorithm.

6.2.2.4 System Useful Questions

This family of questions aims to observe whether the algorithm as a whole is useful in the product development process. Figures 6.9, 6.10 and 6.11 show the questions participants were asked about this topic. The results stay consistent for the Non-Expert group, who generally state that the algorithm output is useful at making predictions and in the overall product development process. The results for the Expert group are also generally consistent, who generally seem to agree or remain neutral that the algorithm output is useful for making predictions (Figure 6.9) and that they would have preferred to see its output before attempting to make predictions (Figure 6.10). However, an increased number of Experts also state that the algorithm would not be useful in the overall product development process (Figure 6.11), which is contradictory to previous responses in the questionnaire e.g. many Experts stated that they would significantly change their list having read the algorithm-generated list.

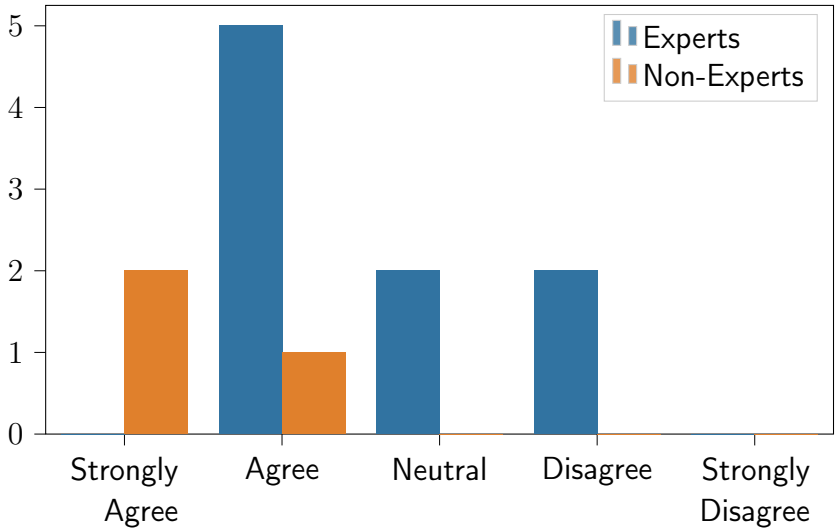

Figure 6.9. The algorithm generated keyphrases would be useful in making my list

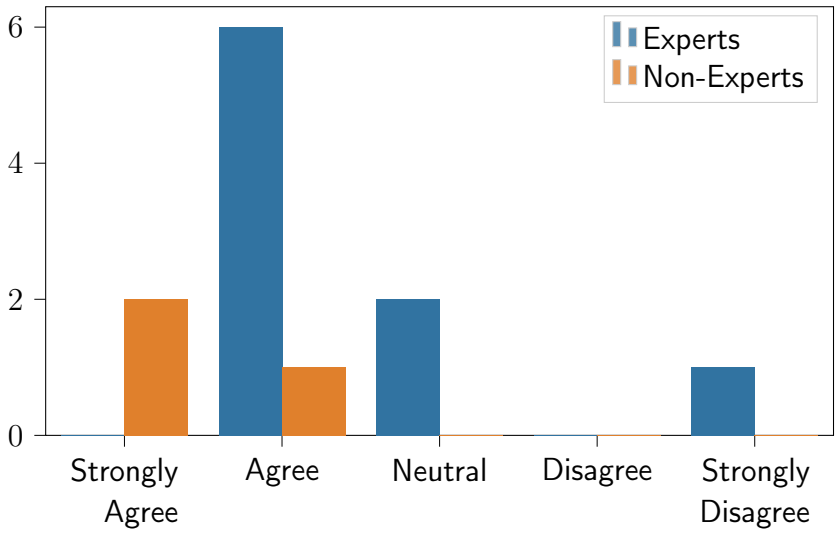

Figure 6.10. I would have preferred to see the algorithm generated list before attempting to generate my list

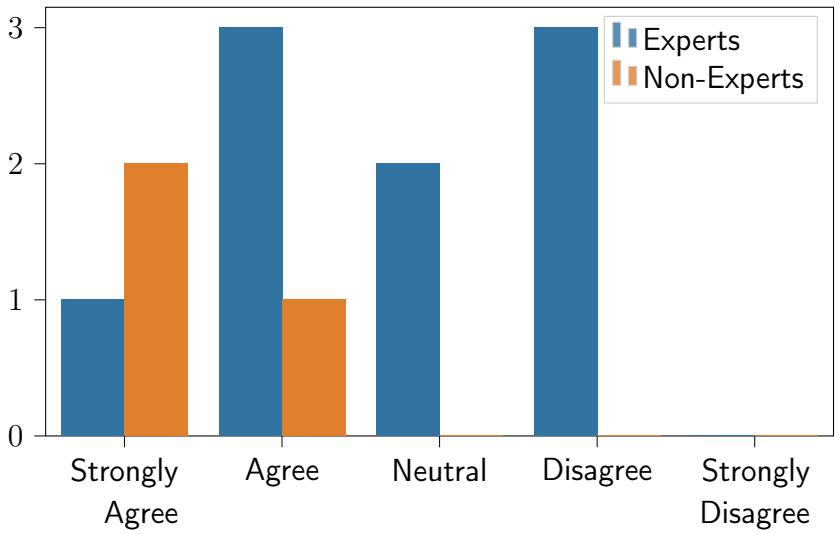

Figure 6.11. I would anticipate that having a generated list of keyphrases to assist with generating our own future lists would be helpful to the product development process

6.3 Summary & Discussion

As detailed in Chapter 2, there is a lack of user studies that evaluate the utility of algorithms at predicting future customer needs. Therefore, this chapter outlines a user-based evaluation methodology to assess whether the algorithm in Chapter 5 is useful for predicting future customer needs for product development teams working in a multi-billion dollar MNC. To carry out this evaluation, 12 participants were recruited across three product categories within the MNC i.e. Shampoo, Dog Food and Toothpaste. The user study consisted of two evaluations. First, each participant predicted future customer need lists which were compared to the algorithm-predicted list to observe whether much overlap occurred between the two predictions (Section 6.1.3). Secondly, the participants were asked questions about the algorithm-predicted list of future customer needs (Section 6.1.4). The first evaluation showed a relatively high overlap between the algorithm and participant-predicted lists, which may help reduce the amount of time taken to perform research on new needs for product developers. In the second evaluation (i.e. questionnaire), respondents stated they found the algorithm output novel and useful. By showing the relatively high overlap between the algorithm and the participant's generated lists along with the fact that respondents stated that they found the algorithm useful, RQ 4 is addressed which questions whether an algorithm predicting future customer needs would be useful to product development teams in an MNC.

In light of this, there are some drawbacks of this study. Firstly, the algorithm predicts future customer needs from April 2023, while the participants predict on November 2023 due to Reddit data unavailability (detailed in Section 6.1.2). Although this makes it more difficult for the algorithm to find these needs, it is still a flaw of the overall study. Secondly, there is some negative feedback in the answers responded by participants answering whether they found the output useful to the product development process. One point that the participants seemed to emphasize was that the algorithm produced a lot of unuseful predictions.

The main takeaway of this chapter is that the algorithm in Chapter 5 is useful for teams developing products. This improves on the evaluation in Chapter 5, which assessed the predictions based on whether they were heavily addressed in a database of new-to-market products. As seen in the evaluation of this chapter, the algorithm was useful for both groups of participants, especially the Non-Expert group. In this chapter, the Non-Expert group consists of participants who were not central to the product development process, however, are still relevant (e.g. marketers, data analysts, etc.). Many SMEs may only have these types of employees working at their firm i.e. who are not highly specific to product development. Therefore, it can be concluded that although the algorithm in Chapter 5 is useful to a large MNC, it may be even more suited to a SME.

Chapter 7: **Conclusion**

This thesis investigates techniques and evaluation approaches that address the task of predicting future customer needs on Reddit. It predicts on the product category level (e.g. cheese), which is a novelty of the thesis, in comparison to previous approaches which focus on the product model level (e.g. Charleville Spreadable with Select Cheddar). The main method used to solve the task is keyphrase ranking/classification, which orders/classifies candidate keyphrases by the extent to which they are likely to be future customer needs. Primarily, methods from text mining and ML were used to solve the keyphrase ranking/-classification task.

Chapter 2 provided an overview of the related work in the customer needs mining literature. Chapter 3 detailed two methods used to curate the ground truth datasets that consist of ranked lists of customer need keyphrases. These two methods used Mintel GNPD (a database of new-to-market product descriptions) when creating these datasets. When evaluating the datasets, it was shown that the annotations had general agreement with the gold standard annotations used for quality control.

Chapter 4 detailed the rule-based approach used to predict future customer needs. It uses text mining techniques to rank keyphrases each month by the extent to which they will be addressed in future products. The evaluation showed that it was able to outperform a simple baseline and also detect 5 out of 6 important customer needs identified by a large MNC specializing in oral care. Chapter 5 detailed an ML approach that uses MTSC techniques to classify candidate keyphrase instances represented by 1263 univariate time series features. These 1263 features come from 10 families of features, all selected for the task of predicting future customer needs. During an evaluation, this approach was shown to significantly outperform the approach in Chapter 4. A MTL model in this chapter was shown to be able to effectively predict categories it had not seen in the training process.

Finally, Chapter 6 defined a user study that investigated whether needs predicted by the ML approach can be useful to a multi-billion dollar (USD) company that develops consumer products. It does this by comparing needs predicted by the algorithm to ones predicted by experienced new product developers. It also had these participants answer a questionnaire enabling them to express their thoughts on the algorithm output and reflect on its potential usefulness. The results from this chapter showed that the algorithm and participant predicted lists had a high overlap and that participants found the algorithm output novel and useful. The rest of this chapter concludes the thesis by summarising the main contributions and discussing some directions for future work.

7.1 Summary of Contributions

This thesis aims to create and evaluate algorithms for predicting customer needs which will be addressed in the marketplace in the future. To do this, it addresses the research questions in Section 1.2. These questions are re-stated below:

RQ 1: To what extent can future customer needs be detected with performance useful for the purposes of product development?

RQ 2: How can a ground truth dataset be curated to allow for the training and evaluation of approaches that predict future customer need keyphrases?

RQ 3: How can MTL be used to train a generalizable model for which future customer needs can be predicted for a product category the model has seen and not seen during training?

RQ 4: Do professional product developers from a large MNC believe the future customer needs generated from an algorithm developed in this thesis could potentially be ones addressed in their new product lines?

The contributions that address these questions are summarized throughout the rest of this section.

RC 1 - Rule-based Approach

A rule-based approach to identifying future customer need keyphrases was addressed in Chapter 4. It works by extracting keyphrases from Reddit and then ranking them by how likely they are to be addressed in future market products. Key to the approach is a novel document filtering method (discovering potentially relevant social media content) and a keyphrase ranking method, which incorporates Google Trends data to identify keyphrases with rising frequency likely to be future customer needs.

The domain of Toothpaste was used to test the performance of the algorithm, which has been the domain of previous research in identifying customer needs using statistical techniques [53]. Given the lack of another baseline performing the task of predicting future customer need keyphrases, a simple random baseline was employed during the evaluation. The results showed the algorithm was significantly better than this baseline. Additionally, its performance was assessed in a real-world scenario by testing if it could detect 6 of the most important customer needs identified by a large MNC specializing in oral care. Here, it was shown that 5 of these needs were detected ahead of the marketplace. Furthermore, 3 of the needs (i.e. charcoal, coconut and vegan) were detected with significant lead times ahead of the marketplace, therefore potentially giving companies a competitive advantage if it was used in a business setting.

RC 1 addresses RQ 1 as it evidences that future customer needs can be detected with performance useful for the purposes of product development. This is seen with the rule-based

algorithm being able to significantly outperform a simple baseline during evaluation. Companies can massively increase profits based on the success of a new product. Therefore, improving this task to the point where it could provide better recommendations to product developers, which could prompt them to address a useful customer need in a product is of importance (i.e. RQ 1). Furthermore, the fact that the algorithm can detect 5 out of 6 important customer needs identified by a large MNC, with 3 of these needs being identified with significant lead times ahead of the marketplace, makes it inherently useful for the purposes of product development (i.e. RQ 1).

RC 2 - Machine Learning Approach

A machine learning approach to identifying future customer needs was addressed in Chapter 5. The approach uses techniques from MTSC to classify keyphrases represented by 1263 univariate time series features. These 1263 features come from 10 families of features, all useful for the task of predicting future customer needs. An MTSC model (i.e. MINIROCKET) was then built on the features that predict whether the keyphrase will appear as a trending need in future new-to-market products.

During evaluation, 15 different product categories were used to test the model. The results showed that it was significantly better for predicting future customer needs than the previous rule-based approach (Chapter 4). Additionally, it was shown that some of the lead times detected by the model (before they were heavily addressed in real products) were significant and would therefore provide competitive advantages to businesses using the model.

RC 2 addresses RQ 1 as it demonstrates that future customer needs can be detected with high performance useful for the purposes of product development. This is shown by the fact the algorithm was able to significantly outperform the approach in Chapter 4. Additionally, its use in product development teams is evidenced in Chapter 6. As discussed, improving this process even slightly can lead to large increases in profits for businesses. Therefore, by significantly improving the process, it is shown that the approach is useful for the purposes of product development (i.e. RQ 1).

RC 3 - Ground Truth Datasets

Two ground truth datasets to train and evaluate algorithms for predicting future customer needs were developed in Chapter 3 i.e. NER-T and TCN. These datasets consist of ranked lists of the most heavily addressed customer need keyphrases each month across multiple product categories. To curate such datasets, keyphrases are extracted from a large database of new-to-market products i.e. Mintel GNPD. The NER-T dataset is curated by having annotators label a sample of product descriptions from Mintel which are used to train a NER model that detects needs over the entire database. The TCN dataset uses text mining techniques to extract keyphrases that are uniquely annotated by humans. Annotators from both datasets followed detailed guidelines when labelling customer needs. These guidelines were built from the product development literature which details what constitutes a customer need. NER-T only formulates a ground truth dataset for 1 product category (i.e. Toothpaste) while TCN curates for 37 product categories within the area of CPG e.g. Toothpaste, Cereal, Dog Food, Beer, etc.

The quality of each of the datasets was also evaluated. NER-T showed a high IAA, with annotators achieving a token-level kappa score greater than 85%. Similarly, TCN achieved a high IAA with kappa scores indicating moderate to substantial agreement and performance across a gold-standard set being very high. Creating these datasets allowed the development of the algorithms in Chapters 4 and 5. Additionally, it allows researchers to test the performance of their algorithms in predicting future customer needs.

RC 3 addresses RQ 2 as it demonstrates that a ground truth dataset can be curated to allow for the training and evaluation of approaches that predict future customer need keyphrases. This was seen with the construction of both the NER-T and TCN datasets, which are both of high quality (i.e. high IAA), applicable for the training and evaluation of algorithms predicting future customer needs.

RC 4 - MTL

A model which incorporates MTL by being trained on multiple product categories to learn what characterises a general future customer need instance looks like is presented in Chapter 5. Specifically, the model is trained on 7 Seen categories and tested on 8 Unseen categories that it doesn't use in the training process. The results show that the model can predict with similar performance on categories it has seen and has not seen during training. Part of the reason for this is because the model is trained on category-agnostic features which allows it to find general patterns in what constitutes a future customer need and not just one for a particular product category.

RC 4 addresses RQ 3 as it shows that MTL can train a generalizable model for which future customer needs can be predicted for a product category the model has seen and not seen during training. This was demonstrated by the MTL model being able to predict with performance similar to the approach of being trained and tested on the same category (Seen categories) and then subsequently achieving similar performance on the categories it hadn't seen during training (Unseen).

RC 5 - User Study

A user study involving participants from a large MNC was carried out in Chapter 6 to investigate whether the needs predicted by a model described in Chapter 5 could be addressed in their new product lines. To carry out this evaluation, 12 employees from a multi-billion (USD) MNC were asked to generate lists of customer need keyphrases which they thought were going to be addressed in the marketplace in the future. These keyphrases were then compared to a list of keyphrases predicted by the algorithm in Chapter 5. Afterward, the same employees viewed the list of future customer needs produced by the algorithm and answered a questionnaire about the usefulness of its predictions. The results from the evaluation showed that there was a high overlap between the employee's lists of predicted future customer needs and the algorithm's output. Furthermore, the responses from the questionnaire showed that the employees found the algorithm output novel and useful (although it did produce some unuseful content).

RC 5 addresses RQ 4, that professional product developers from a large MNC believe the

future customer needs generated from an algorithm developed in this thesis could potentially be ones addressed in their new product lines. As discussed, this is shown by there being a high overlap between the professional participants lists and the algorithm list. Additionally, it is demonstrated by the fact that the same participants believe the output is novel and useful for the purposes of product development.

7.2 Future Work

Avenues for future work have already been discussed in this thesis at the end of each chapter. Therefore, this section discusses the main and most promising areas of future work.

7.2.1 Feature Selection

A potential criticism of the approach in Chapter 5 is that it does not perform feature selection. All 1263 features generated were used to train a MTSC algorithm at predicting future customer needs. As a result, no exploration of which features most impacted the model or could be excluded to allow for potential increases in performance or reductions in training times was carried out i.e. by reducing the input space. Recently, there has been an increase in the number of algorithms performing feature selection for multivariate time series data [348–351, 376, 377]. These algorithms perform experiments on publicly accessible datasets from the UEA/UCR time series classification and clustering repository [242, 378, 379].[1]

It is worth pointing out that these datasets are much smaller than the ones used in Chapter 5. This is seen in Figure 7.1, where the total number of data points for each dataset is plotted in a distribution (blue histogram) and compared to the average monthly total number of data points for each product category in Chapter 5 (red vertical line). The total number of data points is calculated by multiplying the number of instances, the length of each series and the number of dimensions. As pointed out in Chapter 5, the average monthly number of instances per category is $\approx$ 34,000, while the length of each series is 36 (i.e. previous months) and there are 1263 dimensions (i.e. unique features). As seen in the figure, there are a lot more data points in the datasets in Chapter 5 compared to the benchmark datasets in the UEA/UCR repository. From a time complexity standpoint, this made it infeasible to run these approaches over the datasets in Chapter 5, with these feature selection approaches taking large amounts of time to run. Specifically, two publicly available approaches [376, 377] were run.[2] However, these approaches didn't scale to the size of the data in Chapter 5.[3]

[1] A list of more than 40 multivariate time series datasets can be accessed - https://timeseriesclassification.com/dataset.php - last accessed 07/06/2024

[2] https://github.com/mlgig/ChannelSelectionMTSC - last accessed 07/06/2024

[3] It is of note that similarly powered machines were used when experimenting to that reported in [377]

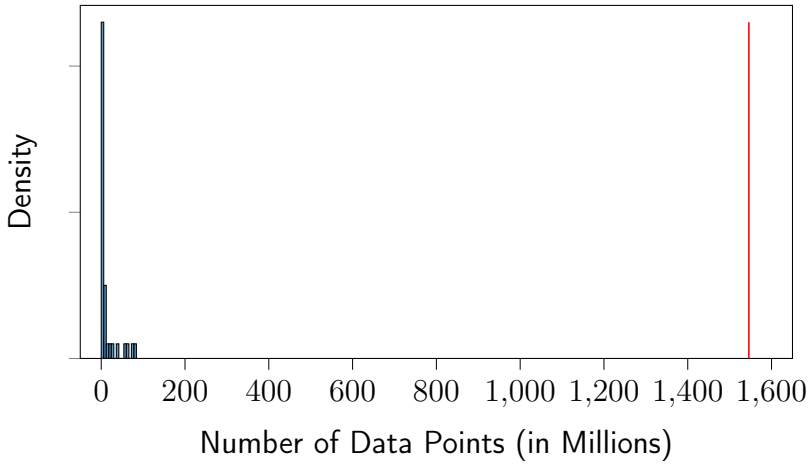

Figure 7.1. Distribution of the total number of data points in the multivariate time series UEA/UCR dataset (blue) compared to the average number of monthly data points for each product category in Chapter 5 (red)

7.2.2 XAI

Another potential criticism of the approach in Chapter 5 is that it does not perform an explainable analysis. Such an analysis would be useful for providing feature-level explanations for why a particular attribute is important for the task. However, more importantly, it would provide instance-level explanations thus allowing analysts to understand why a particular future customer need is predicted by the model e.g. increase in frequency, a high number of positive sentiment posts, etc. As with feature selection, there has been an increase in the number of algorithms allowing for the explainability/interpretability of MTSC tasks [353–356]. However, for the same reasons as with feature selection (i.e. Section 7.2.1), these approaches use datasets from the UEA/UCR repository that didn't scale to the size of the data in Chapter 5. Specifically, the interpretable MrSEQL model [353] was run on the datasets.[4]

Figure 7.2 shows a hypothetical example of the type of instance-level explanation that could be provided to users given the approach described in Chapter 5. An approach such as SHapley Additive exPlanations (SHAP) [380] could be run over a model to explain why a certain keyphrase is predicted to be in a specific class.[5] These shapley values returned could then be summed for each feature family (detailed in Table 5.2) and presented to the user. This could allow a user to better understand why a certain keyphrase (e.g. coconut for Shampoo) is predicted as a future customer need. As seen in the figure, this particular instance is predicted in a specific class because of the following features: 1) frequency, 2) embedding, 3) linguistic and 4) user.

[4] https://www.sktime.net/en/stable/api_reference/auto_generated/sktime.classification.shapelet_based.MrSEQL.html - last accessed 07/06/2024

[5] https://shap.readthedocs.io/ - last accessed 07/06/2024

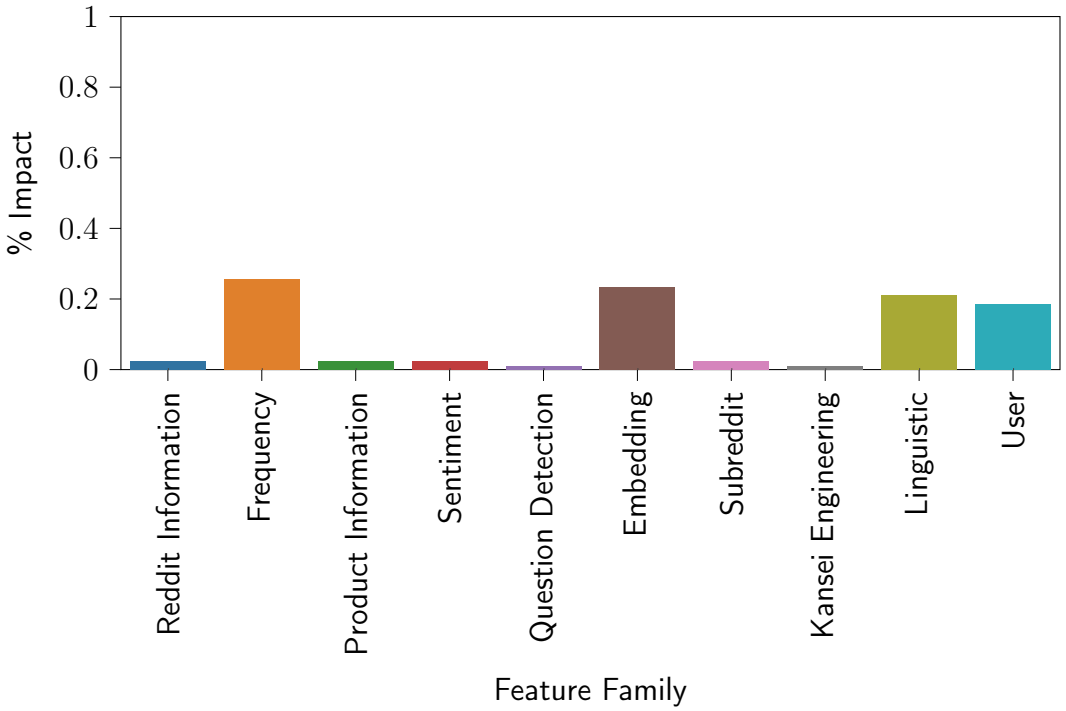

Figure 7.2. A hypothetical explainable analysis that could be used to reveal why a keyphrase (e.g. coconut) is predicted to be a future customer need using SHAP

7.2.3 Data Sources & Types

The only data source used to represent the VOC in this thesis is Reddit. Including new data sources that contain text such as social media platforms (e.g. Twitter, TikTok, Instagram, etc.) or blogs (e.g. Quora) would likely lead to increases in model performance. To extend the approach in Chapter 5 from an ML modelling perspective, two approaches could be used to allow for the addition of new data sources. The first approach could simply generate the same features for each new data source as in Chapter 5 and then build a new larger model with double (i.e. two data sources) or triple (i.e. three data sources) the number of features. Conversely, another approach could generate the same features, however, build separate models for each platform to form an ensemble (as previously done by other multi-platform ML approaches [381, 382]).

New data sources could also be used to enhance the ground truth datasets in Chapter 3. Currently, the ground truth includes the importance of a keyphrase each month by the frequency it is mentioned in a database of new-to-market products (i.e. Mintel GNPD). This has the effect of giving keyphrases more importance if they are addressed in many products e.g. the keyphrase "mint" for the Chocolate product category will be ranked highly if it is mentioned in many product descriptions in March 2016. However, the current state of the ground truth does not take into account the most important factor of whether the customer need leads to increased sales when deciding whether it should be included in a product. The problem with incorporating this knowledge into the ground truth is that sales data is very difficult to obtain for research purposes. Some studies have obtained this data from Nielsen [383, 384] - a marketing firm that offers a paid subscription to obtain sales data. However, this service can be difficult to obtain for research purposes. Sales data can also have the drawback of not containing an associated textual product description, therefore

making it difficult to track customer needs.

In addition to new data sources that could be used, new data types could also be explored. Since 2022, the usage of short-form video media such as TikTok and YouTube Shorts has had a 15-fold increase in four years [385]. Additionally, recent years have seen a rise in the number of users using image-based sites such as Instagram [386]. In parallel, there have been advancements in converting audio to text [387, 388] and images to text [389–391]. This research could be used to extract information on the vast amounts of audio and image data types to increase the performance of algorithms predicting future customer needs.

