Sentimental Product Recommendation

Ruihai Dong†, Michael P. O'Mahony‡, Markus Schaal†, Kevin McCarthy†, Barry Smyth‡
†CLARITY: Centre for Sensor Web Technologies
‡INSIGHT: Centre for Data Analytics
School of Computer Science and Informatics
University College Dublin, Dublin, Ireland
firstname.lastname@ucd.ie

ABSTRACT

This paper describes a novel approach to product recommendation that is based on *opinionated* product descriptions that are automatically mined from user-generated product reviews. We present a recommendation ranking strategy that combines similarity and sentiment to suggest products that are similar but superior to a query product according to the opinion of reviewers. We demonstrate the benefits of this approach across a variety of Amazon product domains.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information filtering; H.3.5 [**Online Information Services**]: Web-based services

Keywords

User-generated Reviews; Opinion Mining; Sentiment-based Product Recommendation

1. INTRODUCTION

Consider the *13" Retina MacBook Pro*. At the time of writing its *product features*, as listed by Amazon, cover technical details such as *screen-size*, *RAM*, *processor speed*, and *price*. These are the type of features that one might expect to find in a conventional content-based recommender [9]. Often, such features can be difficult to locate and can be technical in nature, thereby limiting recommendation opportunities and making it difficult for casual shoppers to judge the relevance of suggestions. However, the *MacBook Pro* has more than 70 reviews which encode valuable insights into a great many of its features; from its *"beautiful design"* and *"great video editing"* capabilities to its *"high price"*. These features capture more detail than a handful of technical (catalog) features. They also encode the *opinions* of real users and thus provide a basis for product comparisons.

We consider the following in this paper: can such features be used as the basis for a new type of *experiential* product recommendation, which is based on genuine user experiences? Are these features sufficiently rich to be a viable alternative to more conventional product descriptions based on meta-data or catalog features? And what type of recommendation strategies might be used?

There have been a number of efforts focused on extracting feature-based product descriptions from reviews. The work in [10] is representative in this regard and describes the use of shallow NLP techniques for explicit feature extraction and sentiment analysis; see also [6]. The features extracted, and the techniques used, are similar to those presented in this paper, although in the case of the former there was a focus on the extraction of meronomic and taxonomic features to describe the parts and properties of a product.

Zhang et al. [11] analyze the sentiment of comparative and subjective sentences in reviews on a per-feature basis to create a semi-order of products, but do not consider the recommendation task with respect to a query product. The work in [5] is relevant in that it uses user-generated micro-reviews as the basis for a text-based content recommender, and recently work in [2] has also tried to exploit user-generated content in similar ways. In [1], a manually defined ontology is used to convert opinions extracted from reviews into a structured form. This ontology, which captures both the reviewer's skill level and experience with the product, is then leveraged for recommendation.

The work in this paper uses existing techniques [4, 6–8] to automatically extract rich, opinionated product case representations from user-generated reviews and extends our earlier work [3] by introducing a new approach to sentiment-based recommendation and a hybrid approach for combining similarity and sentiment during recommendation.

2. MINING PRODUCT EXPERIENCES

The reviews for each product, P, are converted into a rich, feature-based, experiential case in 3 steps as follows (Figure 1). See [3] for more details.

Extracting Review Features. Shallow NLP and statistical methods are used to mine features from reviews [6, 7]. We consider *bi-gram* features which conform to one of two part-of-speech co-location patterns: a noun preceded by an adjective (AN) or by a noun (NN). *Single-noun* features which frequently co-occur ($\geq$ 70% of the time) with sentiment words in the same sentence are also considered [6].

Evaluating Feature Sentiment. We use a version of *opinion pattern mining* to evaluate feature sentiment [8].

For a given feature F_i in sentence S_j of review R_k, we identify the closest sentiment word w_{min} to F_i in S_j; F_i is labeled as *neutral* if no sentiment words are present (sentiment words are those contained in the sentiment lexicon [6]). Next we extract the *opinion pattern*: the part-of-speech tags for w_{min}, F_i and any words that occur between them. After a pass over all features, the frequency of occurrence of all patterns is noted. For valid patterns (those which occur more than once) we assign sentiment to F_i based on that of w_{min} in the sentiment lexicon; sentiment is reversed if S_j contains a negation term within a 4-word distance of w_{min}. Features associated with invalid patterns are labeled as *neutral*.