7.2.4 Large Language Models

Recently, Large Language Models (LLMs) have had a big impact in the field of AI and ML. Popular close-source (e.g. GPT-4 [392], Claude [393], Gemini [394] etc.) and open-source (e.g. Mixtral 8x7B [395]) models are increasingly being used by people and businesses to address various question-answering tasks. Although these models have been shown to be effective at summarizing social media content [396], they may not be the best at fully addressing the research problem in this thesis (i.e. predicting future customer needs) as they focus on returning a coherent response to a question based on previously trained conversations, articles etc. For example, if asked to predict the top future customer needs for Shampoo products, the named LLMs may return a response based on previous text of people discussing their predictions to this question (which may be based on various levels of research). In comparison, the approach in this thesis builds a model from high level features (e.g. frequency, sentiment, word embeddings etc.) that predicts future customer needs which occur in real market products (i.e. from Mintel GNPD) - which is a more specialized approach to solve the problem. That being said, these conversational LLMs can be used to improve the work in this thesis, specifically in two different ways: 1) increase the performance at predicting future customer needs; and 2) improve the efficiency of generating the ground truth data.

The most obvious way to increase the performance at predicting future customer needs by using LLMs would be to use them to generate additional binary features based off textual post data, which has been the approach used in other text classification tasks [397]. Various questions could be asked of the post a candidate keyphrase appears in such as "does this post contain customer needs (yes or no)" or "does this post detail benefits (yes or no)". Various questions could also be asked of the candidate keyphrase itself such as "can chicken be used in dog food products (yes or no)" i.e. if the candidate keyphrase was "chicken" and the product category was "dog food" in this example. In the context of this thesis, these additional features could be used in the approach in Chapter 5 along with the list of other generated features described in Section 5.1.4.

As discussed, LLMs could also be used when generating the ground truth data discussed in this thesis i.e. list of ranked keyphrase representing top customer needs in real market products. Specifically, it could be used when generating the TCN dataset (described in Section 3.4). Instead of having annotators label keyphrases as "customer needs" (as in Section 3.4.3), a LLM could instead label these keyphrases e.g. ask the model if "orange"

is a "customer need". This approach of labelling the ground truth label has been used in other NLP studies e.g. [398] annotates medical texts with the help of LLMs. Using LLMs this way would eliminate the time spent by annotators labelling keyphrases as it would instead be done automatically. This would come with some major drawbacks however, as the decision-making for deciding which keyphrases are customer needs would come from a model instead of human annotators (which could lead to major criticisms of the overall work in this thesis).

7.3 Final Observations & Overall Defense

As discussed in Chapter 2, predicting future customer needs is an important yet under-researched area in the landscape of automated solutions in the product development literature. This thesis therefore presents ground truth datasets (Chapter 3), algorithms (Chapters 4 and 5) and a user study (Chapter 6) which aims to address this problem. By researching the following topics, this thesis provides a primary step in the direction of additional research that may be carried out to predict future customer needs.

Research on predicting future customer needs could face similar criticisms to that of forecasting other real-world events retrospectively e.g. election results [312], stock market selection [399], product sales [400], etc. Chapters 4 and 5 would be susceptible to the same criticisms, as the evaluation of these algorithms is based on past data. However, the evaluation in Chapter 6 addresses this problem by comparing the algorithm output to predictions from real product developers in a multi-billion (USD) valued MNC. A misconception of the work in this thesis is that the overall results seem average or low, which would indicate that it is not useful. For example, some results presented in Chapters 4 and 5 may be considered average or low. For these results it is important to take into account the strict evaluation methodology and performance metrics applied. The overall usefulness this work is clearly demonstrated in Chapter 6 by showing that experts from a multi-billion dollar (USD) firm found the algorithm useful. This evidently shows the research value of this thesis.

Glossary

Candidate Keyphrase A candidate keyphrase is a phrase an algorithm analyzes in order to predict whether it is a keyphrase [69, 70, 72–74].. 4, 5, 7, 25, 52, 53, 55, 69, 70, 72, 73, 86–90, 95, 101, 122, 140, 147

Customer Need A customer need is a description in the customer's own words of the benefit to be fulfilled by a product or service [21]. For the purposes of this thesis, the definition includes the attributes/features of products which have benefits associated by their inclusion in the product. 2–16, 18–33, 35–38, 42–57, 59–63, 65, 67–72, 74, 75, 77, 78, 82–84, 86–89, 94–96, 117, 118, 120–123, 135, 140–142, 147, 148, 181, 182, 185–187, 189, 195, 207

Customer Needs Mining The customer needs mining literature is a group of studies which automatically extract customer needs from User Generated Content [18–20]. x, 2, 3, 5, 9–11, 17, 19, 21, 22, 26, 27, 33, 34, 37, 38, 42, 63, 94, 95, 140

Future Customer Need A future customer need has the same attributes as a regular customer need, however, it is required by consumers in the future. 4–9, 14, 16, 19, 21, 23–26, 29–32, 34, 35, 38, 42, 43, 47, 61, 63–74, 77, 79, 81–83, 85–87, 89, 90, 92–104, 106, 107, 110–114, 117–122, 124–130, 132–134, 136, 137, 139–145, 147, 148, 212, 213

GNPD GNPD is a database offered by Mintel which includes information on a product once it has been launched into the marketplace [79].. xii, 5, 8, 9, 14, 27, 32, 38, 42, 44–46, 48, 53, 55, 56, 67, 140, 142, 146, 147, 182–184

Keyphrase A keyphrase is a term which captures the main topics in a document or summarizes it best [69–72]. 4, 8

Subreddit A subreddit has been described as a "discussion forum" [401] or "community that is focused on a specific topic" [286] on the website Reddit. 3, 4, 16, 27, 63, 65

VOC The Voice of the Customer is a marketing term used to describe customer's requirements for products [4, 8–12].. 1, 2, 12, 13, 18, 38, 146

Acronyms

ABSA Aspect-Based Sentiment Analysis. 23

AHP Analytic Hierarchy Process. 11, 12

AI Artificial Intelligence. 1, 147

AP Affinity Propagation. 22

API Application Programming Interface. 2, 14–16, 18, 56, 63, 67, 90, 93, 128, 207, 208

ARIMA Autoregressive Integrated Moving Average. 23

AUC Area Under Curve. xi, 115–117, 120, 121

BIC Bayesian Information Criterion. 31

BoW Bag-of-Words. 2, 20, 37

CBOW Continuous Bag of Words. 22

CNN Convolutional Neural Network. 2, 21, 33

CPG Consumer Packaged Goods. 5, 6, 8, 34, 42, 44, 53, 58, 59, 67, 84, 85, 87, 121, 123, 126

DCG Discounted Cumulative Gain. 35

EPO European Patent Office. 13

FFE Fuzzy Front End. 10, 11

FMCG Fast Moving Consumer Goods. 126

GCP Google Cloud Platform. 56

IAA Inter Annotator Agreement. 49, 58–60, 143

IOB Inside Outside Beginning. 204

IR Information Retrieval. 23, 31

LDA Latent Dirichlet Allocation. 22, 28

LLM Large Language Model. 147, 148

LOB Lancaster-Oslo-Bergen. 73

LSTM Long Short-Term Memory. 2, 21

MAUs Monthly Active Users. 16, 18

MINIROCKET MINImally RandOm Convolutional KErnel Transform. 37, 96, 97, 99–102, 114, 122, 142

MK Mann-Kendall. 73, 74, 81

ML Machine Learning. 1–3, 5–11, 13, 20, 21, 24, 26, 29, 31–38, 43, 51, 53, 62, 65, 66, 70, 83–86, 89, 90, 95, 98, 100, 102–104, 113, 119–122, 140, 147

MNC Multinational Corporation. 5–9, 33, 34, 36, 38, 66, 68, 74, 78, 82, 83, 124–126, 132, 134, 139–143, 148

MTL Multi-task Learning. viii, xi, 6–9, 26, 38, 60, 85, 87, 95, 97–99, 102, 106–108, 110–113, 115–120, 122, 123, 140, 141, 143, 212, 213

MTSC Multivariate Time Series Classification. 5, 8–10, 26, 37, 38, 85, 86, 89, 95, 96, 116, 121, 122, 140, 142, 144, 145

NaN Not a Number. 90, 202

NDCG Normalized Discounted Cumulative Gain. 35

NER Named Entity Recognition. x, 5, 8, 23, 42, 47–53, 60, 67, 204

NER-T Named Entity Recognition based Toothpaste. x, 47, 48, 52–55, 60, 67–71, 142, 143

NLI Natural Language Inference. 198

NLP Natural Language Processing. 2, 14, 26, 37, 43, 51, 87, 148

NPD New Product Development. 126

NPMI Normalized Pointwise Mutual Information. 75

PCA Principal Component Analysis. 37, 199, 200

POS Part-of-Speech. 4, 5, 20, 50, 54, 73, 75, 77, 87–89, 95, 204

PR Precision-Recall. xi, 115, 116, 120, 121

PTSD Post-Traumatic Stress Disorder. 120, 121

QFD Quality Function Deployment. 11, 12

RC Research Contributions. 1, 7

ROC Receiver Operating Characteristic. xi, 115–117

ROCKET RandOm Convolutional KErnel Transform. 37

RQ Research Questions. 1, 6, 38, 74, 83, 98

SCAMPER Substitute-Combine-Adapt, Modify-Purpose-Eliminate-Reverse. 10

SHAP SHapley Additive exPlanations. 145

SME Small and Medium-sized Enterprise. 1, 139

SOM Self-Organizing Map. 3, 21

STEM Science, Technology, Engineering, and Mathematics. 181

SVM Support Vector Machine. 2, 21

TCN Trending Customer Needs. viii, x, xi, 53, 54, 59, 60, 67, 70, 85–89, 93, 95, 96, 100, 103, 107, 112–114, 116–119, 121–123, 128, 129, 142, 143, 147

TD Treatment Discontinuation. 121

TF-IDF Term Frequency - Inverse Document Frequency. 20

TRIZ Teorija Rezhenija Izobretatelskih Zadach. 11

UGC User Generated Content. 1–3, 11, 14, 19, 20, 23, 24, 28–30, 35, 36, 55, 56, 68, 83

USPTO United States Patent and Trademark Office. 13

XAI Explainable Artificial Intelligence. 122

Bibliography

1. Tagiuri, R. & Davis, J. A. On the goals of successful family companies. *Family Business Review* **5,** 43–62 (1992).
2. Feindt, S., Jeffcoate, J. & Chappell, C. Identifying success factors for rapid growth in SME e-commerce. *Small business economics* **19,** 51–62 (2002).
3. Freund, Y. P. Critical success factors. *Planning Review* **16,** 20–23 (1988).
4. Münch, J., Trieflinger, S. & Heisler, B. *Product Discovery–Building the Right Things: Insights from a Grey Literature Review* in *2020 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC)* (2020), 1–8.
5. Joh, J. M. & Mayfield, M. The discipline of product discovery: identifying breakthrough business opportunities. *Journal of Business Strategy* (2009).
6. Zirger, B. J. & Maidique, M. A. A model of new product development: An empirical test. *Management science* **36,** 867–883 (1990).
7. Takeuchi, H. & Nonaka, I. The new new product development game. *Harvard business review* **64,** 137–146 (1986).
8. Chang, W. & Taylor, S. A. The effectiveness of customer participation in new product development: A meta-analysis. *Journal of Marketing* **80,** 47–64 (2016).
9. Poetz, M. K. & Schreier, M. The value of crowdsourcing: can users really compete with professionals in generating new product ideas? *Journal of product innovation management* **29,** 245–256 (2012).
10. Melander, L. Customer involvement in product development: Using Voice of the Customer for innovation and marketing. *Benchmarking: An International Journal* (2019).
11. Kärkkäinen, H., Piippo, P. & Tuominen, M. Ten tools for customer-driven product development in industrial companies. *International journal of production economics* **69,** 161–176 (2001).
12. Cooper, R. G. The drivers of success in new-product development. *Industrial Marketing Management* **76,** 36–47 (2019).
13. Nishikawa, H., Schreier, M. & Ogawa, S. User-generated versus designer-generated products: A performance assessment at Muji. *International Journal of Research in Marketing* **30,** 160–167 (2013).
14. Kelly, A. in *The Art of Agile Product Ownership* 31–38 (Springer, 2019).
15. Liu, W., Ye, S. & Moultrie, J. Exploring traditional and new web-based methods to involve customers in new product development. *International Journal of Product Development* **23,** 42–64 (2019).
16. Ko, T., Rhiu, I., Yun, M. H. & Cho, S. A novel framework for identifying Customers' unmet needs on online social media using context tree. *Applied Sciences* **10,** 8473 (2020).

17. Gudigantala, N., Madhavaram, S. & Bicen, P. An AI decision-making framework for business value maximization. *AI Magazine* **44,** 67–84 (2023).

18. Jin, J., Jia, D. & Chen, K. Mining online reviews with a Kansei-integrated Kano model for innovative product design. *International Journal of Production Research* **60,** 6708–6727 (2022).

19. Jia, D. & Jin, J. *Mining Affective Needs from Online Opinions for Design Innovation* in *Web and Big Data: 4th International Joint Conference, APWeb-WAIM 2020, Tianjin, China, September 18-20, 2020, Proceedings, Part II 4* (2020), 317–324.

20. Liu, Q., Wang, K., Li, Y. & Liu, Y. Data-driven concept network for inspiring designers' idea generation. *Journal of Computing and Information Science in Engineering* **20,** 031004 (2020).

21. Gaskin, S. P., Griffin, A., Hauser, J. R., Katz, G. M. & Klein, R. L. V oice of the c ustomer. *Wiley International Encyclopedia of Marketing* (2010).

22. Kühl, N., Mühlthaler, M. & Goutier, M. Supporting customer-oriented marketing with artificial intelligence: automatically quantifying customer needs from social media. *Electronic Markets* **30,** 351–367 (2020).

23. Kühl, N. & Satzger, G. Needmining: Designing digital support to elicit needs from social media. *arXiv preprint arXiv:2101.06146* (2021).

24. Chen, D., Zhang, D. & Liu, A. Intelligent Kano classification of product features based on customer reviews. *CIRP annals* **68,** 149–152 (2019).

25. Chiu, M.-C. & Lin, K.-Z. Utilizing text mining and Kansei Engineering to support data-driven design automation at conceptual design stage. *Advanced Engineering Informatics* **38,** 826–839 (2018).

26. Kim, W., Ko, T., Rhiu, I. & Yun, M. H. Mining affective experience for a kansei design study on a recliner. *Applied ergonomics* **74,** 145–153 (2019).

27. Zhou, F. & Jiao, R. J. *Latent customer needs elicitation for big-data analysis of online product reviews* in *2015 IEEE international conference on industrial engineering and engineering management (IEEM)* (2015), 1850–1854.

28. Jiang, K. & Li, Y. *Mining customer requirement from online reviews based on multi-aspected sentiment analysis and Kano model* in *2020 16th Dahe Fortune China Forum and Chinese High-educational Management Annual Academic Conference (DFHMC)* (2020), 150–156.

29. Zha, Z.-J., Yu, J., Tang, J., Wang, M. & Chua, T.-S. Product aspect ranking and its applications. *IEEE transactions on knowledge and data engineering* **26,** 1211–1224 (2013).

30. Joung, J. & Kim, H. M. Automated keyword filtering in latent Dirichlet allocation for identifying product attributes from online reviews. *Journal of Mechanical Design* **143,** 084501 (2021).

31. Hananto, V. R., Kim, S., Kovacs, M., Serdült, U. & Kryssanov, V. *A machine learning approach to analyze fashion styles from large collections of online customer reviews* in *2021 6th International Conference on Business and Industrial Research (ICBIR)* (2021), 153–158.

32. Aman, J. J., Smith-Colin, J. & Zhang, W. Listen to E-scooter riders: Mining rider satisfaction factors from app store reviews. *Transportation research part D: transport and environment* **95,** 102856 (2021).

33. Chen, W.-K., Riantama, D. & Chen, L.-S. Using a text mining approach to hear voices of customers from social media toward the fast-food restaurant industry. *Sustainability* **13,** 268 (2020).

34. Kwon, H.-J., Ban, H.-J., Jun, J.-K. & Kim, H.-S. Topic modeling and sentiment analysis of online review for airlines. *Information* **12,** 78 (2021).

35. Tucker, C. & Kim, H. *Predicting emerging product design trend by mining publicly available customer review data* in *DS 68-6: Proceedings of the 18th International Conference on Engineering Design (ICED 11), Impacting Society through Engineering Design, Vol. 6: Design Information and Knowledge, Lyngby/Copenhagen, Denmark, 15.-19.08. 2011* (2011).

36. Jin, J., Ji, P. & Gu, R. Identifying comparative customer requirements from product online reviews for competitor analysis. *Engineering Applications of Artificial Intelligence* **49,** 61–73 (2016).

37. Joung, J. & Kim, H. Interpretable machine learning-based approach for customer segmentation for new product development from online product reviews. *International Journal of Information Management* **70,** 102641 (2023).

38. Kuehl, N., Scheurenbrand, J. & Satzger, G. Needmining: Identifying micro blog data containing customer needs. *arXiv preprint arXiv:2003.05917* (2020).

39. Giannakis, M., Dubey, R., Yan, S., Spanaki, K. & Papadopoulos, T. Social media and sensemaking patterns in new product development: demystifying the customer sentiment. *Annals of Operations Research* **308,** 145–175 (2022).

40. Tuarob, S. & Tucker, C. S. Quantifying product favorability and extracting notable product features using large scale social media data. *Journal of Computing and Information Science in Engineering* **15,** 031003 (2015).

41. Tuarob, S. & Tucker, C. S. *Automated discovery of product preferences in ubiquitous social media data: A case study of automobile market* in *2016 International Computer Science and Engineering Conference (ICSEC)* (2016), 1–6.

42. Hollerit, B., Kröll, M. & Strohmaier, M. *Towards linking buyers and sellers: detecting commercial intent on twitter* in *Proceedings of the 22nd international conference on world wide web* (2013), 629–632.

43. Wang, J., Cong, G., Zhao, X. & Li, X. *Mining user intents in twitter: A semi-supervised approach to inferring intent categories for tweets* in *Proceedings of the AAAI Conference on Artificial Intelligence* **29** (2015).

44. Kunja, S. R. & Gvrk, A. Examining the effect of eWOM on the customer purchase intention through value co-creation (VCC) in social networking sites (SNSs) A study of select Facebook fan pages of smartphone brands in India. *Management Research Review* **43,** 245–269 (2020).

45. Jeong, B., Yoon, J. & Lee, J.-M. Social media mining for product planning: A product opportunity mining approach based on topic modeling and sentiment analysis. *International Journal of Information Management* **48,** 280–290 (2019).

46. Ko, N., Jeong, B., Choi, S. & Yoon, J. Identifying product opportunities using social media mining: application of topic modeling and chance discovery theory. *IEEE access* **6,** 1680–1693 (2017).

47. Choi, J., Oh, S., Yoon, J., Lee, J.-M. & Coh, B.-Y. Identification of time-evolving product opportunities via social media mining. *Technological Forecasting and Social Change* **156,** 120045 (2020).

48. Cuomo, M. T., Tortora, D., Festa, G., Giordano, A. & Metallo, G. Exploring consumer insights in wine marketing: An ethnographic research on# Winelovers. *Psychology & Marketing* **33,** 1082–1090 (2016).

49. Gupta, V., Varshney, D., Jhamtani, H., Kedia, D. & Karwa, S. *Identifying purchase intent from social posts* in *Proceedings of the International AAAI Conference on Web and Social Media* **8** (2014), 180–186.

50. Alkahtani, M., Choudhary, A., De, A. & Harding, J. A. A decision support system based on ontology and data mining to improve design using warranty data. *Computers & industrial engineering* **128,** 1027–1039 (2019).

51. Buddhakulsomsiri, J., Siradeghyan, Y., Zakarian, A. & Li, X. Association rule-generation algorithm for mining automotive warranty data. *International journal of production research* **44,** 2749–2770 (2006).

52. Rajpathak, D. & De, S. A data-and ontology-driven text mining-based construction of reliability model to analyze and predict component failures. *Knowledge and Information Systems* **46,** 87–113 (2016).

53. Timoshenko, A. & Hauser, J. R. Identifying customer needs from user-generated content. *Marketing Science* **38,** 1–20 (2019).

54. Zhou, F., Ayoub, J., Xu, Q. & Jessie Yang, X. A machine learning approach to customer needs analysis for product ecosystems. *Journal of Mechanical Design* **142** (2020).

55. Zhang, M., Fan, B., Zhang, N., Wang, W. & Fan, W. Mining product innovation ideas from online reviews. *Information Processing & Management* **58,** 102389 (2021).

56. Joung, J., Jung, K., Ko, S. & Kim, K. Customer complaints analysis using text mining and outcome-driven innovation method for market-oriented product development. *Sustainability* **11,** 40 (2018).

57. İkiz, A. K. & Özdağoğlu, G. Text mining as a supporting process for VoC clarification. *Alphanumeric Journal* **3** (2015).

58. Kim, J., Park, S. & Kim, H. Analysis of customer sentiment on product features after the outbreak of coronavirus disease (COVID-19) based on online reviews. *Proceedings of the Design Society* **1,** 457–466 (2021).

59. Shi, Y. & Peng, Q. Definition of customer requirements in big data using word vectors and affinity propagation clustering. *Proceedings of the Institution of Mechanical Engineers, Part E: Journal of Process Mechanical Engineering* **235,** 1279–1291 (2021).

60. Palmer, S. *Crowdsourcing customer needs for product design using text analytics* in *Proceedings of the World Congress on Engineering* **1** (2016), 221–226.

61. Rai, R. *Identifying key product attributes and their importance levels from online customer reviews* in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* **45028** (2012), 533–540.

62. Mirtalaie, M. A., Hussain, O. K., Chang, E. & Hussain, F. K. A decision support framework for identifying novel ideas in new product development from cross-domain analysis. *Information Systems* **69,** 59–80 (2017).

63. Tuarob, S. & Tucker, C. S. *Fad or here to stay: Predicting product market adoption and longevity using large scale, social media data* in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* **55867** (2013), V02BT02A012.

64. Jiang, H., Kwong, C. K. & Yung, K. L. Predicting future importance of product features based on online customer reviews. *Journal of Mechanical Design* **139,** 111413 (2017).

65. Yakubu, H. & Kwong, C. Forecasting the importance of product attributes using online customer reviews and Google Trends. *Technological Forecasting and Social Change* **171,** 120983 (2021).

66. Ramanand, J., Bhavsar, K. & Pedanekar, N. *Wishful thinking-finding suggestions and'buy'wishes from product reviews* in *Proceedings of the NAACL HLT 2010 workshop on computational approaches to analysis and generation of emotion in text* (2010), 54–61.

67. Hartmann, J., Heitmann, M., Schamp, C. & Netzer, O. The power of brand selfies. *Journal of Marketing Research* **58,** 1159–1177 (2021).

68. Lee, T. Y. *Needs-based analysis of online customer reviews* in *Proceedings of the ninth international conference on Electronic commerce* (2007), 311–318.

69. Nguyen, T. D. & Kan, M.-Y. *Keyphrase extraction in scientific publications* in *Asian Digital Libraries. Looking Back 10 Years and Forging New Frontiers: 10th International Conference on Asian Digital Libraries, ICADL 2007, Hanoi, Vietnam, December 10-13, 2007. Proceedings 10* (2007), 317–326.

70. Liu, Z., Li, P., Zheng, Y. & Sun, M. *Clustering to find exemplar terms for keyphrase extraction* in *Proceedings of the 2009 conference on empirical methods in natural language processing* (2009), 257–266.

71. Wu, Y.-f. B., Li, Q., Bot, R. S. & Chen, X. *Domain-specific keyphrase extraction* in *Proceedings of the 14th ACM international conference on Information and knowledge management* (2005), 283–284.

72. Siddiqi, S. & Sharan, A. Keyword and keyphrase extraction techniques: a literature review. *International Journal of Computer Applications* **109** (2015).

73. Hasan, K. S. & Ng, V. *Automatic keyphrase extraction: A survey of the state of the art* in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (2014), 1262–1273.

74. Turney, P. D. Learning algorithms for keyphrase extraction. *Information retrieval* **2,** 303–336 (2000).

75. Sawhney, M., Wolcott, R. C. & Arroniz, I. The 12 different ways for companies to innovate. *MIT Sloan management review* (2006).

76. Kärkkäinen, H., Piippo, P., Puumalainen, K. & Tuominen, M. Assessment of hidden and future customer needs in Finnish business-to-business companies. *R&d Management* **31,** 391–407 (2001).

77. Hoonsopon, D. & Puriwat, W. Organizational agility: Key to the success of new product development. *IEEE Transactions on Engineering Management* **68,** 1722–1733 (2019).

78. Ruiz, A. P., Flynn, M., Large, J., Middlehurst, M. & Bagnall, A. The great multivariate time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery* **35,** 401–449 (2021).

79. Solis, E. Mintel global new products database (GNPD). *Journal of Business & Finance Librarianship* **21,** 79–82 (2016).

80. Zhang, Y. & Yang, Q. An overview of multi-task learning. *National Science Review* **5,** 30–43 (2018).

81. Kilroy, D., Healy, G. & Caton, S. Using machine learning to improve lead times in the identification of emerging customer needs. *IEEE Access* **10,** 37774–37795 (2022).

82. Kilroy, D., Healy, G. & Caton, S. Prediction of Future Customer Needs Using Machine Learning Across Multiple Product Categories (Under Review, Second Round). *PloS one* (2024).

83. Kilroy, D., Caton, S. & Healy, G. The Trending Customer Needs (TCN) Dataset: A Benchmarking and Automated Evaluation Approach for New Product Development (2023).

84. Kilroy, D., Caton, S. & Healy, G. Comparing ML-based and Human Product Development Approaches at Predicting Future Customer Needs (in Preparation) (2024).

85. Yelkur, R. & Herbig, P. Global markets and the new product development process. *Journal of Product & Brand Management* **5,** 38–47 (1996).

86. Kim, J. & Wilemon, D. Focusing the fuzzy front–end in new product development. *R&d Management* **32,** 269–279 (2002).

87. Koen, P. *et al.* Providing clarity and a common language to the "fuzzy front end". *Research-Technology Management* **44,** 46–55 (2001).

88. Rochford, L. Generating and screening new products ideas. *Industrial marketing management* **20,** 287–296 (1991).

89. Hamilton, H. R. Screening business development opportunities. *Business Horizons* **17,** 13–24 (1974).

90. Alam, I. Commercial innovations from consulting engineering firms: An empirical exploration of a novel source of new product ideas. *Journal of Product Innovation Management* **20,** 300–313 (2003).

91. Grissemann, U. S. & Stokburger-Sauer, N. E. Customer co-creation of travel services: The role of company support and customer satisfaction with the co-creation performance. *Tourism management* **33,** 1483–1492 (2012).

92. Piller, F. T., Ihl, C. & Vossen, A. A typology of customer co-creation in the innovation process. *Available at SSRN 1732127* (2010).

93. Merz, M. A., Zarantonello, L. & Grappi, S. How valuable are your customers in the brand value co-creation process? The development of a Customer Co-Creation Value (CCCV) scale. *Journal of Business Research* **82,** 79–89 (2018).

94. O'Hern, M. S. & Rindfleisch, A. Customer co-creation: a typology and research agenda. *Review of marketing research* **6,** 84–106 (2010).

95. VanGundy, A. B. Brain writing for new product ideas: an alternative to brainstorming. *Journal of Consumer Marketing* (1984).

96. Ozyaprak, M. The effectiveness of SCAMPER technique on creative thinking skills. *Journal for the Education of Gifted young scientists* **4,** 31–40 (2016).

97. Dahl, D. W. & Moreau, P. The influence and value of analogical thinking during new product ideation. *Journal of marketing research* **39,** 47–60 (2002).

98. Ilevbare, I. M., Probert, D. & Phaal, R. A review of TRIZ, and its benefits and challenges in practice. *Technovation* **33,** 30–37 (2013).

99. Hayes, B. E. *Measuring customer satisfaction and loyalty: survey design, use, and statistical analysis methods* (Quality Press, 2008).

100. Cooper, R. G. & Dreher, A. Voice-of-customer methods. *Marketing management* **19,** 38–43 (2010).

101. Liang, T., Bell, D. G. & Leifer, L. J. Re-Use or Re-Invent? Understanding and Supporting Learning from Experience of Peers in a Product Development Community. *Journal of Engineering Education* **90,** 519–526 (2001).

102. Griffin, A. & Hauser, J. R. The voice of the customer. *Marketing science* **12,** 1–27 (1993).

103. Timonen, H. & Järvenpää, E. *Knowledge acquisition models of Smes' new product development processes and the role of patent information* in *Conference Proceedings of e-business research forum* (2005), 25–26.

104. Magnusson, P. R., Netz, J. & Wästlund, E. Exploring holistic intuitive idea screening in the light of formal criteria. *Technovation* **34,** 315–326 (2014).

105. Rathore, A. K. & Ilavarasan, P. V. Pre-and post-launch emotions in new product development: Insights from twitter analytics of three products. *International Journal of Information Management* **50,** 111–127 (2020).

106. Kano, N. Attractive quality and must-be quality. *Hinshitsu (Quality, The Journal of Japanese Society for Quality Control)* **14,** 39–48 (1984).

107. Taifa, I. W. & Desai, D. A. Quality Function Deployment integration with Kano model for ergonomic product improvement (Classroom furniture)-a review. *Journal of multidisciplinary engineering science and technology (JMEST)* **2,** 2484–2491 (2015).

108. Basfirinci, C. & Mitra, A. A cross cultural investigation of airlines service quality through integration of Servqual and the Kano model. *Journal of Air Transport Management* **42,** 239–248 (2015).

109. Chaudha, A., Jain, R., Singh, A. & Mishra, P. Integration of Kano's Model into quality function deployment (QFD). *The International Journal of Advanced Manufacturing Technology* **53,** 689–698 (2011).

110. Velikova, N., Slevitch, L. & Mathe-Soulek, K. Application of Kano model to identification of wine festival satisfaction drivers. *International Journal of Contemporary Hospitality Management* **29,** 2708–2726 (2017).

111. Nagamachi, M. Kansei engineering: a new ergonomic consumer-oriented technology for product development. *International Journal of industrial ergonomics* **15,** 3–11 (1995).

112. López, Ó., Murillo, C. & González, A. Systematic literature reviews in kansei engineering for product design—A comparative study from 1995 to 2020. *Sensors* **21,** 6532 (2021).

113. Nagamachi, M. Kansei engineering and its applications in automotive design. *SAE transactions,* 2275–2282 (1999).

114. EL HILALI, N. & MATHIEU, J. P. *The power of concept in product design, smartphones nomad artifact: Semiotic analysis of one of them: Apple IPhone* in *International conference on Kansei engineering and emotion research KEER* (2012), 1.

115. Schütte, S. & Eklund, J. Design of rocker switches for work-vehicles—an application of Kansei Engineering. *Applied ergonomics* **36,** 557–567 (2005).

116. SAITO, E. Analysis of the desirable images for clothes in modern society. *Kansei Engineering International* **1,** 33–38 (2000).

117. Akao, Y. *Quality function deployment: integrating customer requirements into product design* (SteinerBooks, 2004).

118. *Quality function deployment* https://en.wikipedia.org/wiki/Quality_function_deployment. Accessed: 07/06/2024.

119. Saaty, T. L. *What is the analytic hierarchy process?* (Springer, 1988).

120. Vaidya, O. S. & Kumar, S. Analytic hierarchy process: An overview of applications. *European Journal of operational research* **169,** 1–29 (2006).

121. Jin, G., Jeong, Y. & Yoon, B. Technology-driven roadmaps for identifying new product/market opportunities: Use of text mining and quality function deployment. *Advanced Engineering Informatics* **29,** 126–138 (2015).

122. Roh, T., Jeong, Y., Jang, H. & Yoon, B. Technology opportunity discovery by structuring user needs based on natural language processing and machine learning. *PloS one* **14,** e0223404 (2019).

123. Wang, J. & Chen, Y.-J. A novelty detection patent mining approach for analyzing technological opportunities. *Advanced Engineering Informatics* **42,** 100941 (2019).

124. Russo, D., Spreafico, M. & Spreafico, C. Supporting decision making in design creativity through requirements identification and evaluation. *International Journal of Design Creativity and Innovation,* 1–17 (2023).

125. Lee, S., Lee, S., Seol, H. & Park, Y. Using patent information for designing new product and technology: keyword based technology roadmapping. *R&d Management* **38,** 169–188 (2008).

126. Livotov, P. Using patent information for identification of new product features with high market potential. *Procedia engineering* **131,** 1157–1164 (2015).

127. Wittfoth, S., Berger, T. & Moehrle, M. G. Revisiting the innovation dynamics theory: How effectiveness-and efficiency-oriented process innovations accompany product innovations. *Technovation* **112,** 102410 (2022).

128. Bzhalava, L., Kaivo-oja, J., Hassan, S. S. & Gerstlberger, W. D. Identifying entrepreneurial discovery processes with weak and strong technology signals: a text mining approach. *Open Research Europe* **2,** 26 (2022).

129. Qi, G. Building the Organizational Knowledge Networks of SMEs in High-tech Industry. *International Journal of Business and Management* **3,** 35–40 (2008).

130. Wu, S. Warranty data analysis: A review. *Quality and Reliability Engineering International* **28,** 795–805 (2012).

131. Yang, L. *et al. Mave: A product dataset for multi-source attribute value extraction* in *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining* (2022), 1256–1265.

132. Petrovski, P. & Bizer, C. *Extracting attribute-value pairs from product specifications on the web* in *Proceedings of the International Conference on Web Intelligence* (2017), 558–565.

133. Sabeh, K., Kacimi, M. & Gamper, J. *OpenBrand: Open Brand Value Extraction from Product Descriptions* in *Proceedings of The Fifth Workshop on e-Commerce and NLP (ECNLP 5)* (2022), 161–170.

134. Rezk, M., Alemany, L. A., Nio, L. & Zhang, T. *Accurate product attribute extraction on the field* in *2019 IEEE 35th International Conference on Data Engineering (ICDE)* (2019), 1862–1873.

135. Tuarob, S. & Tucker, C. S. Automated discovery of lead users and latent product features by mining large scale social media networks. *Journal of Mechanical Design* **137** (2015).

136. Choi, H. & Varian, H. Predicting the present with Google Trends. *Economic record* **88,** 2–9 (2012).

137. Wijnhoven, F. & Plant, O. *Sentiment Analysis and Google Trends Data for Predicting Car Sales* in *38th International Conference on Information Systems, ICIS 2017* (2017).

138. Hong, H., Xu, D., Wang, G. A. & Fan, W. Understanding the determinants of online review helpfulness: A meta-analytic investigation. *Decision Support Systems* **102,** 1–11 (2017).

139. Diaz, G. O. & Ng, V. *Modeling and prediction of online product review helpfulness: a survey* in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (2018), 698–708.

140. Boorugu, R. & Ramesh, G. *A survey on NLP based text summarization for summarizing product reviews* in *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)* (2020), 352–356.

141. Gupta, P., Tiwari, R. & Robert, N. *Sentiment analysis and text summarization of online reviews: A survey* in *2016 International Conference on Communication and Signal Processing (ICCSP)* (2016), 0241–0245.

142. Shivaprasad, T. & Shetty, J. *Sentiment analysis of product reviews: A review* in *2017 International conference on inventive communication and computational technologies (ICICCT)* (2017), 298–301.

143. Jebaseeli, A. N. & Kirubakaran, E. A survey on sentiment analysis of (product) reviews. *International Journal of Computer Applications* **47** (2012).

144. Davidson, B. I. *et al.* Social Media APIs: A Quiet Threat to the Advancement of Science (2023).

145. Kaplan, A. M. & Haenlein, M. Users of the world, unite! The challenges and opportunities of Social Media. *Business horizons* **53,** 59–68 (2010).

146. Himelboim, I., McCreery, S. & Smith, M. Birds of a feather tweet together: Integrating network and content analyses to examine cross-ideology exposure on Twitter. *Journal of computer-mediated communication* **18,** 154–174 (2013).

147. Morais, I. & Brito-Eliane, E. Productive consumption and marketplace dynamics: A study in the DIY homemade natural beauty products context. *ANPAD, São Paulo, Brazil, Tech. Rep* (2015).

148. Fei, G. *et al. Exploiting burstiness in reviews for review spammer detection* in *Proceedings of the international AAAI conference on web and social media* **7** (2013), 175–184.

149. Chevalier, J. A. & Mayzlin, D. The effect of word of mouth on sales: Online book reviews. *Journal of marketing research* **43,** 345–354 (2006).

150. Aichner, T., Grünfelder, M., Maurer, O. & Jegeni, D. Twenty-five years of social media: a review of social media applications and definitions from 1994 to 2019. *Cyberpsychology, behavior, and social networking* **24,** 215–222 (2021).

151. Popescu, A.-M., Kamath, K. Y. & Caverlee, J. *Mining Potential Domain Expertise in Pinterest.* in *UMAP workshops* (2013).

152. Chang, Y., Tang, L., Inagaki, Y. & Liu, Y. What is tumblr: A statistical overview and comparison. *ACM SIGKDD explorations newsletter* **16,** 21–29 (2014).

153. Dichev, K., Bukhsh, F. & Barrios-Fleitas, Y. *Application of NLP on student's Discord messages for automatic Belbin role identification* in *2022 International Conference on Frontiers of Information Technology (FIT)* (2022), 302–307.

154. Gunawan, T. S., Babiker, A. B. F., Ismail, N. & Effendi, M. R. *Development of Intelligent Telegram Chatbot Using Natural Language Processing* in *2021 7th International Conference on Wireless and Telematics (ICWT)* (2021), 1–5.

155. Wang, B., Jia, M. & Liu, Q. *Text analysis for TikTok comments on World Intangible Cultural Heritage: a case of Chaozhou woodcarving* in *4th International Conference on Information Science, Electrical, and Automation Engineering (ISEAE 2022)* **12257** (2022), 527–532.

156. Bao, H., Li, Q., Liao, S. S., Song, S. & Gao, H. A new temporal and social PMF-based method to predict users' interests in micro-blogging. *Decision Support Systems* **55,** 698–709 (2013).

157. LIU, L. *et al.* Research on the hot topics of health popular science based on 10 WeChat official accounts of traditional Chinese medicine hospitals in Beijing. *Chinese Journal of Hospital Administration,* 585–589 (2019).

158. Hu, G., Han, X., Zhou, H. & Liu, Y. Public perception on healthcare services: evidence from social media platforms in China. *International journal of environmental research and public health* **16,** 1273 (2019).

159. Freelon, D. Computational research in the post-API age. *Political Communication* **35,** 665–668 (2018).

160. Isaak, J. & Hanna, M. J. User data privacy: Facebook, Cambridge Analytica, and privacy protection. *Computer* **51,** 56–59 (2018).

161. Kupferschmidt, K. Twitter's threat to curtail free data access angers scientists. *Science (New York, NY)* **379,** 624–625 (2023).

162. Khan, M. *et al.* Predicting cryptocurrency value, based on sentimental analysis of social media post (2022).

163. Iqbal, M. TikTok revenue and usage statistics (2021). *Business of apps* **1** (2021).

164. Baumgartner, J., Zannettou, S., Keegan, B., Squire, M. & Blackburn, J. *The pushshift reddit dataset* in *Proceedings of the international AAAI conference on web and social media* **14** (2020), 830–839.

165. Goldberg, A. B. *et al. May all your wishes come true: A study of wishes and how to recognize them* in *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics* (2009), 263–271.

166. Dictionary, M.-W. Merriam-webster. *On-line at http://www. mw. com/home. htm* **8** (2002).

167. Gibson, K. BusinessDictionary. com. *Reference Reviews* **23,** 25–26 (2009).

168. Martineau, J., Finin, T., Joshi, A. & Patel, S. *Improving binary classification on text problems using differential word features* in *Proceedings of the 18th ACM conference on Information and knowledge management* (2009), 2019–2024.

169. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

170. Yang, Z. *et al.* Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems* **32** (2019).

171. Pennington, J., Socher, R. & Manning, C. D. *Glove: Global vectors for word representation* in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (2014), 1532–1543.

172. Aggarwal, C. C. & Zhai, C. A survey of text clustering algorithms. *Mining text data,* 77–128 (2012).

173. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. & Dean, J. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems* **26** (2013).

174. Okazaki, N. & Ohsawa, Y. *Polaris: an integrated data miner for chance discovery* in *Proceedings of The Third International Workshop on Chance Discovery and Its Management, Crete, Greece* (2003).

175. Sievert, C. & Shirley, K. *LDAvis: A method for visualizing and interpreting topics* in *Proceedings of the workshop on interactive language learning, visualization, and interfaces* (2014), 63–70.

176. Suryadi, D. & Kim, H. *Automatic identification of product usage contexts from online customer reviews* in *Proceedings of the Design Society: International Conference on Engineering Design* **1** (2019), 2507–2516.

177. Han, X., Li, R., Li, W., Ding, G. & Qin, S. *User requirements dynamic elicitation of complex products from social network service* in *2019 25th International Conference on Automation and Computing (ICAC)* (2019), 1–6.

178. Ayoub, J., Zhou, F., Xu, Q. & Yang, J. *Analyzing customer needs of product ecosystems using online product reviews* in *International design engineering technical conferences and computers and information in engineering conference* **59186** (2019), V02AT03A002.

179. Wang, W., Feng, Y. & Dai, W. Topic analysis of online reviews for two competitive products using latent Dirichlet allocation. *Electronic Commerce Research and Applications* **29,** 142–156 (2018).

180. Rousseeuw, P. J. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics* **20,** 53–65 (1987).

181. Aiello, L. M. *et al.* Sensing trending topics in Twitter. *IEEE Transactions on multimedia* **15,** 1268–1282 (2013).

182. Becker, H., Naaman, M. & Gravano, L. *Beyond trending topics: Real-world event identification on twitter* in *Proceedings of the international AAAI conference on web and social media* **5** (2011), 438–441.

183. Hu, R. *et al.* Technology topic identification and trend prediction of new energy vehicle using LDA modeling. *Complexity* **2022,** 1–20 (2022).

184. Mowlaei, M. E., Abadeh, M. S. & Keshavarz, H. Aspect-based sentiment analysis using adaptive aspect-based lexicons. *Expert Systems with Applications* **148,** 113234 (2020).

185. Pavlopoulos, I. Aspect based sentiment analysis. *Athens University of Economics and Business* (2014).

186. Ruder, S., Ghaffari, P. & Breslin, J. G. A hierarchical model of reviews for aspect-based sentiment analysis. *arXiv preprint arXiv:1609.02745* (2016).

187. Tran, T. K. & Phan, T. T. Mining opinion targets and opinion words from online reviews. *International Journal of Information Technology* **9,** 239–249 (2017).

188. Jin, J., Ji, P. & Liu, Y. *Product characteristic weighting for designer from online reviews: an ordinal classification approach* in *Proceedings of the 2012 Joint EDBT/ICDT Workshops* (2012), 33–40.

189. Yu, J., Zha, Z.-J., Wang, M. & Chua, T.-S. *Aspect ranking: identifying important product aspects from online consumer reviews* in *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies* (2011), 1496–1505.

190. Liu, C., Tang, L. & Shan, W. An extended hits algorithm on bipartite network for features extraction of online customer reviews. *Sustainability* **10,** 1425 (2018).

191. Alrababah, S. A. A. A., Gan, K. H. & Tan, T.-P. *Product aspect ranking using sentiment analysis and TOPSIS* in *2016 Third International Conference on Information Retrieval and Knowledge Management (CAMP)* (2016), 13–19.

192. Alrababah, S. A. A., Gan, K. H. & Tan, T.-P. *Comparative analysis of MCDM methods for product aspect ranking: TOPSIS and VIKOR* in *2017 8th International Conference on Information and Communication Systems (ICICS)* (2017), 76–81.

193. Yang, L., Liu, B., Lin, H. & Lin, Y. Combining local and global information for product feature extraction in opinion documents. *Information Processing Letters* **116,** 623–627 (2016).

194. Varol, O., Ferrara, E., Menczer, F. & Flammini, A. Early detection of promoted campaigns on social media. *EPJ data science* **6,** 1–19 (2017).

195. Ma, Z., Sun, A. & Cong, G. *Will this# hashtag be popular tomorrow?* in *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval* (2012), 1173–1174.

196. Ma, Z., Sun, A. & Cong, G. On predicting the popularity of newly emerging hashtags in t witter. *Journal of the American Society for Information Science and Technology* **64,** 1399–1410 (2013).

197. Nikolov, S. *Trend or no trend: a novel nonparametric method for classifying time series* PhD thesis (Massachusetts Institute of Technology, 2012).

198. Chen, G. H., Nikolov, S. & Shah, D. A latent source model for nonparametric time series classification. *Advances in neural information processing systems* **26** (2013).

199. Jeon, M., Jun, S. & Hwang, E. *Hashtag recommendation based on user tweet and hashtag classification on twitter* in *Web-Age Information Management: WAIM 2014 International Workshops: BigEM, HardBD, DaNoS, HRSUNE, BIDASYS, Macau, China, June 16-18, 2014, Revised Selected Papers 15* (2014), 325–336.

200. Tsur, O. & Rappoport, A. *What's in a hashtag? Content based prediction of the spread of ideas in microblogging communities* in *Proceedings of the fifth ACM international conference on Web search and data mining* (2012), 643–652.

201. Chen, S., Bortsova, G., Garcia-Uceda Juarez, A., Van Tulder, G. & De Bruijne, M. *Multi-task attention-based semi-supervised learning for medical image segmentation* in *Medical Image Computing and Computer Assisted Intervention–MICCAI 2019: 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part III 22* (2019), 457–465.

202. Zhang, Z., Yu, W., Yu, M., Guo, Z. & Jiang, M. A survey of multi-task learning in natural language processing: Regarding task relatedness and training methods. *arXiv preprint arXiv:2204.03508* (2022).

203. Chen, S., Zhang, Y. & Yang, Q. Multi-task learning in natural language processing: An overview. *arXiv preprint arXiv:2109.09138* (2021).

204. Mahmoud, R. A., Hajj, H. & Karameh, F. N. A systematic approach to multi-task learning from time-series data. *Applied Soft Computing* **96,** 106586 (2020).

205. Wei, C., Wang, Z., Yuan, J., Li, C. & Chen, S. Time-frequency based multi-task learning for semi-supervised time series classification. *Information Sciences* **619,** 762–780 (2023).

206. Khoshkangini, R., Mashhadi, P., Tegnered, D., Lundström, J. & Rögnvaldsson, T. Predicting Vehicle Behavior Using Multi-task Ensemble Learning. *Expert systems with applications* **212,** 118716 (2023).

207. Kuehl, N. *Needmining: Towards analytical support for service design* in *International Conference on Exploring Services Science* (2016), 187–200.

208. Ulwick, A. W. Turn customer input into innovation. *Harvard business review* **80,** 91–7 (2002).

209. Ulwick, A. W. What Is Outcome-Driven Innovation®(ODI)? *White Paper* (2009).

210. Killen, C. P., Walker, M. & Hunt, R. A. Strategic planning using QFD. *International Journal of Quality & Reliability Management* (2005).

211. Bi, J.-W., Liu, Y., Fan, Z.-P. & Cambria, E. Modelling customer satisfaction from online reviews using ensemble neural network and effect-based Kano model. *International Journal of Production Research* **57,** 7068–7088 (2019).

212. Zhao, M., Zhang, C., Hu, Y., Xu, Z. & Liu, H. Modelling consumer satisfaction based on online reviews using the improved Kano model from the perspective of risk attitude and aspiration. *Technological and Economic Development of Economy* **27,** 550–582 (2021).

213. Wang, W. M., Li, Z., Tian, Z., Wang, J. & Cheng, M. N. Extracting and summarizing affective features and responses from online product descriptions and reviews: A Kansei text mining approach. *Engineering Applications of Artificial Intelligence* **73,** 149–162 (2018).

214. Lin, S., Shen, T. & Guo, W. Evolution and emerging trends of Kansei engineering: A visual analysis based on citespace. *IEEE Access* **9,** 111181–111202 (2021).

215. Lai, X., Zhang, S., Mao, N., Liu, J. & Chen, Q. Kansei engineering for new energy vehicle exterior design: An internet big data mining approach. *Computers & Industrial Engineering* **165,** 107913 (2022).

216. Jin, J., Jia, D. & Chen, K. Mining online reviews with a Kansei-integrated Kano model for innovative product design. *International Journal of Production Research,* 1–20 (2021).

217. Davies, D. L. & Bouldin, D. W. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence,* 224–227 (1979).

218. Jelinek, F., Mercer, R. L., Bahl, L. R. & Baker, J. K. Perplexity—a measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America* **62,** S63–S63 (1977).

219. Röder, M., Both, A. & Hinneburg, A. *Exploring the space of topic coherence measures* in *Proceedings of the eighth ACM international conference on Web search and data mining* (2015), 399–408.

220. Schwarz, G. Estimating the dimension of a model. *The annals of statistics,* 461–464 (1978).

221. Privitera, M. B. & Murray, D. L. *Applied ergonomics: determining user needs in medical device design* in *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (2009), 5606–5608.

222. Fan, Z.-P., Che, Y.-J. & Chen, Z.-Y. Product sales forecasting using online reviews and historical sales data: A method combining the Bass model and sentiment analysis. *Journal of business research* **74,** 90–100 (2017).

223. Ching-Chin, C., Ieng, A. I. K., Ling-Ling, W. & Ling-Chieh, K. Designing a decision-support system for new product sales forecasting. *Expert Systems with applications* **37,** 1654–1665 (2010).

224. Beheshti-Kashi, S., Karimi, H. R., Thoben, K.-D., Lütjen, M. & Teucke, M. A survey on retail sales forecasting and prediction in fashion markets. *Systems Science & Control Engineering* **3,** 154–161 (2015).

225. Nayebi, M., Dicke, L., Ittyipe, R., Carlson, C. & Ruhe, G. ESSMArT way to manage customer requests. *Empirical Software Engineering* **24,** 3755–3789 (2019).

226. Tang, H. A Novel Framework Based on Word-of-mouth Mining for Non-prosumer Decision Support (2014).

227. Chen, L. & Pu, P. Experiments on the preference-based organization interface in recommender systems. *ACM Transactions on Computer-Human Interaction (TOCHI)* **17,** 1–33 (2010).

228. Gottschalk, S., Parvez, S., Yigitbas, E. & Engels, G. *Designing Platforms for Crowd-Based Software Prototype Validation: A Design Science Study* in *Product-Focused Software Process Improvement: 23rd International Conference, PROFES 2022, Jyväskylä, Finland, November 21–23, 2022, Proceedings* (2022), 334–350.

229. Danone, Y., Kuflik, T. & Mokryn, O. *Visualizing reviews summaries as a tool for restaurants recommendation* in *23rd International conference on intelligent user interfaces* (2018), 607–616.

230. Dash, A., Zhang, D. & Zhou, L. Personalized ranking of online reviews based on consumer preferences in product features. *International Journal of Electronic Commerce* **25,** 29–50 (2021).

231. Wang, C. *et al. Do Humans Prefer Debiased AI Algorithms? A Case Study in Career Recommendation* in *27th International Conference on Intelligent User Interfaces* (2022), 134–147.

232. Lee, B. C. G., Lo, K., Downey, D. & Weld, D. S. Explanation-based tuning of opaque machine learners with application to paper recommendation. *arXiv preprint arXiv:2003.04315* (2020).

233. Li, X., Zhang, Y., Leung, J., Sun, C. & Zhao, J. EDAssistant: Supporting Exploratory Data Analysis in Computational Notebooks with In Situ Code Search and Recommendation. *ACM Transactions on Interactive Intelligent Systems* **13,** 1–27 (2023).

234. Kim, J. *et al.* No more one liners: bringing context into emoji recommendations. *ACM Transactions on Social Computing* **3,** 1–25 (2020).

235. Oh, I. S., Lee, J. E. & Kim, K. J. Proactive News Article Summarization Service Using Personal Intention Models. *Evolutionary and Institutional Economics Review* **11,** 105–120 (2014).

236. Hughes, J., Aycock, S., Caines, A., Buttery, P. & Hutchings, A. *Detecting trending terms in cybersecurity forum discussions* in (2020).

237. Gropp, M., Nöth, E. & Riedhammer, K. *A Novel lecture browsing system using ranked key phrases and streamgraphs* in *Text, Speech and Dialogue: 14th International Conference, TSD 2011, Pilsen, Czech Republic, September 1-5, 2011. Proceedings 14* (2011), 17–24.

238. Järvelin, K. & Kekäläinen, J. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems* **20,** 422–446 (2002).

239. Asur, S. & Huberman, B. A. *Predicting the future with social media* in *2010 IEEE/WIC/ACM international conference on web intelligence and intelligent agent technology* **1** (2010), 492–499.

240. Wang, X., Yucesoy, B., Varol, O., Eliassi-Rad, T. & Barabási, A.-L. Success in books: predicting book sales before publication. *EPJ Data Science* **8,** 1–20 (2019).

241. Nezhadbiglari, M., Gonçalves, M. A. & Almeida, J. M. *Early prediction of scholar popularity* in *Proceedings of the 16th ACM/IEEE-CS on Joint Conference on Digital Libraries* (2016), 181–190.

242. Bagnall, A. *et al.* The UEA multivariate time series classification archive, 2018. *arXiv preprint arXiv:1811.00075* (2018).

243. Löning, M. *et al.* sktime: A unified interface for machine learning with time series. *arXiv preprint arXiv:1909.07872* (2019).

244. Hsieh, R.-J., Chou, J. & Ho, C.-H. *Unsupervised online anomaly detection on multivariate sensing time series data for smart manufacturing* in *2019 IEEE 12th Conference on Service-Oriented Computing and Applications (SOCA)* (2019), 90–97.

245. Xiahou, X. & Harada, Y. B2C E-Commerce Customer Churn Prediction Based on K-Means and SVM. *Journal of Theoretical and Applied Electronic Commerce Research* **17,** 458–475 (2022).

246. Dempster, A., Schmidt, D. F. & Webb, G. I. *Minirocket: A very fast (almost) deterministic transform for time series classification* in *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining* (2021), 248–257.

247. Dempster, A., Petitjean, F. & Webb, G. I. ROCKET: exceptionally fast and accurate time series classification using random convolutional kernels. *Data Mining and Knowledge Discovery* **34,** 1454–1495 (2020).

248. Wittenstam, M. & Andersson Neale, N. *Innovation domains Dimensions of controlled & uncontrolled and internal & external* MA thesis (2017).

249. Vlachos, I. & Dyra, S. C. Theorizing coordination, collaboration and integration in multi-sourcing triads (B3B triads). *Supply Chain Management: An International Journal* **25,** 285–300 (2020).

250. Abhari, K. *Modeling actor behavior in collaborative innovation networks: The case of social product-development* PhD thesis (University of Hawai'i at Manoa, 2014).

251. Beretta, M., Björk, J. & Magnusson, M. Moderating ideation in web-enabled ideation systems. *Journal of Product Innovation Management* **35,** 389–409 (2018).

252. Ogawa, S. & Piller, F. T. Reducing the risks of new product development. *MIT Sloan management review* (2006).

253. Mirkovski, K., Von Briel, F. & Lowry, P. B. Semantic learning-based innovation framework for social media. *It Professional* **18,** 26–32 (2016).

254. Salord, T. *et al.* Packaged foods with pulse ingredients in Europe: A dataset of text-mined product formulations. *Data in Brief* **42,** 108173 (2022).

255. Wu, C.-S., Kuo, C.-J., Su, C.-H., Wang, S.-H. & Dai, H.-J. Using text mining to extract depressive symptoms and to validate the diagnosis of major depressive disorder from electronic health records. *Journal of affective disorders* **260,** 617–623 (2020).

256. Cohen, J. A coefficient of agreement for nominal scales. *Educational and psychological measurement* **20,** 37–46 (1960).

257. Hripcsak, G. & Rothschild, A. S. Agreement, the f-measure, and reliability in information retrieval. *Journal of the American medical informatics association* **12,** 296–298 (2005).

258. Grouin, C. *et al. Proposal for an Extension of Traditional Named Entitites: from Guidelines to Evaluation, an Overview* in *5th Linguistics Annotation Workshop (The LAW V)* (2011), 92–100.

259. Deleger, L., Li, Q., Lingren, T., Kaiser, M., Molnar, K., *et al. Building gold standard corpora for medical natural language processing tasks* in *AMIA Annual Symposium Proceedings* **2012** (2012), 144.

260. Brandsen, A. *et al. Creating a dataset for named entity recognition in the archaeology domain* in *Conference Proceedings LREC 2020* (2020), 4573–4577.

261. Landis, J. R. & Koch, G. G. The measurement of observer agreement for categorical data. *biometrics,* 159–174 (1977).

262. Wei, J. & Zou, K. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196* (2019).

263. Sharifirad, S., Jafarpour, B. & Matwin, S. *Boosting text classification performance on sexist tweets by text augmentation and text generation using a combination of knowledge graphs* in *Proceedings of the 2nd workshop on abusive language online (ALW2)* (2018), 107–114.

264. Dai, X. & Adel, H. *An Analysis of Simple Data Augmentation for Named Entity Recognition* in *Proceedings of the 28th International Conference on Computational Linguistics* (2020), 3861–3867.

265. Dietterich, T. G. *et al.* Ensemble learning. *The handbook of brain theory and neural networks* **2,** 110–125 (2002).

266. Yang, X., Zhang, H., He, X., Bian, J. & Wu, Y. Extracting Family History of Patients From Clinical Narratives: Exploring an End-to-End Solution With Deep Learning Models. *JMIR Medical Informatics* **8,** e22982 (2020).

267. Copara, J., Naderi, N., Knafou, J., Ruch, P. & Teodoro, D. Named entity recognition in chemical patents using ensemble of contextual language models. *arXiv preprint arXiv:2007.12569* (2020).

268. Tafti, A. P. *et al. Artificial intelligence to organize patient portal messages: a journey from an ensemble deep learning text classification to rule-based named entity recognition* in *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* (2019), 1380–1387.

269. Segura Bedmar, I., Martinez, P. & Herrero Zazo, M. *Semeval-2013 task 9: Extraction of drug-drug interactions from biomedical texts (ddiextraction 2013)* in (2013).

270. Romo-Fernández, L. M., Guerrero-Bote, V. P. & Moya-Anegón, F. Co-word based thematic analysis of renewable energy (1990–2010). *Scientometrics* **97,** 743–765 (2013).

271. Trinquart, L. & Galea, S. Mapping epidemiology's past to inform its future: meta-knowledge analysis of epidemiologic topics in leading journals, 1974–2013. *American journal of epidemiology* **182,** 93–104 (2015).

272. Kleinberg, J. Bursty and hierarchical structure in streams. *Data mining and knowledge discovery* **7,** 373–397 (2003).

273. Uchitpe, M., Uddin, S. & Lynn, C. Predicting the future of project management research. *Procedia-Social and Behavioral Sciences* **226,** 27–34 (2016).

274. Krishnamoorthy, S. Linguistic features for review helpfulness prediction. *Expert Systems with Applications* **42,** 3751–3759 (2015).

275. QasemiZadeh, B. & Handschuh, S. *The ACL RD-TEC: a dataset for benchmarking terminology extraction and classification in computational linguistics* in *Proceedings of the 4th International Workshop on Computational Terminology* (2014), 52–63.

276. Khare, R. *et al.* Scaling drug indication curation through crowdsourcing. *Database* **2015** (2015).

277. Ni, J., Li, J. & McAuley, J. *Justifying recommendations using distantly-labeled reviews and fine-grained aspects* in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing* (2019), 188–197.

278. Oleson, D., Sorokin, A., *et al. Programmatic gold: Targeted and scalable quality assurance in crowdsourcing* in *Workshops at the Twenty-Fifth AAAI conference on artificial intelligence* (2011).

279. Comito, C., Forestiero, A. & Pizzuti, C. Bursty event detection in Twitter streams. *ACM Transactions on Knowledge Discovery from Data (TKDD)* **13,** 1–28 (2019).

280. Brownlee, J. *Imbalanced classification with Python: better metrics, balance skewed classes, cost-sensitive learning* (Machine Learning Mastery, 2020).

281. Aoyama, H. A study of stratified random sampling. *Ann. Inst. Stat. Math* **6,** 1–36 (1954).

282. Iliyasu, R. & Etikan, I. Comparison of quota sampling and stratified random sampling. *Biom. Biostat. Int. J. Rev* **10,** 24–27 (2021).

283. Brooks, J. K., Bashirelahi, N. & Reynolds, M. A. Charcoal and charcoal-based dentifrices: a literature review. *The Journal of the American Dental Association* **148,** 661–670 (2017).

284. Hosadurga, R., Boloor, V. A., Rao, S. N. & MeghRani, N. Effectiveness of two different herbal toothpaste formulations in the reduction of plaque and gingival inflammation in patients with established gingivitis–A randomized controlled trial. *Journal of traditional and complementary medicine* **8,** 113–119 (2018).

285. Lahitani, A. R., Permanasari, A. E. & Setiawan, N. A. *Cosine similarity to determine similarity measure: Study case in online essay assessment* in *2016 4th International Conference on Cyber and IT Service Management* (2016), 1–6.

286. Anderson, K. E. Ask me anything: what is Reddit? *Library Hi Tech News* **32,** 8–11 (2015).

287. Gorwa, R. & Guilbeault, D. Unpacking the social media bot: A typology to guide research and policy. *Policy & Internet* **12,** 225–248 (2020).

288. Cresci, S., Lillo, F., Regoli, D., Tardelli, S. & Tesconi, M. *$ FAKE: Evidence of spam and bot activity in stock microblogs on Twitter* in *Proceedings of the International AAAI Conference on Web and Social Media* **12** (2018).

289. Merrouni, Z. A., Frikh, B. & Ouhbi, B. Automatic keyphrase extraction: a survey and trends. *Journal of Intelligent Information Systems,* 1–34 (2019).

290. Singh, A. S. & Tucker, C. S. *Investigating the heterogeneity of product feature preferences mined using online product data streams* in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* **57083** (2015), V02BT03A020.

291. Mubarok, M. S., Adiwijaya & Aldhi, M. D. *Aspect-based sentiment analysis to review products using Naive Bayes* in *AIP Conference Proceedings* **1867** (2017), 020060.

292. Chen, X. *et al.* A novel feature extraction methodology for sentiment analysis of product reviews. *Neural Computing and Applications* **31,** 6625–6642 (2019).

293. Pearson, K. X. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* **50,** 157–175 (1900).

294. Francis, W. N. & Kucera, H. Brown corpus manual. *Letters to the Editor* **5,** 7 (1979).

295. Atwell, E., Leech, G. & Garside, R. *Analysis of the LOB Corpus: progress and prospects* in *Corpus Linguistics* (1984), 41–52.

296. Paquot, M. & Bestgen, Y. in *Corpora: Pragmatics and discourse* 247–269 (Brill Rodopi, 2009).

297. Palomino, M. A. & Wuytack, T. *Unsupervised extraction of keywords from news archives* in *Language and Technology Conference* (2009), 544–555.

298. Rayson, P. Corpus analysis of key words. *The encyclopedia of applied linguistics* (2012).

299. Ahmad, I., Tang, D., Wang, T., Wang, M. & Wagan, B. Precipitation trends over time using Mann-Kendall and spearman's rho tests in swat river basin, Pakistan. *Advances in Meteorology* **2015** (2015).

300. Sharma, D., Kumar, B. & Chand, S. A Trend Analysis of Machine Learning Research with Topic Models and Mann-Kendall Test. *International Journal of Intelligent Systems and Applications* **11,** 70–82 (2019).

301. Malakar, S., Goswami, S. & Chakrabarti, A. in *Industry Interactive Innovations in Science, Engineering and Technology* 185–193 (Springer, 2018).

302. Hurtado, S., Ray, P. & Marculescu, R. *Bot detection in reddit political discussion* in *Proceedings of the Fourth International Workshop on Social Sensing* (2019), 30–35.

303. Raju, S., Pingali, P. & Varma, V. *An unsupervised approach to product attribute extraction* in *European Conference on Information Retrieval* (2009), 796–800.

304. Raju, S., Shishtla, P. & Varma, V. *A Graph Clustering Approach to Product Attribute Extraction.* in *IICAI* (2009), 1438–1447.

305. Jakob, N. & Gurevych, I. *LRTwiki: Enriching the likelihood ratio test with encyclopedic information for the extraction of relevant terms* in *in Proceedings of the WikiAI 09-IJCAI Workshop: User Contributed Knowledge and Artificial Intelligence: An Evolving Synergy* (2009).

306. Hamilton, L. M. & Lahne, J. Fast and automated sensory analysis: Using natural language processing for descriptive lexicon development. *Food Quality and Preference* **83,** 103926 (2020).

307. Pandis, N. The chi-square test. *American journal of orthodontics and dentofacial orthopedics* **150,** 898–899 (2016).

308. Mann, H. B. & Whitney, D. R. On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics,* 50–60 (1947).

309. Olad, A. A. & Valilai, O. F. *Using of Social Media Data Analytics for Applying Digital Twins in Product Development* in *2020 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)* (2020), 319–323.

310. Chen, W.-K., Riantama, D. & Chen, L.-S. Using a text mining approach to hear voices of customers from social media toward the fast-food restaurant industry. *Sustainability* **13,** 268 (2021).

311. Koss, J. & Bohnet-Joschko, S. Social Media Mining in Drug Development Decision Making: Prioritizing Multiple Sclerosis Patients' Unmet Medical Needs (2022).

312. Gayo-Avello, D. No, you cannot predict elections with Twitter. *IEEE Internet Computing* **16,** 91–94 (2012).

313. Honnibal, M., Montani, I., Van Landeghem, S. & Boyd, A. spaCy: Industrial-strength natural language processing in python. *Zenodo, Honolulu, HI, USA* (2020).

314. Weischedel, R. *et al.* Ontonotes release 5.0 ldc2013t19. *Linguistic Data Consortium, Philadelphia, PA* **23** (2013).

315. Read, J., Dridan, R., Oepen, S. & Solberg, L. J. *Sentence boundary detection: A long solved problem?* in *Proceedings of COLING 2012: Posters* (2012), 985–994.

316. Zesch, T. & Gurevych, I. *Approximate matching for evaluating keyphrase extraction* in *Proceedings of the International Conference RANLP-2009* (2009), 484–489.

317. Berend, G. Exploiting extra-textual and linguistic information in keyphrase extraction. *Natural Language Engineering* **22,** 73–95 (2016).

318. Gopan, E., Rajesh, S., Vishnu, G., Thushara, M., *et al. Comparative study on different approaches in keyword extraction* in *2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC)* (2020), 70–74.

319. Li, X. & Song, F. *Keyphrase extraction and grouping based on association rules* in *The Twenty-Eighth International Flairs Conference* (2015).

320. Papagiannopoulou, E. & Tsoumakas, G. Local word vectors guiding keyphrase extraction. *Information Processing & Management* **54,** 888–902 (2018).

321. Ahel, R., Dalbelo Bašic, B. & Šnajder, J. Automatic keyphrase extraction from Croatian newspaper articles. *The Future of Information Sciences, Digital Resources and Knowledge Sharing,* 207–218 (2009).

322. Simon, H. & Leker, J. Using startup communication for opportunity recognition—an approach to identify future product trends. *International Journal of Innovation Management* **20,** 1640016 (2016).

323. Demszky, D. *et al. GoEmotions: A Dataset of Fine-Grained Emotions* in *58th Annual Meeting of the Association for Computational Linguistics (ACL)* (2020).

324. Reimers, N. & Gurevych, I. *Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks* in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing* (Association for Computational Linguistics, Nov. 2019). https://arxiv.org/abs/1908.10084.

325. Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C. & Joulin, A. *Advances in Pre-Training Distributed Word Representations* in *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)* (2018).

326. Stahlmann, S., Ettrich, O., Kurka, M. & Schoder, D. *What Do Customers Say About My Products? Benchmarking Machine Learning Models for Need Identification* in *Proc. of the HICSS* (2023).

327. Kuang, Z., Li, Z., Zhao, T. & Fan, J. *Deep multi-task learning for large-scale image classification* in *2017 IEEE Third International Conference on Multimedia Big Data (BigMM)* (2017), 310–317.

328. Kaur, H., Pannu, H. S. & Malhi, A. K. A systematic review on imbalanced data challenges in machine learning: Applications and solutions. *ACM Computing Surveys (CSUR)* **52,** 1–36 (2019).

329. Hancock, J., Johnson, J. M. & Khoshgoftaar, T. M. *A Comparative Approach to Threshold Optimization for Classifying Imbalanced Data* in *2022 IEEE 8th International Conference on Collaboration and Internet Computing (CIC)* (2022), 135–142.

330. Kilroy, D., Caton, S. & Healy, G. *Finding Short Lived Events on Social Media.* in *AICS* (2020), 49–60.

331. Hart, A. Mann-Whitney test is not just a test of medians: differences in spread can be important. *Bmj* **323,** 391–393 (2001).

332. Cowles, M. & Davis, C. On the origins of the. 05 level of statistical significance. *American Psychologist* **37,** 553 (1982).

333. Boyd, K., Eng, K. H. & Page, C. D. *Area under the precision-recall curve: point estimates and confidence intervals* in *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2013, Prague, Czech Republic, September 23-27, 2013, Proceedings, Part III 13* (2013), 451–466.

334. Davis, J. & Goadrich, M. *The relationship between Precision-Recall and ROC curves* in *Proceedings of the 23rd international conference on Machine learning* (2006), 233–240.

335. Saito, T. & Rehmsmeier, M. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PloS one* **10,** e0118432 (2015).

336. Alami Merrouni, Z., Frikh, B. & Ouhbi, B. Automatic keyphrase extraction: a survey and trends. *Journal of Intelligent Information Systems* **54,** 391–424 (2020).

337. Merrouni, Z. A., Frikh, B. & Ouhbi, B. *Automatic keyphrase extraction: An overview of the state of the art* in *2016 4th IEEE international colloquium on information science and technology (CiSt)* (2016), 306–313.

338. Bora, S., Singh, H., Sen, A., Bagchi, A. & Singla, P. On the role of conductance, geography and topology in predicting hashtag virality. *Social Network Analysis and Mining* **5,** 1–15 (2015).

339. Yilmaz, I., Masum, R. & Siraj, A. *Addressing imbalanced data problem with generative adversarial network for intrusion detection* in *2020 IEEE 21st International Conference on Information Reuse and Integration for Data Science (IRI)* (2020), 25–30.

340. Vieira Bernat, M. *Topical Classification of Images in Wikipedia: Development of topical classification models followed by a study of the visual content of Wikipedia* 2023.

341. Held, P. *et al.* Who will respond to intensive PTSD treatment? A machine learning approach to predicting response prior to starting treatment. *Journal of psychiatric research* **151,** 78–85 (2022).

342. Kurasawa, H. *et al.* Treatment Discontinuation Prediction in Patients With Diabetes Using a Ranking Model: Machine Learning Model Development. *JMIR Bioinformatics and Biotechnology* **3,** e37951 (2022).

343. Lu, H., Ehwerhemuepha, L. & Rakovski, C. A comparative study on deep learning models for text classification of unstructured medical notes with various levels of class imbalance. *BMC Medical Research Methodology* **22,** 181 (2022).

344. Judson, K., Schoenbachler, D. D., Gordon, G. L., Ridnour, R. E. & Weilbaker, D. C. The new product development process: let the voice of the salesperson be heard. *Journal of Product & Brand Management* **15,** 194–202 (2006).

345. Frishammar, J. Managing information in new product development: A literature review. *International Journal of Innovation and Technology Management* **2,** 259–275 (2005).

346. Chong, Y. T. & Chen, C.-H. Customer needs as moving targets of product development: a review. *The International Journal of Advanced Manufacturing Technology* **48,** 395–406 (2010).

347. Klein, A., Falkner, S., Bartels, S., Hennig, P. & Hutter, F. *Fast bayesian optimization of machine learning hyperparameters on large datasets* in *Artificial intelligence and statistics* (2017), 528–536.

348. Kathirgamanathan, B. & Cunningham, P. Correlation based feature subset selection for multivariate time-series data. *arXiv preprint arXiv:2112.03705* (2021).

349. Sun, Y. *et al.* Using causal discovery for feature selection in multivariate numerical time series. *Machine Learning* **101,** 377–395 (2015).

350. Pistorius, F., Baumann, D. & Sax, E. *Differential Correlation Approach for Multivariate Time Series Feature Selection* in *Proceedings of the Future Technologies Conference (FTC) 2021, Volume 1* (2022), 928–942.

351. Kathirgamanathan, B. & Cunningham, P. *A feature selection method for multi-dimension time-series data* in *Advanced Analytics and Learning on Temporal Data: 5th ECML PKDD Workshop, AALTD 2020, Ghent, Belgium, September 18, 2020, Revised Selected Papers 6* (2020), 220–231.

352. Younus, A., Qureshi, M. A., Jeon, M., Kazemi, A. & Caton, S. *XAI Analysis of Online Activism to Capture Integration in Irish Society Through Twitter* in *International Conference on Social Informatics* (2022), 233–244.

353. Le Nguyen, T., Gsponer, S., Ilie, I., O'reilly, M. & Ifrim, G. Interpretable time series classification using linear models and multi-resolution multi-domain symbolic representations. *Data mining and knowledge discovery* **33,** 1183–1222 (2019).

354. Fauvel, K., Lin, T., Masson, V., Fromont, É. & Termier, A. Xcm: An explainable convolutional neural network for multivariate time series classification. *Mathematics* **9,** 3137 (2021).

355. Assaf, R., Giurgiu, I., Bagehorn, F. & Schumann, A. *Mtex-cnn: Multivariate time series explanations for predictions with convolutional neural networks* in *2019 IEEE International Conference on Data Mining (ICDM)* (2019), 952–957.

356. Ozyegen, O., Ilic, I. & Cevik, M. Evaluation of interpretability methods for multivariate time series forecasting. *Applied Intelligence,* 1–17 (2022).

357. Abdolmohammadi, M. J. Intellectual capital disclosure and market capitalization. *Journal of intellectual capital* **6,** 397–416 (2005).

358. Davis, D., Chelliah, J. & Minter, S. New product development processes in the Australian FMCG industry. *Contemporary Management Research* **10** (2014).

359. Cometto, T., Nisar, A., Palacios, M., Le Meunier-FitzHugh, K. & Labadie, G. J. Organizational linkages for new product development: Implementation of innovation projects. *Journal of Business Research* **69,** 2093–2100 (2016).

360. Karp, J. M. *Perceptions of craft beer brands as determined by female consumers in Ireland* PhD thesis (Dublin Business School, 2018).

361. Sharenkova, A. *et al.* Integration of marketing research data in new product development. Case study: Food industry company (2015).

362. Amon, M. *Digital representation of an innovation cycle in the FMCG industry* PhD thesis (Wien, 2020).

363. Davidson, B. I. *et al.* Platform-controlled social media APIs threaten Open Science. *Nature Human Behaviour,* 1–4 (2023).

364. Reid, B., Wagner, M., d'Amorim, M. & Treude, C. Software engineering user study recruitment on prolific: An experience report. *arXiv preprint arXiv:2201.05348* (2022).

365. Du, Y., Antoniadi, A. M., McNestry, C., McAuliffe, F. M. & Mooney, C. The Role of XAI in Advice-Taking from a Clinical Decision Support System: A Comparative User Study of Feature Contribution-Based and Example-Based Explanations. *Applied Sciences* **12,** 10323 (2022).

366. Abdulrazzaq, A. H. Encouraging to Improving the use and search techniques skills on the webbased e-scholarly Databases subscribed by UOBL. *International Research: Journal of Library and Information Science* **4** (2014).

367. Sullivan Jr, J. *et al. Explaining why: How instructions and user interfaces impact annotator rationales when labeling text data* in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (2022), 521–531.

368. Belkacem, S. *Machine learning approaches to rank news feed updates on social media* PhD thesis (Université des Sciences et de la Technologie Houari Boumediene Alger, 2021).

369. Vermeer, S. A., Araujo, T., Bernritter, S. F. & van Noort, G. Seeing the wood for the trees: How machine learning can help firms in identifying relevant electronic word-of-mouth in social media. *International Journal of Research in Marketing* **36,** 492–508 (2019).

370. Sindhu, S., Nair, D. S., Maya, V., Thanseeha, M. & Hari, C. V. *Disaster management from social media using machine learning* in *2019 9th International Conference on Advances in Computing and Communication (ICACC)* (2019), 246–252.

371. Chau, D. H., Kittur, A., Hong, J. I. & Faloutsos, C. *Apolo: making sense of large network data by combining rich user interaction and machine learning* in *Proceedings of the SIGCHI conference on human factors in computing systems* (2011), 167–176.

372. Rauschenberger, M., Baeza-Yates, R. & Rello, L. *Screening risk of dyslexia through a web-game using language-independent content and machine learning* in *Proceedings of the 17th International Web for All Conference* (2020), 1–12.

373. Ray, P. P. ChatGPT: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope. *Internet of Things and Cyber-Physical Systems* (2023).

374. Wu, T. *et al.* A brief overview of ChatGPT: The history, status quo and potential future development. *IEEE/CAA Journal of Automatica Sinica* **10,** 1122–1136 (2023).

375. Morgan, C. J. Use of proper statistical techniques for research studies with small samples. *American Journal of Physiology-Lung Cellular and Molecular Physiology* **313,** L873–L877 (2017).

376. Dhariyal, B., Nguyen, T. L. & Ifrim, G. *Fast Channel Selection for Scalable Multivariate Time Series Classification* in *International Workshop on Advanced Analytics and Learning on Temporal Data* (2021), 36–54.

377. Dhariyal, B., Le Nguyen, T. & Ifrim, G. Scalable classifier-agnostic channel selection for multivariate time series classification. *Data Mining and Knowledge Discovery* **37,** 1010–1054 (2023).

378. Chen, Y. *et al.* The UCR time series classification archive (2015).

379. Dau, H. A. *et al.* The UCR time series archive. *IEEE/CAA Journal of Automatica Sinica* **6,** 1293–1305 (2019).

380. Slack, D., Hilgard, S., Jia, E., Singh, S. & Lakkaraju, H. *Fooling lime and shap: Adversarial attacks on post hoc explanation methods* in *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society* (2020), 180–186.

381. Gallacher, J. D. Leveraging cross-platform data to improve automated hate speech detection. *arXiv preprint arXiv:2102.04895* (2021).

382. Swamy, S. D., Jamatia, A. & Gambäck, B. *Studying generalisability across abusive language detection datasets* in *Proceedings of the 23rd conference on computational natural language learning (CoNLL)* (2019), 940–950.

383. E. Prescott, M. Big data and competitive advantage at Nielsen. *Management Decision* **52,** 573–601 (2014).

384. Diaz, M. C., Donovan, E. M., Schillo, B. A. & Vallone, D. Menthol e-cigarette sales rise following 2020 FDA guidance. *Tobacco control* (2020).

385. Ambalov, I. A. An Examination of the Influences of Habit, Compatibility, and Experience on the Continued Use of Short-Form Video-Sharing Services: A Case of TikTok. *International Journal of e-Collaboration (IJeC)* **18,** 1–19 (2022).

386. Ferreira, F. P. *Consumers' perception on companies who have a social purpose: the role of influencers* PhD thesis (2022).

387. Basystiuk, O. *et al. The Developing of the System for Automatic Audio to Text Conversion.* in *IT&AS* (2021), 1–8.

388. Tsap, V., Shakhovska, N. & Sokolovskyi, I. *The Developing of the System for Automatic Audio to Text Conversion.* in *MoMLeT+ DS* (2021), 75–84.

389. He, X. & Deng, L. Deep learning for image-to-text generation: A technical overview. *IEEE Signal Processing Magazine* **34,** 109–116 (2017).

390. Andrzejczak, N., Trainin, G. & Poldberg, M. From Image to Text: Using Images in the Writing Process. *International Journal of Education & the Arts* **6,** 1–17 (2005).

391. Qiao, T., Zhang, J., Xu, D. & Tao, D. *Mirrorgan: Learning text-to-image generation by redescription* in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), 1505–1514.

392. Achiam, J. *et al.* Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).

393. Wu, S. *et al.* A comparative study of open-source large language models, gpt-4 and claude 2: Multiple-choice test taking in nephrology. *arXiv preprint arXiv:2308.04709* (2023).

394. Ahmed, I. & Islam, R. Gemini-the most powerful LLM: Myth or Truth. *Authorea Preprints* (2024).

395. Jiang, A. Q. *et al.* Mixtral of experts. *arXiv preprint arXiv:2401.04088* (2024).

396. Kazemi, A., Younus, A., Jeon, M., Qureshi, M. A. & Caton, S. InÉire: An Interpretable NLP Pipeline Summarising Inclusive Policy Making Concerning Migrants in Ireland. *IEEE Access* (2023).

397. Hollmann, N., Müller, S. & Hutter, F. Large language models for automated data science: Introducing caafe for context-aware automated feature engineering. *Advances in Neural Information Processing Systems* **36** (2024).

398. Goel, A. *et al. Llms accelerate annotation for medical information extraction* in *Machine Learning for Health (ML4H)* (2023), 82–100.

399. Tsai, P.-F., Gao, C.-H. & Yuan, S.-M. Stock Selection Using Machine Learning Based on Financial Ratios. *Mathematics* **11,** 4758 (2023).

400. Saraswathi, K., Renukadevi, N., Nandhinidevi, S., Gayathridevi, S. & Naveen, P. *Sales prediction using machine learning approaches* in *AIP Conference Proceedings* **2387** (2021).

401. Duguay, P. A. Read it on Reddit: Homogeneity and ideological segregation in the age of social news. *Social Science Computer Review* **40,** 1186–1202 (2022).

402. Burnham, J. F. Scopus database: a review. *Biomedical digital libraries* **3,** 1–8 (2006).

403. Gaffney, D. & Matias, J. N. Caveat emptor, computational social science: Large-scale missing data in a widely-published Reddit corpus. *PloS one* **13,** e0200162 (2018).

404. Speer, R. *rspeer/wordfreq: v3.0* version v3.0.2. Sept. 2022. https://doi.org/10.5281/zenodo.7199437.

405. Wolf, T. *et al. Transformers: State-of-the-art natural language processing* in *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations* (2020), 38–45.

406. Liu, Y. *et al.* Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).

407. Gamzu, I., Gonen, H., Kutiel, G., Levy, R. & Agichtein, E. Identifying helpful sentences in product reviews. *arXiv preprint arXiv:2104.09792* (2021).

408. Wang, W., Zheng, V. W., Yu, H. & Miao, C. A survey of zero-shot learning: Settings, methods, and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)* **10,** 1–37 (2019).

409. Mukherjee, S., Awadallah, A. H. & Gao, J. *XtremeDistilTransformers: Task Transfer for Task-agnostic Distillation* 2021. arXiv: 2106.04563 [cs.CL].

410. Bhargava, P., Drozd, A. & Rogers, A. *Generalization in NLI: Ways (Not) To Go Beyond Simple Heuristics* 2021. arXiv: 2110.01518 [cs.CL].

411. Bowman, S. R., Angeli, G., Potts, C. & Manning, C. D. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326* (2015).

412. Williams, A., Nangia, N. & Bowman, S. R. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426* (2017).

413. Fellbaum, C. in *Theory and applications of ontology: computer applications* 231–243 (Springer, 2010).

414. Loper, E. & Bird, S. Nltk: The natural language toolkit. *arXiv preprint cs/0205028* (2002).

415. Nivre, J. *et al. Universal dependencies v1: A multilingual treebank collection* in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)* (2016), 1659–1666.

416. Santorini, B. Part-of-speech tagging guidelines for the penn treebank project (3rd revision). *Technical Reports (CIS),* 570 (1990).

417. To, H. Q., Nguyen, K. V., Nguyen, N. L.-T. & Nguyen, A. G.-T. *Gender prediction based on vietnamese names with machine learning techniques* in *Proceedings of the 4th International Conference on Natural Language Processing and Information Retrieval* (2020), 55–60.

418. Shelton, M., Lo, K. & Nardi, B. Online media forums as separate social lives: A qualitative study of disclosure within and beyond Reddit. *IConference 2015 Proceedings* (2015).

419. Li, W. & Dickinson, M. *Gender Prediction for Chinese Social Media Data.* in *RANLP* (2017), 438–445.

420. Kosse, R., Schuur, Y. & Cnossen, G. *Mixing traditional methods with neural networks for gender prediction* in *Proceedings of the Ninth International Conference of the CLEF Association (CLEF 2018)* (2018).

421. Khandelwal, A., Swami, S., Akhtar, S. S. & Shrivastava, M. Gender prediction in english-hindi code-mixed social media content: Corpus and baseline system. *Computación y Sistemas* **22,** 1241–1247 (2018).

422. Węglarczyk, S. *Kernel density estimation and its application* in *ITM Web of Conferences* **23** (2018), 00037.

423. Chen, Y.-C. A tutorial on kernel density estimation and recent advances. *Biostatistics & Epidemiology* **1,** 161–187 (2017).

Appendices

Appendix A: Platforms Used in Customer Need Mining Studies

To generate each graph in Figure 2.1, the Scopus database for articles relating to the platform of interest along with terms that are commonly used in "customer need" mining studies is searched for.[1] Scopus is an abstract and indexing database that provides access to a vast number of Science, Technology, Engineering, and Mathematics (STEM) journals [402]. Specifically, the search requires the occurrence of at least one of the terms of: 1) customer needs; 2) customer requirements; 3) product ideas; and 4) product innovations. It also requires at least one of the terms of: 1) text mining or 2) machine learning.[2] Finally, it requires the social media platform for each graph e.g. "instagram" for the Instagram graph generated. These terms are searched across "All fields" in Scopus.

By searching for the occurrence of the following terms some indication of the number of papers relative to the searched social media platform are expected to be achieved. However, it is of note that terms can occur in studies for reasons other than them being used as the primary platform e.g. "follow the Twitter link for updates regarding our study".

[1] https://www.scopus.com/ - last accessed 07/06/2024

[2] This is done to capture more studies which use automated methods i.e. are related to data analysis

Appendix B: Number of Products Each Month For Each Product Category on Mintel

A time series of the number of product records per month for each product category data is collected for is recorded in Figure B.1, Figure B.2 and Figure B.3. It is possible to form a time series of this nature for each product category due to the timestamp associated with each product record in GNPD (indicating when it was first available for retail). The figures show that there is a large number of products available each month for tracking customer needs over time. Data at each month is required due to the ground truth data curation methods applied in this thesis, requiring data at monthly time intervals/windows.

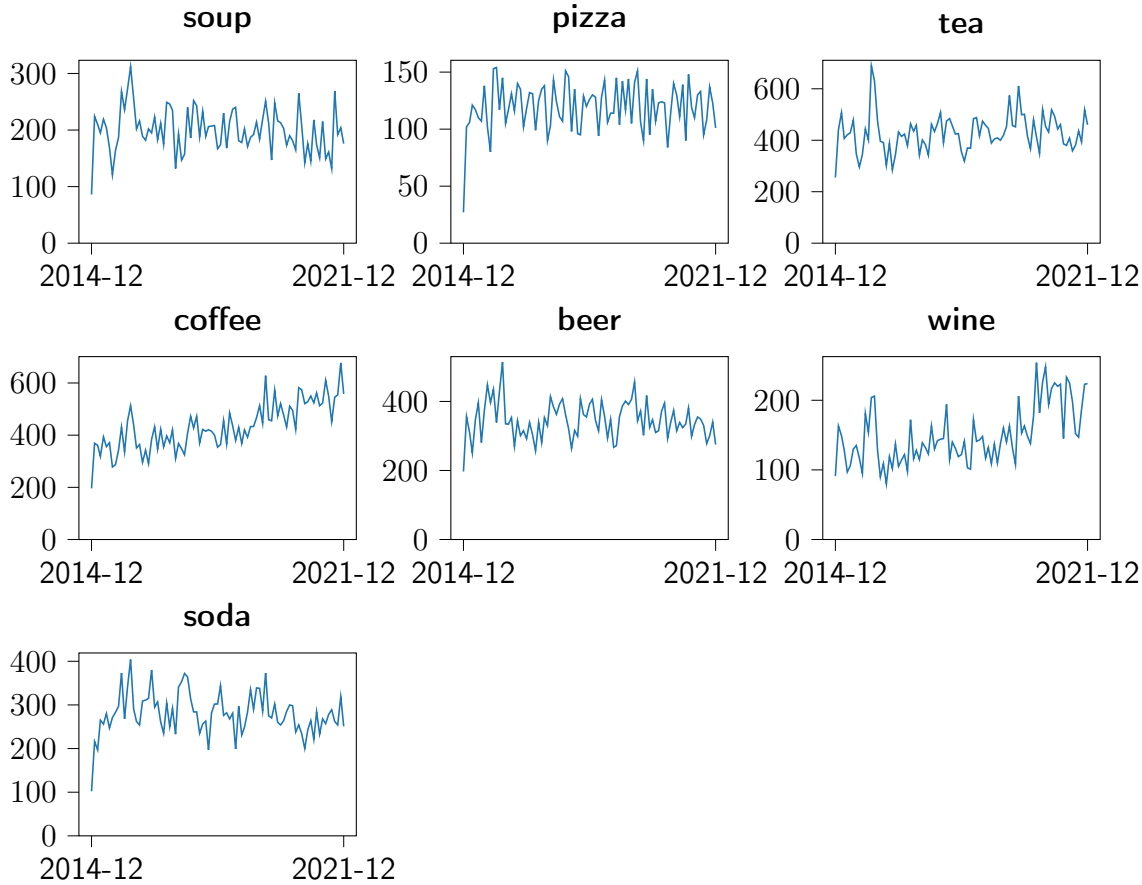

Figure B.1. Time series showing the number of products collected from GNPD each month across every product category (Part 1)

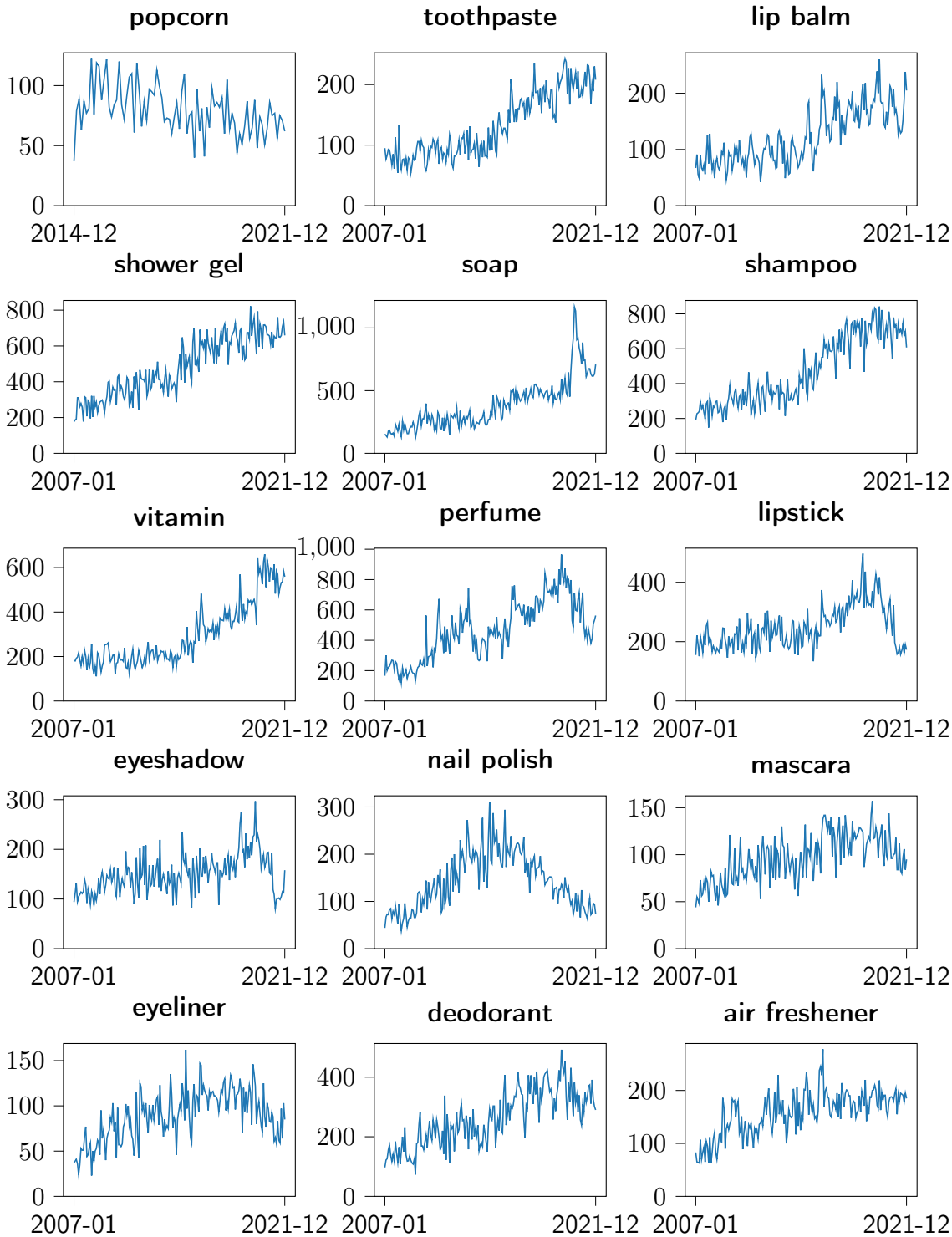

Figure B.2. Time series showing the number of products collected from GNPD each month across every product category (Part 2)

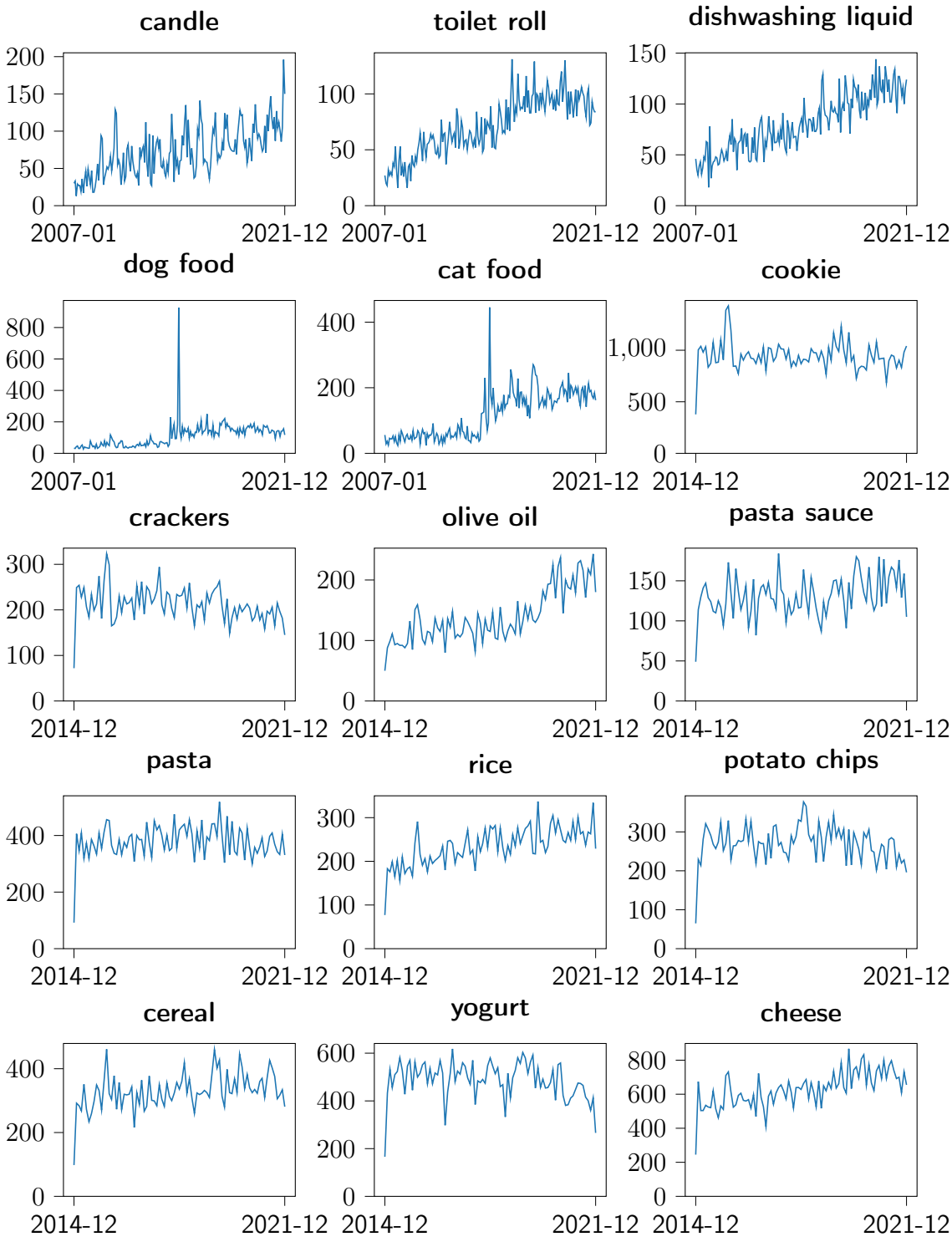

Figure B.3. Time series showing the number of products collected from GNPD each month across every product category (Part 3)

Appendix C: Ground Truth Dataset 2: Annotation Guidelines

This appendix details the guidelines annotators are given prior to labelling keyphases as customer needs. The guidelines are broken into 2 main sections: 1) Main Label Definitions; 2) Additional Guidelines and Edge Cases.

C.1 Main Label Definitions

As discussed in Section 3.4, annotators label into 5 categories during annotation: 1) direct customer need; 2) indirect customer need; 3) not a customer need; 4) "customer need and not a customer need" conflict; and 5) "direct and indirect customer need" conflict. These are labelled based on the following criteria.

A **direct customer need** is a phrase which stands alone as a benefiting specification of a product. They can be nouns (e.g. gingivitis, manicure), adjectives (e.g. microwavable), verbs (e.g. whiten), participle verbs (e.g. baking, whitening, fried) or adverbs (e.g. fresh) Such phrases could include:

- benefits a user gets from using the product e.g. "antiseptic" for soap products, "microwaveable" for popcorn products , "vegan" for ice-cream products or "frying" for olive oil products.
- benefits a user overcomes from using the product e.g. "dry hands" for hand cream products or "gingivitis" for toothpaste products.
- benefits a user can perform as an activity or treatment with the product e.g. "manicure" for nail polish products

A **indirect customer need** is a phrase which is a feature of a product that the user can touch, taste, hear or smell which has benefits associated by its inclusion in the product. These phrases are mainly nouns as they are actual features of the product (i.e. they are real things). Such phrases could include:

- ingredients of products such as "honey" in cereal products or "jasmine" in tea products
- flavors/scents of products such as "raspberry" for shampoo products or "mango" for beer products
- tangible features of the product such as "tongue scraper" for toothbrush products

A **not a customer need** is a phrase which does not fit the description of direct/indirect customer need.

A **customer need and not a customer need** conflict occurs if only some of the product categories in which the phrase appears in adheres to a customer need.

A **direct customer need and indirect customer need** conflict occurs If the phrase is a direct customer need in some product categories but an indirect customer need in others.

In addition to these main definitions, annotators are given the following pointer to help them when annotating:

As indirect customer needs are real things (e.g. ingredients, flavors, features of products etc.) they will be quite easy to label. The main difficulty will be to label direct customer needs as they can be action phrases (e.g. microwavable) which can be more ambiguous to label in the correct class bucket. If you are having trouble deciding whether one of these action phrases are direct customer needs, ask yourself "is the phrase X a customer need of products which contain it" e.g. "is microwavable a customer need of products which contain it OR does it help if you use it" (yes) or "is effortlessly a customer need of products which contain it OR does it help if you use it" (no). An additional point to remember is that all the products under analysis in this study are from the domain of Consumer Packaged Goods, which are mainly grocery store and cosmetic items. Therefore, when labeling direct needs, ask yourself if the phrase is a need for products which can be sold in the supermarket or a pharmacy.

C.2 Additional Guidelines and Edge Cases

Due to some of the Main Definitions (i.e. Section C.1) being in conflict with each other, annotators are given the following additional guidelines which meet edge cases not addressed in the main definitions. These edge cases are mainly built from the pilot study conducted prior to the main annotation task (as detailed in Section 3.4). Usually, the edge cases are in line with the original definitions, hence their inclusion e.g. a customer need phrase is not the name of a company. However, in some cases it is unsure whether a phrase should be labelled as a customer need or not. In these situations a decision is made so that annotators label consistently e.g. a color is a customer need. In addition to these edge cases, clarification is given on how multiword phrases should be labelled, which require extra consideration seeing as they stretch multiple words.

The following lists of additional guidelines are given which annotators should not label a customer need as (be it either direct or indirect):

- The actual product being sold (e.g. toothpaste, cereal etc) or another product which comes as a deal with the product (e.g. toothbrush with toothpaste)
- The phrase contains the word "product" as a single word or in a multiword phrase

- Deals that come as part of the product being sold e.g. "buy one get one free"
- Names of companies (e.g. Oral-B), brands (e.g. Centrum) , social media (e.g. Twitter), stamps of approvals (e.g. EcoCertificate), logos (e.g. EU Green Leaf logo)
- Specific targeted groups of people who promote (e.g. dentists, doctors, celebrities) or actually use the product (e.g. adults, elderly people)
- Anything to do with the packaging of the product such as size of the product (e.g. 100ml, twin pack)
- Region names e.g. "Columbia Valley" and "Bordeaux" for wine products or "Irish" or "Italian" for coffee products
- Phrases from languages other than English
- Phrase descriptions of other things in the product description e.g. "stainless steel" e.g. for a wine product - "this product was made in a stainless steel basin"
- Any marketing "power word" e.g. premium, exquisite, explosive, advanced, effective, hurry, rare, limited, proven, unique etc.[1]
- Anything about the pricing of the product e.g. bargain, affordable, low cost, $2, £12.50 etc.

The following lists of additional guidelines are given which annotators should label a customer need as (be it either direct or indirect):

- the phrase is a specific type of the target product being sold e.g. "sauvignon blanc" for wine products. The specific type of product being sold (e.g. sauvignon blanc) should not be confused with the actual product being sold (e.g. wine) in this case i.e. annotators should label the phrase "sauvignon blanc" but not "wine"
- the phrase represents a group of ingredients and not a specific ingredient e.g. dietary fiber, amino acid, spices etc. These needs should be labeled as indirect needs.
- the phrase is a ingredient/taste/feature which is explicitly said not to be included in the product e.g. paraben in "this product is free from paraben" (indirect need)
- the phrase is a color e.g. yellow, green etc. (indirect need)
- the phrase is an environmental description that the product can be e.g. recyclable, biodegradable etc (direct).

The following list of additional guidelines are given to annotators about how multiword phrases (e.g. tea tree oil) should be labelled. These types of needs require extra consideration as they stretch multiple words.

[1] https://www.masterclass.com/articles/power-words - last accessed 07/06/2024

- Annotators should label a multiword phrase if the words in the phrase add some value to the meaning of each other (direct need) or are actual features of the product (indirect need) e.g. long lasting vision (direct need) or spring water (indirect need). Phrases which add some meaning to an existing direct/indirect need should also be labeled e.g. "healthy" being added to "healthy oxidative balance" or "prevent" being added to "prevent heart disease".
- Do not label a multiword phrase if it contains a meaningless word which is attached to an existing direct/indirect customer need e.g. "good microwavable", "wash coconut", "contain aluminum", "provide health", "more resilient" etc.
- Do not label a multiword phrase if it is part of another larger multiword phrase e.g. do not label "sodium lauryl" as it is part of the larger phrase "sodium lauryl sulfate"
- In the cases where a multiword phrase contains both direct and indirect needs, annotators should label the phrase as a direct or indirect need based on what they think the focus word in the phrase is. For example, "extra sharp cheddar" has the direct need of "extra sharp" and the indirect need of "cheddar". Here the phrase should be annotated as an "indirect need" as the focus is on cheddar. As another example, "white smile" has the indirect need of "white" (as its a color) and the direct need of "smile". Here the phrase should be annotated as a "direct need" as the focus is on smile
- Annotators should not label any multiword phrases if they contain any words from the "annotators should not label ..." guidelines.

Appendix D: Number of Reddit Posts Each Month For Each Analysis

A time series of the number of posts each month across every product category for each analysis in this thesis is recorded in this appendix. Figure D.1 shows the number of posts collected each month for Chapter 4 which makes predictions for 1 category. Figure D.3 shows the number of posts collected each month for Chapter 5 which makes predictions for 37 categories. Finally, Figure D.2 shows the number of posts collected each month for Chapter 6 which makes predictions for 3 categories. The plots in each figure show the total number of posts (y-axis) that are collected each month (x-axis).

An adequate number of posts each month are required as lists of predicted customer need keyphrases are generated each month. The figures in this appendix show this. There are, however, some months where there appears to be a downward spike in the number of posts available for a specific product category e.g. toothpaste in Figure 6. This is due to the well-documented gap in some observations in the Pushshift dataset at certain periods of time [403]. Unfortunately, this is a fact that cannot be controlled.

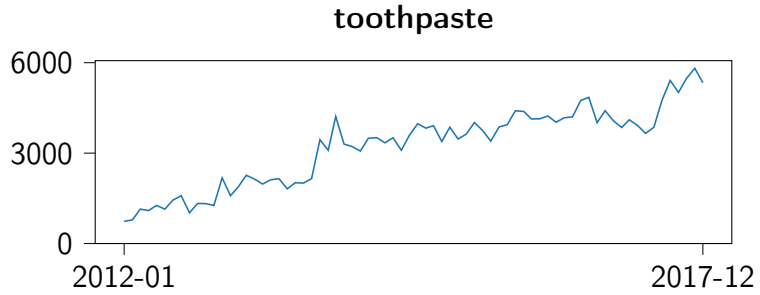

Figure D.1. Number of posts each month collected for the analysis detailed in Chapter 4. The x-axis shows the total number of posts collected each month while the y-axis shows the time period in which posts are collected.

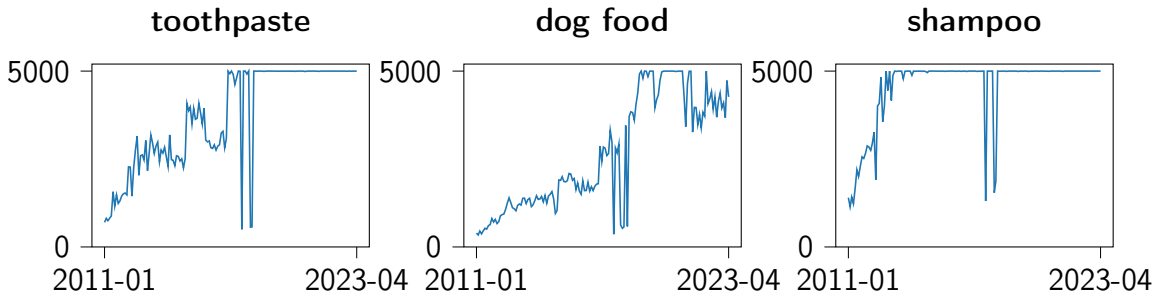

Figure D.2. Number of posts each month collected for the analysis detailed in Chapter 6. The x-axis shows the total number of posts collected each month while the y-axis shows the time period in which posts are collected.

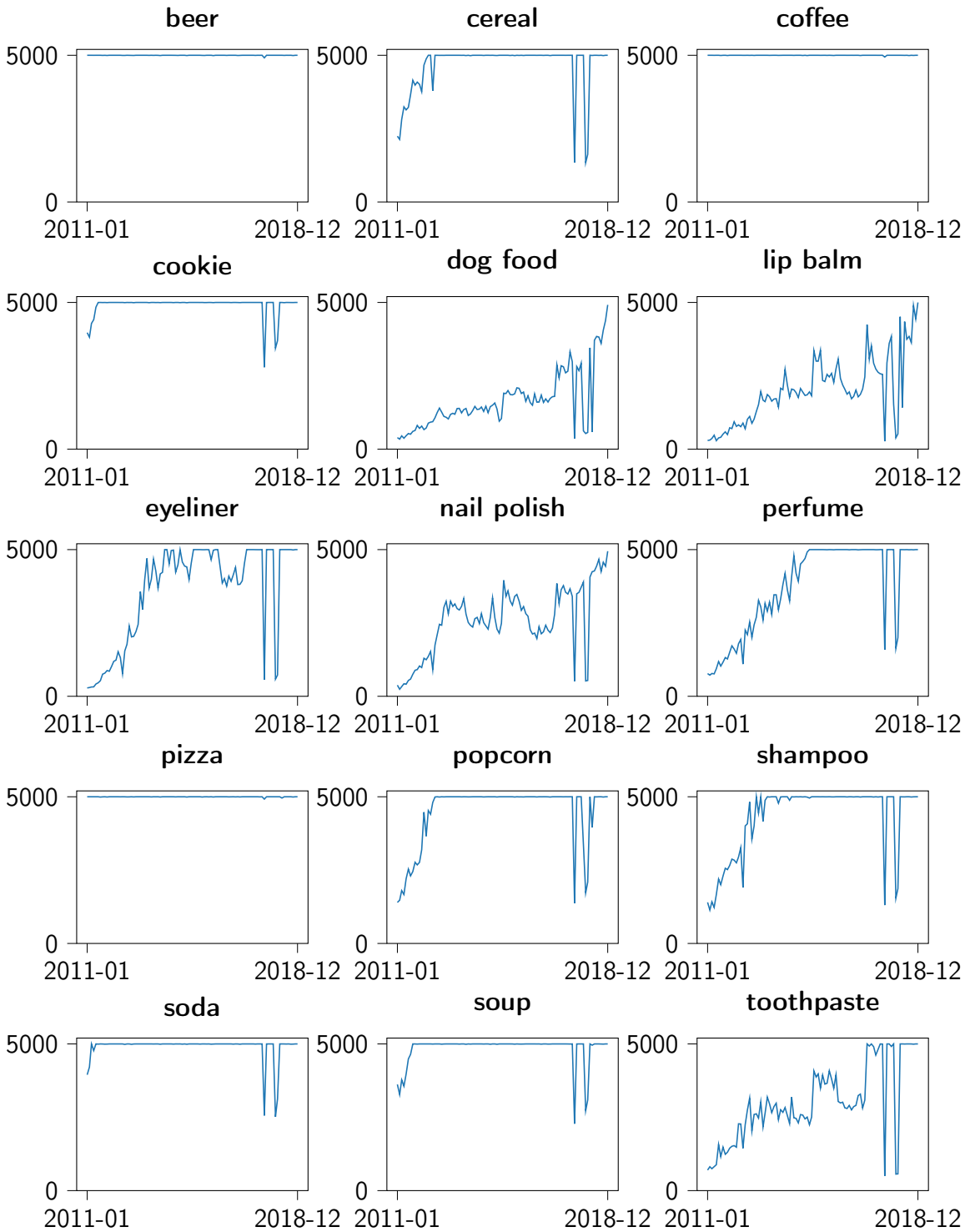

Figure D.3. Number of posts each month collected for the analysis detailed in Chapter 5. The x-axis shows the total number of posts collected each month while the y-axis shows the time period in which posts are collected.

Appendix E: **Rule-based Algorithm Results Distribution**

The results for the parameter range values in Table 4.2 (i.e. Range Values Column) don't deviate much from the mean. In order to illustrate this, in Figure E.1, the distribution of the results for this parameter range for the mean results is plotted in Table 4.3 (i.e. approach A). The figure shows multiple distributions of results for the List Mean Precision and List Recall values over the various values of K, as recorded in the initial experiment (i.e. 5, 10, 15, and 20). Above each distribution, the maximum (max), minimum (min), mean (μ) and standard deviation (σ) are shown for the results. The result of the baseline above each distribution is also provided (i.e. approach B as recorded in Table 4.3). The baseline result as a distribution is not shown as it is a single value. Along with the fact that the results don't seem to deviate too much from each other, it is also noteworthy that the minimum results for each metric of the algorithm over every value of K performs better than the baseline.

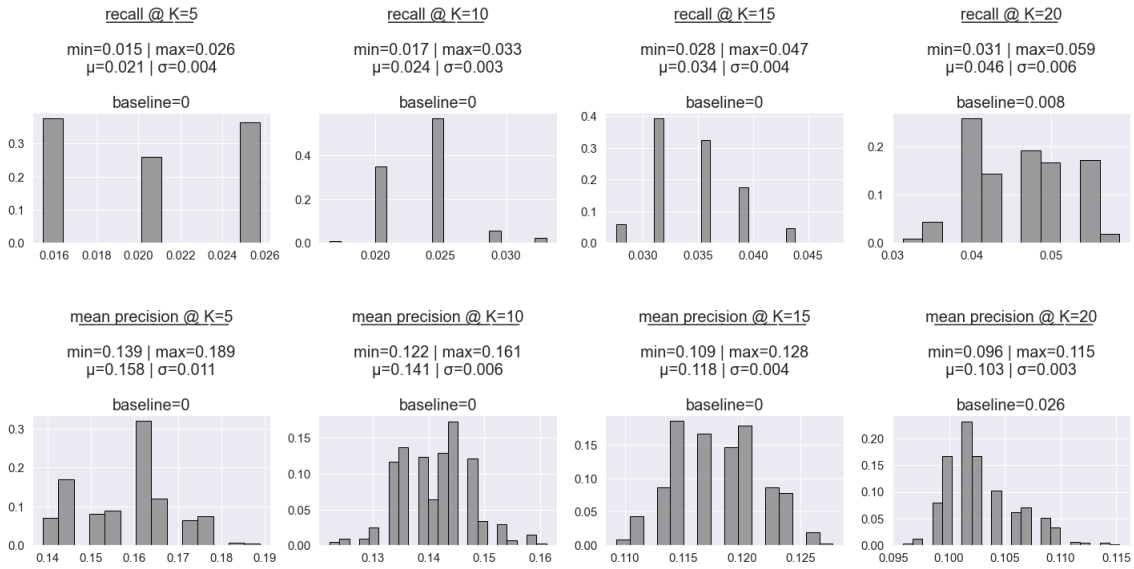

Figure E.1. Table 4.3 Results Distribution

Appendix F: Rule-based Algorithm Results Distribution

The results for the parameter range values in Table 4.2 (i.e. Range Values Column) don't deviate much from the mean. In order to illustrate this, in Figure F.1, the distribution of results are plotted for the "Mean Num Times Detected" and "Mean 1st Date Social Media Detected" columns in Table 4.4. The figure shows multiple distributions only for the needs which were detected by the algorithm i.e. "charcoal", "coconut", "vegan", "eco-friendly" and "bamboo". The results can deviate slightly in some cases (e.g. the "1st Date Social Media" column for the need "eco-friendly"), however, the results generally show low deviation.

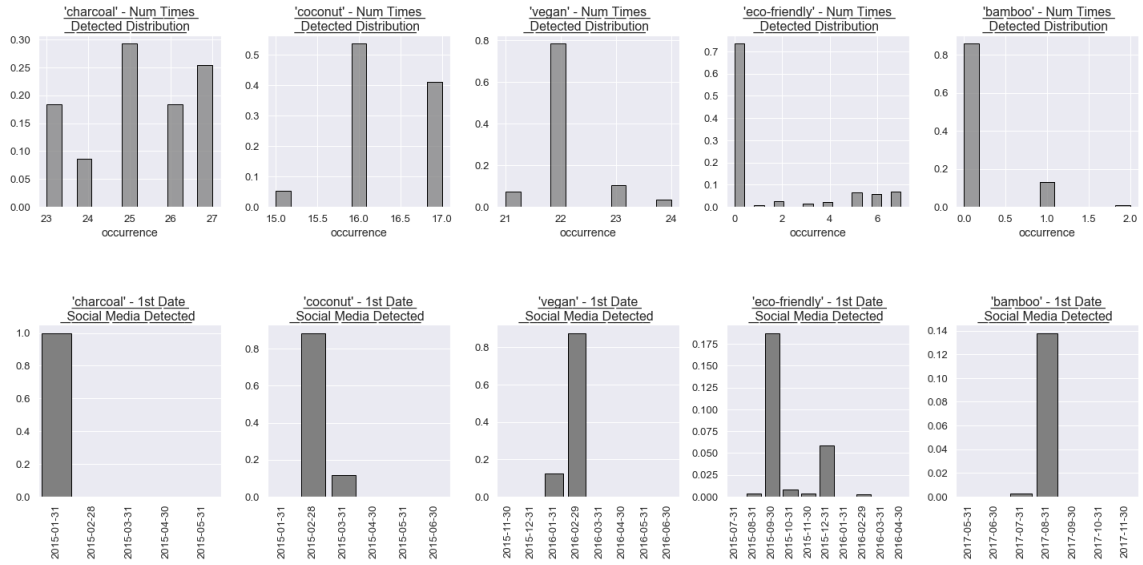

Figure F.1. Table 4.4 Results Distribution

Appendix G: **Reddit Information Based Series**

For the Reddit Information Based Series, 18 features are recorded, as shown in Table G.1. These account for 51 of the univariate time series in the classification task i.e. 26 boolean based series (i.e. 13 features times 2 summary statistics); 2) 12 continuous based series (i.e. 3 features times 4 summary statistics); and 3) 13 string based series (i.e. 5 plus 8 type matches).

The string features analyzed are 1) *thumbnail* and 2) *whitelist_status*. Specifically, the percent of times the *thumbnail* equals a) "self", b) "default", c) "nsfw", d) "image" and e) "spoiler" and *whitelist_status* equals a) "all_ads", b) "no_ads", c) "some_ads", d) "promo_adult_nsfw", e) "house_only", f) "promo_all", g) "promo_adult" and h) "promo_specified". The following values are searched across both the mentioned fields (e.g. self, default, etc. for *thumbnail*) as they are an exhaustive list of the values found in all the Reddit post data collected.

Table G.1. Reddit Features Used in Analysis

Name	Type	Num Series	Name	Type	Num Series	Name	Type	Num Series
is_robot_indexable	bool	2	locked	bool	2	pinned	bool	2
is_original_content	bool	2	no_follow	bool	2	num_comments	cont	4
is_reddit_media_domain	bool	2	over_18	bool	2	num_crossposts	cont	4
is_self	bool	2	send_replies	bool	2	score	cont	4
is_video	bool	2	spoiler	bool	2	thumbnail	str	5
is_crosspostable	bool	2	stickied	bool	2	whitelist_status	str	8

Appendix H: **Frequency Based Series**

For the Frequency Based Series, 4 keyphrase-level features are recorded, as shown in Table H.1. As these are keyphrase-level statistics, they result in 4 univariate time series, as detailed in Section 5.1.3.

As seen in the table, the difference between the *Document Frequency* and the *Relative Document Frequency* is that the *Document Frequency* reports the number of posts the keyphrase appeared in the Fixed Time Window (i.e. month). The *Relative Document Frequency*, on the other hand, reports the total number of posts the keyphrase appeared in divided by the number of posts in the Fixed Time Window (i.e. month).

For the *Document Frequency Compared to a Background Corpus* field, the chi-square test [293] is used for the purposes of distinguishing if there is a significant difference between the keyphrase's expected frequency on Reddit compared to its observed frequency in a large-scale background reference corpus. As in [296–298], the test is computed for each keyphrase using a 2-by-2 contingency table for which a chi-square test statistic is returned. As this statistic only measures if there is a difference between the observed and expected frequency (i.e. high values for big differences and close to zero for small differences), the statistic is multiplied by minus 1 if the expected frequency is greater than the observed frequency. This is done in order to distinguish that the observed frequency is greater than the expected frequency (or vice-versa) to the classification algorithm. The Python library *wordfreq* [404] is used (representative of a normal distribution of words) to act as the background corpus.

Table H.1. Frequency Based Features Used in Analysis

Name	Type	Num Series	Name	Type	Num Series
Document Frequency	keyphrase-level	1	Document Frequency Compared to a Background Corpus	keyphrase-level	1
Relative Document Frequency	keyphrase-level	1	% Posts which are Submissions	keyphrase-level	1

Appendix I: Product Information Based Series

For the Product Based Series, 6 continuous features are recorded, as shown in Table I.1. These 6 continuous features result in 24 univariate time series when summarized, as described in Section 5.1.3. Initially, various pre-trained models distinguishing different types of product information (e.g. buy intent, purchase intent, etc.) are run over the posts to generate features. For each of the models, the probability value of the post being associated with the output class is reported, therefore making it a continuous value (e.g. 0.98) rather than a boolean (e.g. True).

Three separate pre-trained text classification models are run over the posts from the Python library *Hugging Face* [405] to generate the features seen in Table I.1: 1) *Sell/Buy Intent*, 2) *Purchase Intent* and 3) *Review Helpfulness*. For the *Sell/Buy Intent* features, the pre-trained model tries to classify posts into either having "selling" or "buying" intent. An example of buying intent includes "I am looking for the purple ombre dress with floral bodice in a size 12 for my wedding in June this year" while an example of selling intent includes "Boiler over 7 years old".[1] For *Purchase Intent*, features from a *RoBERTa*-based model [406] are generated which is fine-tuned on a dataset of 2000 purchase-intent and non-purchase-intent documents [67].[2] For the *Review Helpfulness* features, a model trained on a dataset of customer reviews from Amazon is used which contains an output label of an Amazon helpfulness score [407].[3]

A pre-trained zero-shot model from *Hugging Face* is used to generate the 3 remaining features: 1) *Discuss Product Features*, 2) *Discuss Product Ideas* and 3) *Discuss Customer Needs*.[4] Zero-shot classification is a learning paradigm that aims to train a model to predict instances belonging to an unseen class [408]. The model is provided with unseen labels to generate more features for the task. Specifically, the model is provided with the following class names: 1) product feature (i.e. *Discuss Product Features*), 2) product idea (i.e. *Discuss Product Ideas*), and 3) customer need (i.e. *Discuss Customer Needs*).

[1] https://huggingface.co/obsei-ai/sell-buy-intent-classifier-bert-mini - last accessed 07/06/2024

[2] https://huggingface.co/j-hartmann/purchase-intention-english-roberta-large - last accessed 07/06/2024

[3] https://huggingface.co/banjtheman/distilbert-base-uncased-helpful-amazon - last accessed 07/06/2024

[4] https://huggingface.co/cross-encoder/nli-distilroberta-base - last accessed 07/06/2024

Table I.1. Product Based Features Used in Analysis

Name	Type	Num Series	Name	Type	Num Series
Sell/Buy Intent	cont	4	Discuss Product Features	cont	4
Purchase Intent	cont	4	Discuss Product Ideas	cont	4
Review Helpfulness	cont	4	Discuss Customer Needs	cont	4

Appendix J: **Sentiment Based Series**

For the Sentiment Based Series, the 28 features recorded are all shown in Table J.1. These 28 continuous features result in 112 univariate time series when summarized, as described in Section 5.1.3. For the single pre-trained sentiment model run over the Reddit posts, the probability value of the post being associated with the output class is reported, therefore making it a continuous value (e.g. 0.98) rather than a boolean (e.g. True).

The pre-trained sentiment model used in the analysis is from *Hugging Face* [405]. It is a fine-tuned version of a *XtremeDistilTransformers* model [409] which is run over the *GoEmotions* dataset [323].[1] The *GoEmotions* dataset itself contains 28 output classes each representing a feeling/sensation e.g. *Anger*, *Caring*, *Disappointment*, *Excitement*, etc. These output classes make up the features in Table J.1.

Table J.1. Sentiment Based Features Used in Analysis

Name	Type	Num Series	Name	Type	Num Series	Name	Type	Num Series
Admir -ation	cont	4	Disapp -roval	cont	4	Nervous -ness	cont	4
Amuse -ment	cont	4	Disgust	cont	4	Neutral	cont	4
Anger	cont	4	Embarr -assment	cont	4	Optim -ism	cont	4
Annoy -ance	cont	4	Excite -ment	cont	4	Pride	cont	4
Approval	cont	4	Fear	cont	4	Realiz -ation	cont	4
Caring	cont	4	Gratitude	cont	4	Relief	cont	4
Confusion	cont	4	Grief	cont	4	Remorse	cont	4
Curiosity	cont	4	Joy	cont	4	Sadness	cont	4
Desire	cont	4	Love	cont	4	Surprise	cont	4
Disapp -ointment	cont	4						

[1] https://huggingface.co/bergum/xtremedistil-l6-h384-go-emotion - last accessed 07/06/2024

Appendix K: Question Detection Based Series

For the Question Detection Based Series, the 5 features recorded are all shown in Table K.1. These 5 continuous features result in 20 univariate time series when summarized, as detailed in Section 5.1.3. For the pre-trained models run over the Reddit posts, the probability value of the post being associated with the output class is reported, therefore making it a continuous value (e.g. 0.98) rather than a boolean (e.g. True).

The first model tries to detect whether a post is asking a question or stating an answer i.e. question answer model. A pre-trained model for this task from *Hugging Face* is used which is trained on a dataset of questions and statements from *Kaggle*.[1][2] This generates the two features in Table K.1: a) *Question* and b) *Statement*.

The second model tries to detect whether a statement is true (*Entailment*), false (*Contradiction*), or undetermined (*Neutral*) in the way it is expressed i.e. Natural Language Inference (NLI). The pre-trained model used for this is also from *HuggingFace* [410].[3] This generates three features in Table K.1: a) *Entailment*, b) *Contradiction* and c) *Neutral*.

Table K.1. Question Based Detection Features Used in Analysis

Name	Type	Num Series	Name	Type	Num Series
Question	cont	4	Contradiction	cont	4
Statement	cont	4	Neutral	cont	4
Entailment	cont	4			

[1] https://huggingface.co/shahrukhx01/bert-mini-finetune-question-detection - last accessed 07/06/2024

[2] https://www.kaggle.com/stefanondisponibile/quora-question-keyword-pairs - last accessed 07/06/2024

[3] https://huggingface.co/prajjwal1/bert-mini-mnli - last accessed 07/06/2024

Appendix L: Embedding Based Series

For the Embedding Based Series, 2 main types of features are recorded, as shown in Table L.1. These 2 main types of features can be split up into 75 continuous features for *Document Embeddings* (i.e. 300 univariate time series) and 50 keyphrase-level features for *Phrase Embeddings* (i.e. 50 univariate time series), resulting in 350 univariate time series when summarized, as detailed in Section 5.1.3. In this section, embeddings on the post (or document) and keyphrase level are generated.

Table L.1. Embedding Based Features Used in Analysis

Name	Type	Num Series
Document Embeddings	cont	300
Phrase Embeddings	keyphrase-level	50

For the post-level embeddings (i.e. *Document Embeddings*), the Python libraries *spaCy* [313] and *SBERT* [324] are used. Specifically, the *en_core_web_lg* model from *spaCy* (as used in Section 5.1.2) and the *all-MiniLM-L6-v2* model from *SBERT* are used.[1] The *en_core_web_lg* and the *all-MiniLM-L6-v2* models produce 300 and 384 dimensional embeddings respectively. As each dimension produced by the model needs to be summarized as a continuous feature (as described in Section 5.1.3), this leads to major increases in the running time of the experiment if all of the features produced by the model were to be summarized.

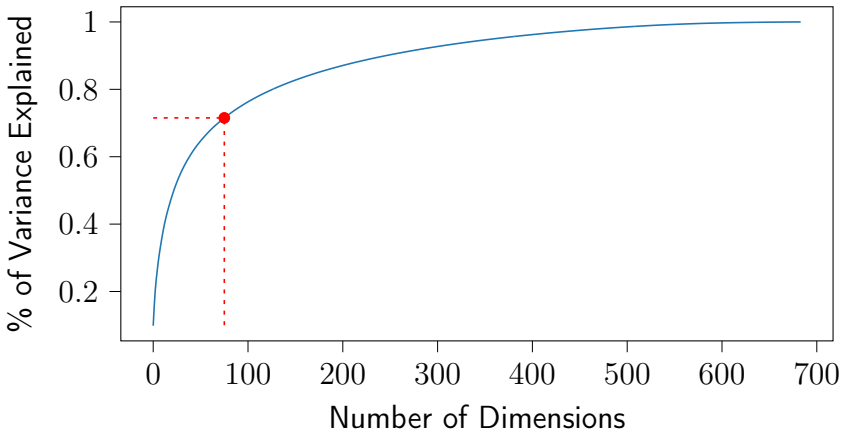

Figure L.1. Percentage of Explained Variance Plot for PCA run over Document Embeddings - 75 components kept capturing 71.5% of the variance

To mitigate this time complexity issue, PCA is run on the outputs of the embedding models and the fitted model is used to transform new input data to a lower dimensional space. Specifically, a PCA model is fit on the document embeddings output of the *SBERT* and *spaCy* models concatenated. This is done as there are correlations between the two embedding outputs. The suggested approach provided by *SBERT* is followed when training the

[1] https://www.sbert.net/docs/pretrained_models.html - last accessed 07/06/2024

PCA model, which recommends training on the transformed embedding output of 20,000 random documents from the *ALLNLI* dataset.[2] The *ALLNLI* dataset represents a highly general corpus - which is a combination of the *SNLI* [411] and *MultiNLI* datasets [412].[3] Figure L.1 shows a proportion of variance explained plot for the PCA model run over the concatenated document embeddings, which shows the % of variance explained on the y-axis and the number of dimensions on the x-axis. 75 dimensions are selected to use in this analysis as it seems to provide a good proportion of the variance at a low number of components (as seen in Figure L.1). To recap, when generating the *Document Embedding* series, the document embeddings produced by the *SBERT* and *spaCy* models are first concatenated. Then, the embeddings are transformed using the discussed trained PCA model and the first 75 components are taken. These 75 new features are then summarized to form 300 univariate time series (detailed in Section 5.1.3).

Embeddings on the phrase level are also recorded by using pre-trained word vectors from the Python library *fasttext* [325]. Specifically, the *crawl-300d-2M-subword* model from *fasttext* is used which produces 300 dimensions.[4] The model is very useful as it can generate an embedding for any keyphrase as it uses subword information in its training. This can be helpful when dealing with the diverse range of phrases and misspellings on Reddit. As with the *Document Embeddings*, PCA is applied to retrieve a lower number of dimensions for time complexity purposes. Figure L.2 shows a proportion of variance explained plot for the PCA model run over the phrase embeddings, which shows the % of variance explained on the y-axis and the number of dimensions on the x-axis. 50 dimensions are selected to use as they seemed to provide a good proportion of the variance at a low number of components (as seen in Figure L.2). To recap, when generating the *Phrase Embedding* series, first an embedding for the candidate keyphrase is generated using the described *fasttext* model. Then, the embeddings are transformed using the discussed PCA model and the first 50 dimensions are selected. Unlike the *Document Embeddings*, these embeddings are calculated on the keyphrase level and are hence not summarized (detailed in Section 5.1.3).

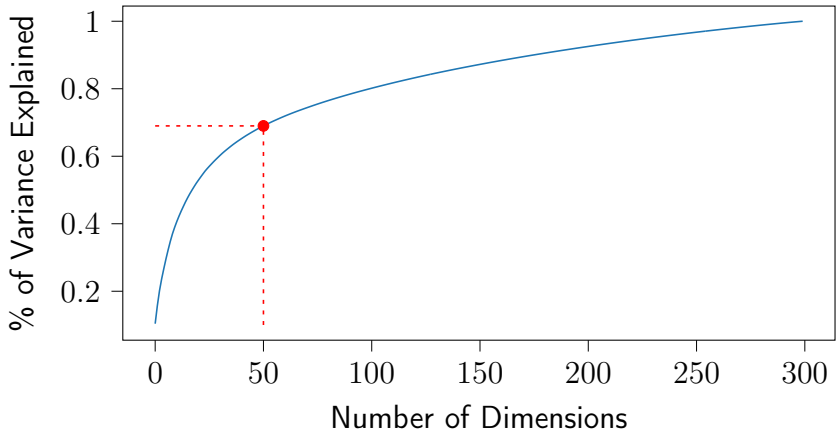

Figure L.2. Percentage of Explained Variance Plot for PCA run over Phrase Embeddings - 50 components kept capturing 69% of the variance

[2] https://github.com/UKPLab/sentence-transformers/blob/master/examples/training/distillation/dimensionality_reduction.py - last accessed 07/06/2024

[3] https://www.sbert.net/examples/datasets/README.html#allnli-dataset - last accessed 07/06/2024

[4] https://fasttext.cc/docs/en/english-vectors.html - last accessed 07/06/2024

Appendix M: **Subreddit Based Series**

For the Subreddit Based Series, 100 search string features are recorded. As these are search string features, they result in 100 univariate time series, as detailed in Section 5.1.3.

Specifically, 100 subreddit strings are searched for from the most subscribed subreddits at the time of experimentation. These subreddits are found from a website containing an updated list of the most subscribed subreddits.[1] The names of these subreddits are shown in Table M.1.

Table M.1. Subreddit Names used as Search Strings in the analysis

Name	Name	Name	Name
announcements	DIY	wallstreetbets	dadjokes
funny	mildlyinteresting	wholesomememes	AnimalsBeingBros
AskReddit	sports	AdviceAnimals	tattoos
gaming	space	interestingasfuck	buildapc
aww	gadgets	Fitness	photography
Music	Documentaries	politics	AnimalsBeingJerks
pics	tifu	WTF	nba
worldnews	photoshopbattles	oddlysatisfying	BikiniBottomTwitter
science	GetMotivated	travel	Damnthatsinteresting
todayilearned	UpliftingNews	lifehacks	MadeMeSmile
movies	listentothis	Minecraft	FoodPorn
videos	television	relationship_advice	instant_regret
news	memes	facepalm	gardening
Showerthoughts	dataisbeautiful	BlackPeopleTwitter	reactiongifs
EarthPorn	history	NatureIsFuckingLit	AnimalsBeingDerps
food	philosophy	Whatcouldgowrong	woahdude
IAmA	InternetIsBeautiful	leagueoflegends	WatchPeopleDieInside
askscience	Futurology	bestof	Overwatch
Jokes	WritingPrompts	pcmasterrace	mildlyinfuriating
gifs	OldSchoolCool	me_irl	PewdiepieSubmissions
nottheonion	nosleep	dankmemes	programming
LifeProTips	personalfinance	nextfuckinglevel	PublicFreakout
books	creepy	Tinder	pokemon
explainlikeimfive	TwoXChromosomes	PS4	ContagiousLaughter
Art	technology	Unexpected	EatCheapAndHealthy

[1] https://redditlist.com/ - last accessed 07/06/2024

Appendix N: **Kansei Engineering Based Series**

For the Kansei Engineering Based Series, 32 boolean features are recorded, as shown in Table N.1. As these are boolean statistics with no NaN values they result in 32 univariate time series, as detailed in Section 5.1.3.

As detailed in Section 5.1.4, to integrate knowledge from Kansei Engineering into the model, 16 Kansei attributes are identified. Each Kansei attribute makes up 2 features in the model, as each consists of 2 sets of bipolar words where users select their feelings towards a product e.g. common and unique. The entire process from [213] is followed to identify these sets of words i.e. 16 Kansei attributes which make up 32 sets of words. The initial words in each set are obtained from the literature. These words are then expanded by finding synonyms in each set. They are further expanded by finding antonyms in each of the bipolar groups, which are then added to the corresponding group for the Kansei attribute e.g. add antonyms for "common" to the "unique-personalized-rare" group. To find these synonyms and antonyms, Wordnet is used [413] through the Python library *NLTK* [414]. The defined rules in [213] are also followed for resolving conflicts when a new word appears in two opposing Kansei groups. These expanded lists of Kansei groups are then used to classify a post as being associated with it, by the post containing a word in one of the Kansei groups. As stated, this results in 32 boolean features added to each post, which creates 32 univariate time series as there are no NaN values present when calculating this feature. To note, each single row in Table N.1 shows a Kansei attribute with the left column usually donating a positive group and the right a negative group (e.g. *Elegant* vs *Artless* belong to two different groups but the same attribute).

Table N.1. Kansei Based Features Used in Analysis

Name	Type	Num Series	Name	Type	Num Series
Elegant	bool	1	Artless	bool	1
Simple	bool	1	Complex	bool	1
Comfortable	bool	1	Restrained	bool	1
Classic	bool	1	Hi-tech	bool	1
Soft	bool	1	Hard	bool	1
Loose	bool	1	Coarse	bool	1
Quality	bool	1	Unreliable	bool	1
Personalized	bool	1	Common	bool	1
Stylish	bool	1	Traditional	bool	1
Luxurious	bool	1	Low-cost	bool	1
Portable	bool	1	Bulky	bool	1
Pleasant	bool	1	Unpleasant	bool	1
Fresh	bool	1	Boring	bool	1
Practical	bool	1	Useless	bool	1
Bright	bool	1	Dim	bool	1
Professional	bool	1	Amateur	bool	1

Appendix O: **Linguistic Based Series**

For the Linguistic Based Series, 38 features are recorded, as shown in Table O.1. These result in 456 univariate time series, as detailed in Section 5.1.3. As discussed in Section 5.1.4, the families of linguistic features recorded are: 1) tagging information, 2) document information and 3) phrase-level information. For most of the features recorded in this section, the *en_core_web_lg* model from spaCy [313] is used, as used in Section 5.1.2.

For tagging, information related to POS tags, dependency labels and named entities are recorded. When recording POS, information provided by *spaCy* is considered which is in the Universal Dependencies format [415] and the Penn Treebank format [416]. The Universal Dependencies format records general POS tags (e.g. verbs or adjectives), while the Penn Treebank format records POS information in a more in-depth manner (e.g. gerund verbs and comparative adjectives).[1] As these POS tags are strings, string matching is performed to generate features which are later turned into univariate time series, as described in Section 5.1.3. The chosen strings matched for in the "*POS tags (Uni Dep)*" field (i.e. Universal Dependencies format) and the "*POS tags (Penn Treebank)*" field (i.e. Penn Treebank format) are 1) verbs, 2) adjectives, 3) nouns, 4) proper nouns and 5) adverbs. This is done as these are the only POS tags accepted when generating candidate keyphrases, as described in Section 5.1.2. Specifically, all combinations of the Universal Dependencies tag list format (which contains a total of 5 tags) and the Penn Treebank tag list format (which contains a total of 17 tags) within an n-gram range of 1-2 grams are searched for. An example of a one-gram POS string could be a single verb, while a two-gram POS string may be a verb-adjective pair. Combinations of these tags generate 30 new univariate series for the "*POS tags (Uni Dep)*" field (i.e. 25 two-gram strings plus 5 one-gram strings) and 306 new series for the "*POS tags (Penn Treebank)*" field (i.e. 289 two-gram strings plus 17 one-gram strings). An n-gram range above 2 grams is not searched for as it would result in too many new features, which would lead to a considerable increase in computational complexity when classifying time series e.g. a three-gram range for the *POS tags (Penn Treebank)* field would result in 2744 more univariate series. Due to this, the strings searched for are changed by instead including a string as a match if it is contained as a subset of a searched string e.g. a noun-noun-verb string would be matched by a searched noun-noun string. When searching for dependency labels (i.e. *Dep tags* field), all of the 45 labels provided by spaCy (which are trained on OntoNotes 5.0 [314]) are searched across.[2] Only search for direct dependency labels and not co-occurring ones (as with POS tags) are searched for as doing it for two grams would add a total of 2025 more univariate time series. For the same reasons as with POS tags, it is considered a match if it contains a subset. *SpaCy* also performs NER and provides tagging for its 18 named entities i.e. the *NER tags* field. These entities are searched for and added as univariate series into the model. SpaCy also includes the Inside Outside Beginning (IOB) tags of these recorded entities (i.e. the

[1] These POS labels can be accessed under token.pos_ (Universal Dependencies) and token.tag_ (Penn Treebank) in spaCy - https://spacy.io/usage/linguistic-features#pos-tagging - last accessed 07/06/2024

[2] https://spacy.io/models/en#en_core_web_lg - last accessed 07/06/2024

IOB tags field) - these tags are also searched for and added as series.

For the document-level linguistic information, general statistical information about the document is recorded e.g. how many tokens are in the post. Information about a lot of these series can be found in spaCy's *span* documentation.[3]

For the phrase-level linguistic information, general statistical information about each phrase is recorded e.g. number of vowels. Information about a lot of these series can be found in spaCy's *token* documentation.[4] Some additional simple phrase-level information is manually created i.e. *num_vowels*, *contains_@*, *contains_#*, *num_tokens*, *contains_original* (i.e. contains the corresponding Target Keyphrase in Table 3.6 - "lip balm" for a dataset of lip balm products) and *contains_original_subset* (i.e. contains part of the corresponding Target Keyphrase in Table 3.6 - "lip" for a dataset of lip balm products). The features made manually are created using simple logic e.g. the binary feature *contains_@* is created by string searching for an @ symbol.

Table O.1. Linguistic Features Used in Analysis

Name	Type	Num Series	Name	Type	Num Series	Name	Type	Num Series
POS tags (Uni Dep)	string	30	contains_non_lower	bool	1	contains_space	key-phrase	1
POS tags (Penn Treebank)	string	306	contains_oov	bool	1	contains_stop	key-phrase	1
Dep tags	string	45	length_text	cont	4	contains_email	key-phrase	1
IOB tags	string	3	token_index	cont	4	contains_num	key-phrase	1
NER tags	string	18	char_index	cont	4	contains_url	key-phrase	1
end	cont	4	contains_non_alpha	key-phrase	1	contains_@	key-phrase	1
end_char	cont	4	contains_non_ascii	key-phrase	1	contains_#	key-phrase	1
start	cont	4	contains_currency	key-phrase	1	length_lemma	key-phrase	1
start_char	cont	4	contains_digit	key-phrase	1	num_vowels	key-phrase	1

[3] https://spacy.io/api/span - last accessed 07/06/2024
[4] https://spacy.io/api/token - last accessed 07/06/2024

Table O.1 Continued: Linguistic Features Used in Analysis

Name	Type	Num Series	Name	Type	Num Series	Name	Type	Num Series
contains_upper	bool	1	contains_left_punct	key-phrase	1	num_tokens	key-phrase	1
contains_title	bool	1	contains_punct	key-phrase	1	contains_original_subset	key-phrase	1
contains_sent_start	bool	1	contains_quote	key-phrase	1	contains_original	key-phrase	1
contains_sent_end	bool	1	contains_right_punct	key-phrase	1			

Appendix P: User Based Series

For the User Based Series, 8 features are recorded, as shown in Table P.1. These features result in 114 univariate time series, as detailed in Section 5.1.3.

The types of features recorded for users follows the literature, which can be divided into two broad categories: 1) personal user information and 2) user interaction information. For personal information, the types of features include publicly listed details on the user's profile e.g. gender, age, location, number of friends/followers/connections, etc. [194, 207]. For interaction information, users are linked to form a network graph using some type of relation (e.g. follower/followee). Statistics are then calculated on the graph to measure how users interact e.g. number of edges, strength distribution, shortest path, etc. [194]. In this analysis, it is challenging to create features for authors due to the Pushshift API not yet providing access to user-level information.[1] In comparison to Pushshift (i.e. historical API), the general Reddit API also doesn't provide a lot of personal user information at the time of experimentation e.g. age, gender, location, etc.[2] It also has a limit of five times less than Pushshift [164]. This makes extracting information from it infeasible from a time complexity perspective, as users are coming from more than 4 million posts in this analysis which will need to have data collected for them. Therefore when creating personal and interaction information for users in this analysis, the features that can be generated are restricted.

For personal user information, different types of features other than publicly listed details on the user's profile are generated (as discussed). Specifically, the author's username is mined as it's some of the only personal information available for users through the Pushshift API. However, in addition to not being able to extract a lot of user information from Reddit, it's also difficult to infer attributes from author's usernames as done in previous studies e.g. predict gender from usernames [417]. This is because authors on Reddit use fake names (e.g. Stuck_In_the_Matrix) instead of real ones [418]. Due to the mentioned factors, how user-based features are generated differs slightly from the norm. Although it's not their real name, a basic analysis of the username field is carried out which is provided with each post when generating features about the personal information of Reddit authors. Specifically, defined strings in usernames are searched for to extract high-level information about them. Some of these defined strings may include "bot" or "mod" (as in Chapter 4) which are commonly put in the names of internet robots and moderators respectively. It'd be useful to distinguish these types of authors as they don't discuss customer needs but rather post spam [287, 288] or point out lapses in other authors' "reditiquette" [286]. Specifically, the mentioned searched strings are generated automatically instead of defining ones e.g. bot or mod. This is done as a data-driven way to find the most common

[1] As of 12/04/2023 Pushshift has not developed an endpoint to "analyze a Reddit user's activity" - https://github.com/pushshift/api#list-of-endpoints - last accessed 07/06/2024

[2] https://praw.readthedocs.io/en/stable/code_overview/models/redditor.html - last accessed 07/06/2024

substrings in author's usernames is more useful instead of guessing them. To find these substrings, author usernames from 20,000 random Reddit posts are collected using the Pushshift API.[3] Across these usernames, the 100 most common substrings 3-8 character n-grams in length are searched for. To search for these substrings using the following parameters the Bag-of-Words model in *sklearn* is used.[4] For the n-gram range (i.e. 3-8), a minimum of 3 n-grams is chosen as it's desired that the substrings at least capture some high-level character information (e.g. not 2-grams like "as" or "an") while a maximum of 8 n-grams is chosen for the upper range as most substrings don't exceed 8 grams.[5] The total number of substrings is chosen to be 100 it is desired to keep the number of substring searches low for computational purposes. From the list of character substrings generated some pertain to interesting topics. Two include "man" and "her" which help infer the gender of Reddit authors. In fact, this use of character-level n-gram information from usernames has been used to predict gender on social media before [419–421]. Another example from the list includes the substring "red", which may infer other information about the authors. A complete list of Reddit username substrings can be found in the GitHub repository which accompanies this study. As described in Section 5.1.3, 100 univariate time series are generated from these substrings by matching strings in the author's username associated with each Reddit post.

For the user interaction information, authors are linked using a network graph for a different type of relation other than some measure of friendship (e.g. follower/followee/friend), as this information is not available through the Pushshift API. Specifically, a graph is instead formed by linking authors to the subreddits they post in i.e. subreddit graph. Graph-based statistics are then calculated which make up the interaction features. For all the statistics calculated on the network graph in this analysis, the Python library *networkx* is used.[6] To calculate statistics for this information two graphs are created: 1) *All Posts Graph* and 2) *Sub Posts Graph*. The *Sub Posts Graph* is a subgraph of the *All Posts Graph*. Graph-based statistics are calculated on the *Sub Posts Graph* which make up the the user-interaction features in this analysis. The two graphs along with the features are calculated each Fixed Time Window as with all the other features in this analysis (discussed in Section 5.1.3). When creating the *All Posts Graph*, all the posts in the Fixed Time Window are taken and the authors from the subreddits they post in are linked. The authors are linked in this way due to restrictions on accessing user-level information through the Reddit API's. From these interactions, an undirected graph is formed where the nodes are authors and the edges are chosen based on if the users post in the same subreddit during the Fixed Time Window of analysis. An undirected graph is used as if the authors post in the same graph they are deemed to be connected, which is unlike a situation in a directed graph e.g. follower/followee network. When forming the *Sub Posts Graph*, a subgraph of the *All Posts Graph* is formed. This only contains users from the posts in which the candidate keyphrase is mentioned. This way user-level information can be extracted at the candidate keyphrase level. How the *Sub Posts Graph* is formed is the same as [194], which calculates user-

[3] These collected posts are independent of the main data collection process (described in Chapter 3)

[4] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html - last accessed 07/06/2024

[5] Many of the substrings used in this analysis are of low n-gram range (e.g. 3 grams like "bot") as the most common 100 grams are taken

[6] https://networkx.org/ - last accessed 07/06/2024

interaction features when classifying hashtags as "organic" or "promoted". As discussed, the graph-based features calculated are recorded in Table P.1. A lot of the features are picked based on speed i.e. are computed quickly. Some of the features in the table are quite self-explanatory (e.g. *Num Nodes*, *Num Edges*, *Density* and *Density of Largest Connected Component*), however, some are not. For the "*Degree of each Node*" feature, the degree of each node is found for the *Sub Posts Graph* and the feature is treated as a continuous one - making up 4 univariate time series.[7] For the *Num Nodes in Connected Components* feature, the connected components of the *Sub Posts Graph* are generated and then again the feature is treated as a continuous one - making up 4 univariate time series.[8]

Although not discussed in Section 5.1.4, the *author_premium* field provided by Pushshift is also added to this family of features (i.e. User Based Series). This is one of the only personal user-based features provided by Pushshift. This feature is included in this section rather than Appendix G (i.e. Reddit Based Series) as it is an attribute of users.

Table P.1. User Features Used in Analysis

Name	Type	Num Series
Username Substrings	str	100
Num Nodes	keyphrase-level	1
Num Edges	keyphrase-level	1
Degree of each Node	cont	4
Num Nodes in Connected Components	cont	4
Density	keyphrase-level	1
Density of Largest Connected Component	keyphrase-level	1
author_premium	bool	2

[7] https://networkx.org/documentation/stable/reference/classes/generated/networkx.Graph.degree.html - last accessed 07/06/2024

[8] https://networkx.org/documentation/stable/reference/algorithms/generated/networkx.algorithms.components.connected_components.html - last accessed 07/06/2024

Appendix Q: **Baseline Parameters**

Table Q.1. Hyper-parameters used in the baseline approach (i.e. Chapter 4)

Parameter Name	Parameter Description	Product Category	Parameter Value	Step Size	Range Values
Gold Standard Subreddit	Subreddit related to the analyzed product	Toothpaste Dog Food Perfume	r/Dentistry r/dogs r/fragrance	x	x
Google Trends Category	Google trend category related to the analyzed product	Toothpaste	Oral & Dental Care	x	x
		Dog Food	Dogs		
		Perfume	Perfumes & Fragrances		
% Most Similar to Gold Standard Subreddit	Controls the number of posts used in the analysis by excluding posts based on their similarity to the *Gold Standard Subreddit* parameter	All	x	0.01	0.05 - 0.2
Social Media Min Document Frequency	The min doc freq of a keyphrase in a set of posts in order for it to be considered a need	All	x	0.00001	0.00005 - 0.0002
Min Chi Square P-value	The min chi square value a keyphrase must have when it's frequency on Reddit is compared to a reference corpus	All	x	0.01	0.01 - 0.03

The baseline algorithm (i.e. algorithm described in Chapter 4) requires many parameters for each category. Table Q.1 shows these parameters and includes a brief explanation of their role in the algorithm. Although not stated in the baseline, these parameters can be broadly split into two types: a) dynamic and b) static.

The dynamic parameters are searched across multiple different values and are the same for each of the categories analyzed in the baseline. These consist of a) *% Most Similar to Gold Standard Subreddit*; b) *Social Media Min Document Frequency*; and c) *Min Chi Square P-value*. An exhaustive grid search is carried out in the baseline study, which tries out multiple values for these 3 parameters to see how it affects the performance of finding future customer needs in the Toothpaste product category (as described in Chapter 4).

During the evaluation of the baseline (i.e. Section 4.2), it wasn't desired to only show that a specific combination of values for these parameters performs well e.g. when the *% Most Similar to Gold Standard Subreddit* is 0.06, the *Social Media Min Document Frequency* is 0.00009 and the *Min Chi Square P-value* is 0.02. Instead, a value range for each of the parameters where the model performs well was reported i.e. not overfitting. All the possible values for a parameter are donated by the Range Values and Step Size columns in Table Q.1 e.g. the *Min Chi Square P-value* parameter uses the values 0.01, 0.02 and 0.03 as the Range Values is 0.01 - 0.03 and the Step Size is 0.01.[1] As multiple combinations of values are tried out for each parameter the mean results for each metric in the evaluation is reported, as done in the baseline experiment. The mean results are also reported in Chapter 5, however, for different reasons than the baseline i.e. stochastic processes performed when transforming data not trying out different combinations of parameters (as detailed in Section 5.2.1). Although these parameter ranges are only found for the Toothpaste category (the only category analyzed in the baseline experiment), these ranges are also used for the other two categories analyzed i.e. Dog Food and Perfume. This is done in the lack of other recommended parameter values for these categories along with the need to do a multi-category baseline comparison.

Different from dynamic parameters, the static parameters are not searched across multiple different values and are different for each of the categories analyzed. These consist of a) *Gold Standard Subreddit* and b) *Google Trends Category*. The *Gold Standard Subreddit* is used in the data reduction step and is defined as "the subreddit which is related to the product under analysis" (Chapter 4 - Table 4.1). The baseline uses the subreddit r/Dentistry for the Toothpaste category for this parameter. The baseline approach is followed by using the same subreddit for the Toothpaste category. However, the subreddits r/fragrance for the Perfume category and r/dogs for the Dog Food category are selected for the two remaining categories. This is done as these subreddits are highly similar to the product category under analysis (e.g. r/fragrance discusses Perfume products) and due to the lack of the baseline study providing any recommendations for this parameter for these categories. The *Google Trends Category* is used when collecting Google Trends data and impacts the final ranking of keyphrases used in the baseline.[2] The baseline uses the "Oral & Dental Care" category for the Toothpaste product category. This value is followed from the baseline approach. However, for the Perfume and Dog Food product categories, "Perfumes & Fragrances" and "Dogs" are used respectively. This is done as they are similar to the corresponding product categories in the lack of recommended values for these product categories.

[1] These are the same value ranges reported in Chapter 4 i.e. Table 4.2.

[2] https://github.com/pat310/google-trends-api/wiki/Google-Trends-Categories - last accessed 07/06/2024

Appendix R: Seen & Unseen Categories F1 Score Distribution

To further visualize the fact that the F1 scores for the Seen and Unseen Testing Categories for the Multiple Category approach don't differ much from each other, a kernel density estimate plot of these scores is plotted in Figure R.1 using the Python library seaborn.[1] This plot is used as it's more visually intuitive than a histogram [422, 423]. For the *bandwidth* (a key parameter used to smooth the plot produced by kernel density estimation [423]), the default value in seaborn is used i.e. *bandwidth*=1. This is done across all the 10 runs for each category, therefore recording 80 F1 scores for the Unseen Testing Categories (i.e. 8 categories multiplied by 10 runs) and 70 F1 scores for the Seen Testing Categories (i.e. 7 categories multiplied by 10 runs). The plot shows that although the results from the Seen Testing Categories are higher than the Unseen Testing Categories, they don't deviate much from each other. Hence, it can be concluded that the MTL approach can still predict future customer needs on a category it has not seen during training with relatively similar performance to ones it has seen during training.

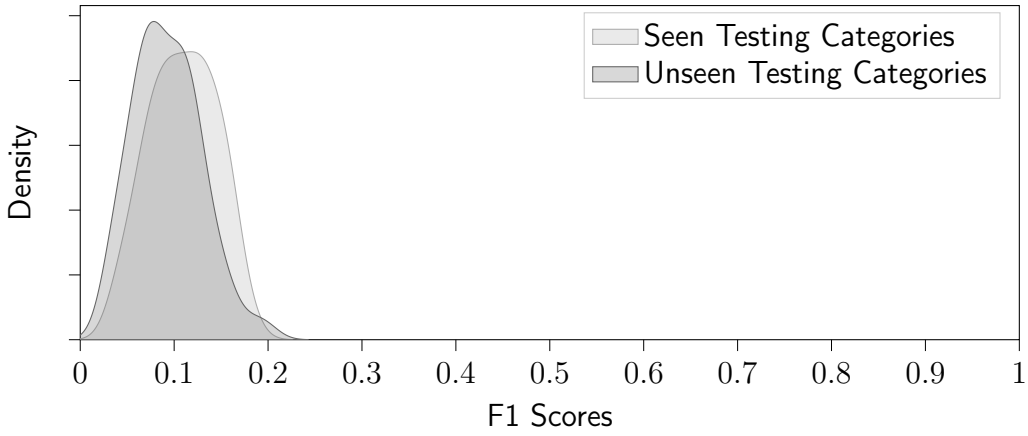

Figure R.1. Seen vs Unseen Testing Category F1 result distribution - the seen and unseen categories predict with relatively similar accuracy. The x-axis shows the F1 scores while the y-axis shows the density of the distribution.

[1] https://seaborn.pydata.org/generated/seaborn.kdeplot.html - last accessed 07/06/2024

Appendix S: Seen & Unseen Categories Mean Precision and Recall Score Distribution

To further visualize the fact that the List Mean Precision and List Recall scores for the Seen and Unseen Testing Categories for the Multiple Category approach don't differ much from each other, multiple kernel density estimate plots of these scores across each value of K (i.e. number of submitted keyphrases) are shown in Figure S.1.[1] For each plot this is done across all 10 runs for each category. The plots show the results don't deviate much from each other. Hence, it can be concluded that the MTL approach can still predict future customer needs on a category it has not seen during training with relatively similar performance to ones it has seen during training.

[1] This is shown instead of a histogram for the same reasons as detailed in Appendix R (more visually intuitive). The same library as in Appendix R is also used to generate the plots i.e seaborn.

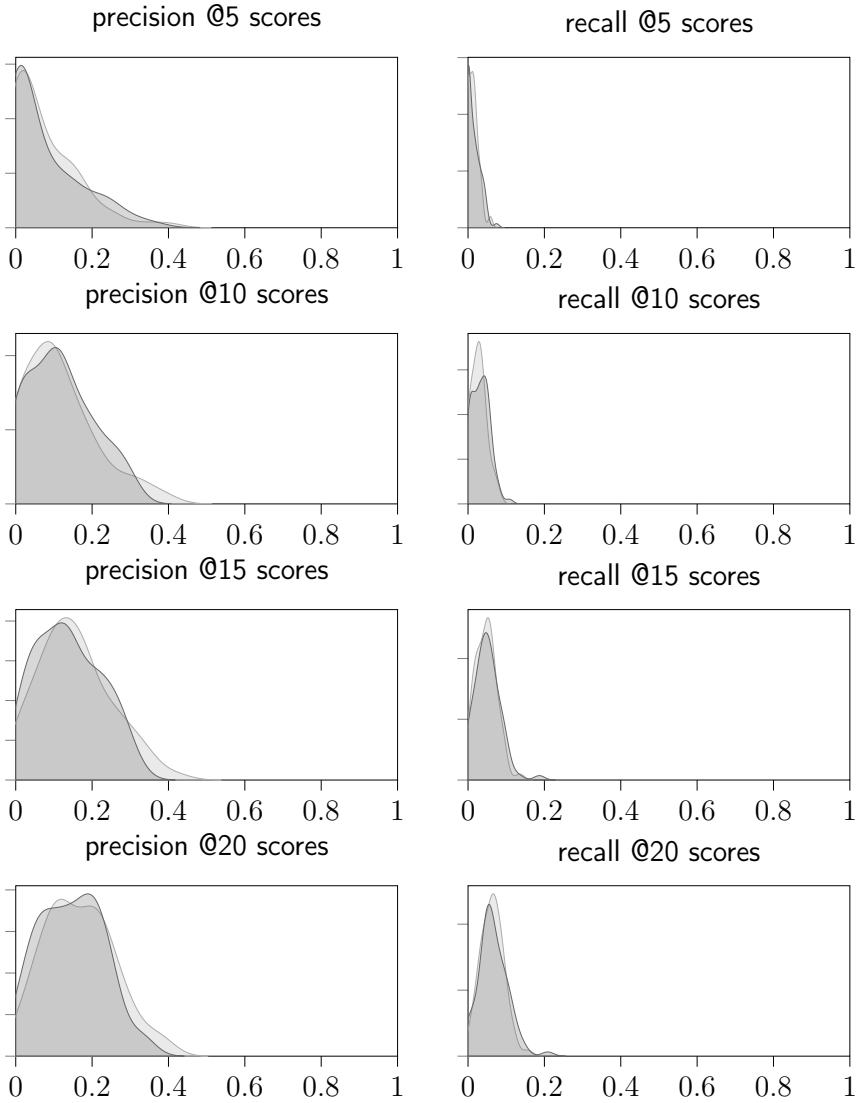

Figure S.1. Multiple Category approach for the Seen/Unseen results of List Evaluation. For each plot, the x-axis shows the results while the y-axis shows the density of the values. For each plot the unseen distribution is in the darker grey.

Appendix T: **Generate List Guidelines**

The following lists the guidelines on what participants should do when generating lists. Participants are asked to: *"generate a ranked list of the top 10 phrases which most represent customer needs which will trend in the marketplace 1-3 years in the future".* This statement can be quite loaded/confusing, therefore the following guidelines below give more detail on what this statement is looking for.

T.1 What is a "phrase"?

A phrase is a small group of words standing together as a unit. For the purposes of this work participants should mainly keep the total number of words used low - around 1 to 3 words e.g. "apple", "dry skin", "anti aging" etc.

T.2 What is a "customer need"?

A "customer need" is a reason/motivation for why someone buys a product. For the purposes of this work it is split into two main categories: 1) benefiting descriptions; 2) features of products.

A benefiting description is a phrase which stands alone as a benefiting specification of a product. They can be nouns (e.g. gingivitis, manicure), adjectives (e.g. microwavable), verbs (e.g. whiten), participle verbs (e.g. baking, whitening, fried) or adverbs (e.g. fresh). Such phrases could include:

- benefits a user gets from using the product e.g. "antiseptic" for soap products, "microwave- able" for popcorn products, "vegan" for ice-cream products or "frying" for olive oil products
- benefits a user overcomes from using the product e.g. "dry hands" for hand cream products or "gingivitis" for toothpaste products
- benefits a user can perform as an activity or treatment with the product e.g. "manicure" for nail polish products

A feature of a product is a descriptive phrase which the user can touch, taste, hear or smell which has benefits associated with its inclusion in the product. These phrases are mainly

nouns as they are actual features of the product (i.e. they are real things). Such phrases could include:

- ingredients of products such as "honey" in cereal products or "jasmine" in tea products
- flavours/scents of products such as "raspberry" for shampoo products or "mango" for beer products
- tangible features of the product such as "tongue scraper" for toothbrush products

T.3 What a "customer need" is not

Below are some additional guidelines on what a customer need is not. Do not include these items in the final predicted list. Some of these guidelines are obvious (e.g. a customer need is not a brand), while others are less obvious however are noted for experimental reasons.

- Deals that come as part of the product being sold e.g. "buy one get one free"
- Names of companies (e.g. Oral-B), brands (e.g. Centrum) , social media (e.g. Twitter), stamps of approvals (e.g. EcoCertificate), logos (e.g. EU Green Leaf logo)
- Specific targeted groups of people who promote (e.g. dentists, doctors, celebrities) or actually use the product (e.g. adults, elderly people)
- Anything to do with the packaging of the product such as size of the product (e.g. 100ml,twin pack)
- Region names e.g. "Columbia Valley" and "Bordeaux" for wine products or "Irish" or "Italian" for coffee products
- Any marketing "power word" e.g. premium, exquisite, explosive, advanced, effective, hurry, rare, limited, proven, unique etc.
- Anything about the pricing of the product e.g. bargain, affordable, low cost, $2, £12.50 etc.

T.4 What is a "customer need which will trend in the marketplace 1-3 years in the future"?

Generally a "future customer need" is a need which will be of importance in the future e.g. "unsalted" for Popcorn. Here participants are asked to predict what needs they think will be popular in the future given their prior knowledge. For the purposes of this exercise, participants are asked to try to restrict the scope of this future time period to between

1-3 years in the future. It is understood that this is a difficult task, however, a general lower and upper limit is required to provide scope for what needs should and shouldn't be included. As stated, anything outside of this 1-3 year time period is not to be included in the generated list e.g. the need of "reverse rotting" for Toothpaste products which may be on the market in 20 years time.

T.5 Overall

Participants are asked to *"generate a list of the top 10 phrases which most represent customer needs which will trend in the marketplace 1-3 years in the future"*. Here are some examples of lists which may be generated:

Popcorn Future Customer Needs: 1: truffle oil; 2) pre popped; 3) vegan; 4) low calorie; 5) chocolate; 6) nut free; 7) organic; 8) dried fruit; 9) spicy; and 10) kosher.

Eyeliner Future Customer Needs: 1) glue free; 2) long lasting; 3) waterproof; 4) anti-ageing; 5) uv protection; 6) scented; 7) charcoal; 8) metallic; 9) hypoallergenic; and 10) quick drying.

As seen in the lists, the phrases are:

- Between 1-3 words long
- Conform to either a "benefiting description" or a "feature of a product"
- Are not unrealistic future customer needs in 1-3 years time

T.6 Mistakes Made in Pilot Study Exercise by Participants

- After your list is generated please reread the guidelines in order to ensure the your list adheres to it i.e. quality control to ensure there is no conflicts
- Make sure to generate keyphrases for the product at hand e.g. if generating for Toothpaste do not include "tongue scraper" as a keyphrase as this is a need of Toothbrush products

T.7 Additional Note(s)

- Participants should not use any additional resources when formulating the lists of keyphrases e.g. Google. Lists should be made based on their current knowledge.
- It is recommended that participants read these guidelines a minimum of 2 times.

Appendix U: **Compare Output Guidelines**

The following provides guidelines for what constitutes a match between two phrases i.e. your phrase and an algorithm generated phrase.

- An exact match - your output and algorithm output both have the phrase "cinnamon"
- Synonym match - your output and algorithm output both have the phrases with the same meaning. This includes phrases with the almost exact same meaning (e.g. "baking soda" and "sodium bicarbonate") or even a similar meaning (e.g. "cocoa" and "chocolate")
- A match baring an insignificant word in the phrase - your output and the algorithm output match baring a word which doesn't provide much information e.g. if the phrases are "paraben" and "free from paraben" then the phrases match because they contain the main word i.e. "paraben"

When comparing your generated output to the algorithm-generated output, report the location of the occurrence of the first match in the algorithm-generated list. Do not use Control-F when searching for matching keyphrases - each keyphrases should be gone through with thought to see whether it is a synonym or a match baring an insignificant word.

Appendix V: Questions Asked To Participants

Table V.1 shows all the questions asked to participants during the questionnaire evaluation. In total, 4 Novelty Questions, 2 List Changed Questions, 1 Unuseful Question and 3 System-Useful Questions were asked. All the questions requiring a raw number ask for an estimate of the question preceding them (e.g. the second question asks for an estimate of the 1st question in Table V.1).

Table V.1. All questions asked to participants

Question Number	Question Family	Question	Potential Answers
1	Novelty	This algorithm generated list contains _____ keyphrases which weren't considered when my list was made	i) Many ii) Hardly Any iii) None
2	Novelty	From the answer above, estimate the number of phrases which weren't considered when finalising your list (raw number e.g. 60). Answers are rounded to the nearest 10.	Raw Number (0-100)
3	Novelty	From the new keyphrases there are _____ which would be considered for further investigation	i) Many ii) Hardly Any iii) None
4	Novelty	From the answer above, estimate the number of phrases which would be considered when finalising your list (raw number e.g. 60). Answers are rounded to the nearest 10.	Raw Number (0-100)
5	List Changed	My generated list would now change significantly having read the algorithm generated list	i) Strongly Agree ii) Agree iii) Neutral iv) Disagree v) Strongly Disagre
6	List Changed	From the answer above, estimate the number of phrases which are not useful (raw number e.g. 60). Answers are rounded to the nearest 10.	Raw Number (0-100)
7	Unuseful	There are _____ keyphrases in the algorithm generated list which are definitely not useful	i) Many ii) Hardly Any iii) None

Table V.1 Continued: All questions asked to participants

8	System-Useful	The algorithm generated keyphrases would be useful in making my list	i) Strongly Agree ii) Agree iii) Neutral iv) Disagree v) Strongly Disagree
9	System-Useful	I would have preferred to see the algorithm generated list before attempting to generate my list	i) Strongly Agree ii) Agree iii) Neutral iv) Disagree v) Strongly Disagree
10	System-Useful	I would anticipate that having a generated list of keyphrases to assist with generating our own future lists would be helpful to the product development process	i) Strongly Agree ii) Agree iii) Neutral iv) Disagree v) Strongly Disagree

Appendix W: All List Comparison Evaluation Answers

Table W.1 records all of the general locations of the participant's phrases in the algorithm-generated list.

Table W.1. Locations of participant's phrases in algorithm

Person #	Department	Expert Status	General Location	Person #	Department	Expert Status	General Location
1	Dog Food	Expert	not found	4	Dog Food	Non-Expert	61-65
1	Dog Food	Expert	not found	4	Dog Food	Non-Expert	not found
1	Dog Food	Expert	not found	4	Dog Food	Non-Expert	not found
1	Dog Food	Expert	not found	4	Dog Food	Non-Expert	46-50
1	Dog Food	Expert	not found	4	Dog Food	Non-Expert	not found
1	Dog Food	Expert	not found	4	Dog Food	Non-Expert	not found
1	Dog Food	Expert	91-95	4	Dog Food	Non-Expert	not found
1	Dog Food	Expert	not found	5	Shampoo	Expert	86-90
1	Dog Food	Expert	not found	5	Shampoo	Expert	not found
1	Dog Food	Expert	1-5	5	Shampoo	Expert	not found
2	Dog Food	Expert	31-35	5	Shampoo	Expert	not found
2	Dog Food	Expert	not found	5	Shampoo	Expert	not found
2	Dog Food	Expert	91-95	5	Shampoo	Expert	not found
2	Dog Food	Expert	not found	5	Shampoo	Expert	91-95
2	Dog Food	Expert	not found	5	Shampoo	Expert	1-5
2	Dog Food	Expert	not found	5	Shampoo	Expert	31-35
2	Dog Food	Expert	not found	5	Shampoo	Expert	56-60
2	Dog Food	Expert	not found	6	Shampoo	Expert	71-75
2	Dog Food	Expert	not found	6	Shampoo	Expert	not found
2	Dog Food	Expert	not found	6	Shampoo	Expert	66-70
3	Dog Food	Expert	not found	6	Shampoo	Expert	not found
3	Dog Food	Expert	not found	6	Shampoo	Expert	11-15
3	Dog Food	Expert	11-15	6	Shampoo	Expert	not found
3	Dog Food	Expert	not found	6	Shampoo	Expert	6-10
3	Dog Food	Expert	not found	6	Shampoo	Expert	1-5
3	Dog Food	Expert	81-85	6	Shampoo	Expert	not found
3	Dog Food	Expert	11-15	6	Shampoo	Expert	not found
3	Dog Food	Expert	21-25	7	Shampoo	Expert	not found
3	Dog Food	Expert	not found	7	Shampoo	Expert	not found
3	Dog Food	Expert	not found	7	Shampoo	Expert	not found
4	Dog Food	Non-Expert	31-35	7	Shampoo	Expert	not found

Table W.1 Continued: Locations of participant's phrases in algorithm

4	Dog Food	Non-Expert	91-95	**7**	Shampoo	Expert	6-10
4	Dog Food	Non-Expert	not found	**7**	Shampoo	Expert	91-95
7	Shampoo	Expert	not found	**10**	Toothpaste	Expert	66-70
7	Shampoo	Expert	not found	**10**	Toothpaste	Expert	41-45
7	Shampoo	Expert	11-15	**10**	Toothpaste	Expert	not found
7	Shampoo	Expert	not found	**10**	Toothpaste	Expert	not found
8	Shampoo	Non-Expert	26-30	**10**	Toothpaste	Expert	not found
8	Shampoo	Non-Expert	not found	**10**	Toothpaste	Expert	not found
8	Shampoo	Non-Expert	not found	**10**	Toothpaste	Expert	not found
8	Shampoo	Non-Expert	not found	**11**	Toothpaste	Expert	not found
8	Shampoo	Non-Expert	21-25	**11**	Toothpaste	Expert	not found
8	Shampoo	Non-Expert	11-15	**11**	Toothpaste	Expert	not found
8	Shampoo	Non-Expert	not found	**11**	Toothpaste	Expert	66-70
8	Shampoo	Non-Expert	96-100	**11**	Toothpaste	Expert	not found
8	Shampoo	Non-Expert	56-60	**11**	Toothpaste	Expert	not found
8	Shampoo	Non-Expert	86-90	**11**	Toothpaste	Expert	not found
9	Toothpaste	Expert	not found	**11**	Toothpaste	Expert	not found
9	Toothpaste	Expert	96-100	**11**	Toothpaste	Expert	96-100
9	Toothpaste	Expert	not found	**11**	Toothpaste	Expert	not found
9	Toothpaste	Expert	not found	**12**	Toothpaste	Non-Expert	91-95
9	Toothpaste	Expert	96-100	**12**	Toothpaste	Non-Expert	not found
9	Toothpaste	Expert	21-25	**12**	Toothpaste	Non-Expert	not found
9	Toothpaste	Expert	not found	**12**	Toothpaste	Non-Expert	not found
9	Toothpaste	Expert	not found	**12**	Toothpaste	Non-Expert	not found
9	Toothpaste	Expert	not found	**12**	Toothpaste	Non-Expert	71-75
9	Toothpaste	Expert	not found	**12**	Toothpaste	Non-Expert	31-35
10	Toothpaste	Expert	not found	**12**	Toothpaste	Non-Expert	not found
10	Toothpaste	Expert	not found	**12**	Toothpaste	Non-Expert	not found
10	Toothpaste	Expert	not found	**12**	Toothpaste	Non-Expert	11-15

Appendix X: **All Questionnaire Answers**

Table X.1 records all of the participant's answers to the questionnaire. The corresponding "Question" columns in the table map to the "Question Number" in Table V.1 (Appendix V). It is of note that the "Person #" in the table is the same as the one in Table W.1 (Appendix W). This may be useful for an additional analysis to discover if there is a relationship between positive answers in the questionnaire evaluation and a high overlap in the list comparison evaluation.

Table X.1. All answers recorded by participants

Person #	Depart-ment	Expert Status	Question 1	Question 2	Question 3	Question 4
1	Dog Food	Expert	Many	80	Hardly Any	8
2	Dog Food	Expert	Many	50	Hardly Any	20
3	Dog Food	Expert	Many	75	Hardly Any	10
4	Dog Food	Non-Expert	Many	70	Hardly Any	50
5	Shampoo	Expert	Many	50	Many	20
6	Shampoo	Expert	Many	60	Hardly Any	5
7	Shampoo	Expert	Many	50	Hardly Any	20
8	Shampoo	Non-Expert	Hardly Any	90	Hardly Any	10
9	Toothpaste	Expert	Many	70	Many	10
10	Toothpaste	Expert	Many	80	Many	10
11	Toothpaste	Expert	Many	80	Hardly Any	10
12	Toothpaste	Non-Expert	Many	30	Many	70

All the "Question" columns (e.g. Question 1) can be seen in the "Question Number" column in Table V.1

Table X.1 Continued: All answers recorded by participants

Person #	Question 5	Question 6	Question 7	Question 8	Question 9	Question 10
1	Disagree	Many	80	Agree	Agree	Neutral
2	Neutral	Hardly Any	20	Disagree	Agree	Agree
3	Neutral	Many	30	Agree	Agree	Agree
4	Agree	Hardly Any	10	Strongly Agree	Strongly Agree	Strongly Agree
5	Agree	Hardly Any	10	Agree	Agree	Agree
6	Agree	Many	50	Neutral	Neutral	Neutral
7	Agree	Many	70	Neutral	Strongly Disagree	Disagree
8	Strongly Agree	Hardly Any	5	Strongly Agree	Strongly Agree	Strongly Agree
9	Agree	Many	50	Agree	Agree	Strongly Agree
10	Strongly Disagree	Many	90	Disagree	Neutral	Disagree
11	Neutral	Many	80	Agree	Agree	Disagree
12	Agree	Hardly Any	20	Agree	Agree	Agree

All the "Question" columns (e.g. Question 5) can be seen in the "Question Number" column in Table V.1

Appendix Y: Additional Novelty Question Responses

Figures Y.1 and Y.2 show the follow-up questions asked to participants after the questions in Figures 6.5 and 6.6. The response in Figure Y.1 is highly consistent with 6.5. Similarly, the response in Figure Y.2 is consistent with Figure 6.6.

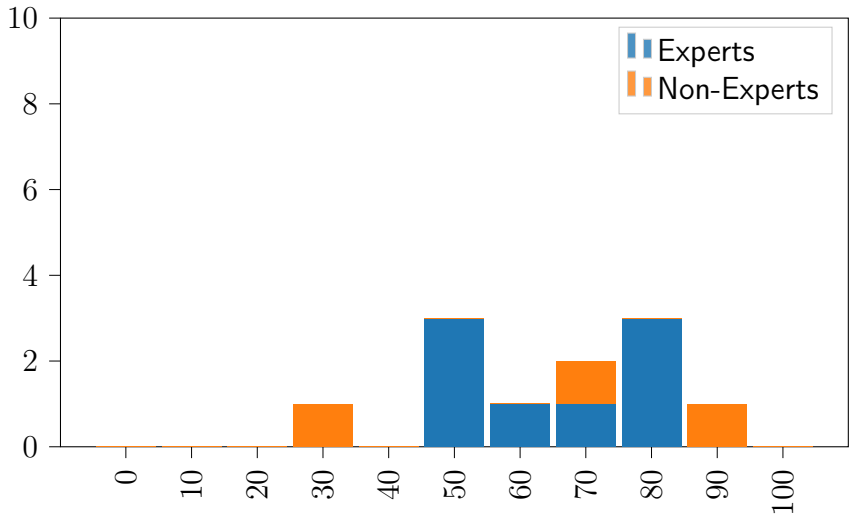

Figure Y.1. From the answer above (i.e. Figure 6.5), estimate the number of phrases which weren't considered when finalising your list (raw number e.g. 60). Answers are rounded to the nearest 10.

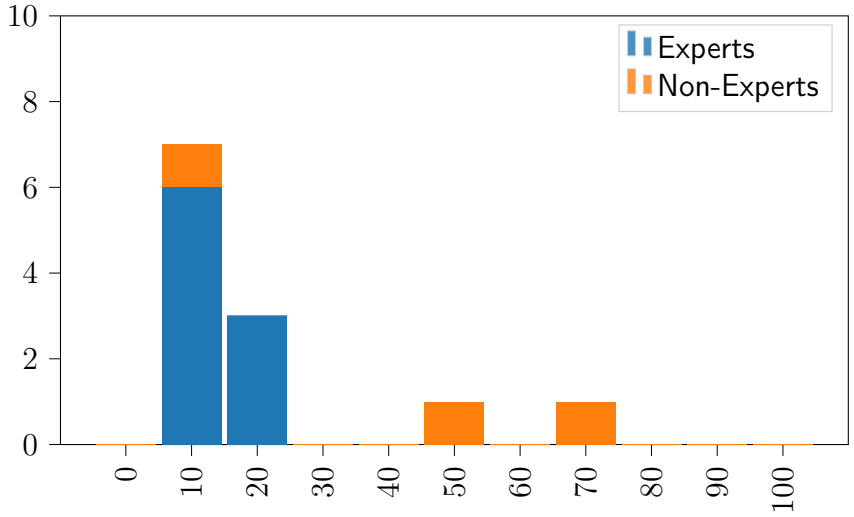

Figure Y.2. From the answer above (i.e. Figure 6.6), estimate the number of phrases which would be considered when finalising your list (raw number e.g. 60). Answers are rounded to the nearest 10.

Appendix Z: Additional Unuseful Question Responses

Figure Z.1 shows the follow up question asked to participants after the question in Figure 6.8 i.e. "there are _____ keyphrases in the algorithm generated list which are definitely not useful". The response in the figure is somewhat consistent with Figure 6.8.

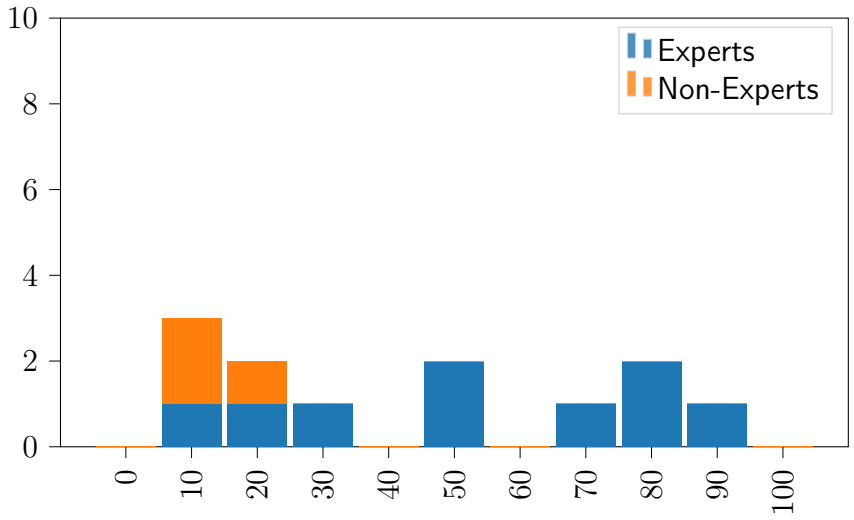

Figure Z.1. From the answer above (i.e. Figure 6.8), estimate the number of phrases which are not useful (raw number e.g. 60). Answers are rounded to the nearest 10.