Generating Experiential Product Cases. For each product P we now have a set of features $F(P) = \{F_1, ..., F_m\}$ extracted from the reviews of P $\big(Reviews(P)\big)$ and each feature F_i has an associated set of *positive, negative, or neutral* sentiment labels $(L_1, L_2, \dots)$. Features which are mentioned in $\geq 10\%$ of reviews for that product are only considered and overall sentiment (Equation 1) and popularity (Equation 2) scores are calculated; $Pos(F_i, P)$ (resp. $Neg(F_i, P)$, $Neut(F_i, P)\big)$ denotes the number of positive (resp. negative, neutral) sentiment labels for feature F_i. The product case, $Case(P)$, is then given by Equation 3.

$$Sent(F_i, P) = \frac{Pos(F_i, P) - Neg(F_i, P)}{Pos(F_i, P) + Neg(F_i, P) + Neut(F_i, P)} \quad (1)$$

$$Pop(F_i, P) = \frac{|\{R_k \in Reviews(P) : F_i \in R_k\}|}{|Reviews(P)|} \quad (2)$$

$$Case(P) = \{[F_i, Sent(F_i, P), Pop(F_i, P)] : F_i \in F(P)\} \quad (3)$$

3. RECOMMENDING PRODUCTS

The above case representation leads to a content-based recommendation approach based on feature similarity to a query product. However, the availability of feature sentiment suggests another approach in which products that offer *better* quality features compared to the query product are recommended. These techniques are described below.

Similarity-Based Recommendation. Each product case is represented as a vector of features, where feature *values* represent their popularity in reviews (Equation 2) as a proxy for their importance. The cosine similarity between query product, Q, and candidate recommendation, C, is given by:

$$Sim(Q, C) = \frac{\sum_{F_i \epsilon F(Q) \cup F(C)} Pop(F_i, Q) \times Pop(F_i, C)}{\sqrt{\sum_{F_i \epsilon F(Q)} Pop(F_i, Q)^2} \sqrt{\sum_{F_i \epsilon F(C)} Pop(F_i, C)^2}} \quad (4)$$

Using this approach, a set of top n recommendations are generated, ranked according to their query product similarity [9].

Sentiment-Enhanced Recommendation. Rather than recommend products using *similarity* alone, feature sentiment can also be used to seek products with *better* sentiment than the query product. Equation 5 computes a score for feature F_i between query product Q and recommendation candidate C; a positive (resp. negative) score means that C has higher (resp. lower) sentiment for F_i compared to Q.

$$better(F_i, Q, C) = \frac{Sent(F_i, C) - Sent(F_i, Q)}{2} \quad (5)$$

Equation 6 computes an average better score at the product level across the *shared* features between Q and C. However, this approach ignores any *residual features* that are unique to Q or C. Thus, Equation 7 computes an average better score across the *union* of features in Q and C; non-shared features are assigned a neutral sentiment score of 0.

$$B1(Q, C) = \frac{\sum_{F_i \in F(Q) \cap F(C)} better(F_i, Q, C)}{|F(Q) \cap F(C)|} \quad (6)$$

$$B2(Q, C) = \frac{\sum_{F_i \in F(Q) \cup F(C)} better(F_i, Q, C)}{|F(Q) \cup F(C)|} \quad (7)$$

Combining Similarity and Sentiment. The sentiment-based approaches above prioritise products that enjoy more positive reviews across a range of features relative to the query product. However, these recommendations may not necessarily be very similar to the query product. Thus, Equation 8 ranks recommendations based on their combined (controlled by w) similarity and sentiment with respect to Q; $B(Q, C)$ denotes $B1$ or $B2$, normalised to [0, 1].

$$Score(Q, C) = (1 - w)\ Sim(Q, C) + w\big(B(Q, C) + 1\big)/2 \quad (8)$$

4. EVALUATION

The above approaches are evaluated using data extracted from Amazon.com during October 2012. We considered 6 product domains; here we present representative results for 3 domains (Table 1). For each product with ≥ 10 reviews, we extracted review texts and helpfulness information, and the top n recommendations as suggested by Amazon.

Domain	#Reviews	#Products	$\#Features$ μ (σ)	Sims. μ (σ)
Tablets	17,936	166	26 (10)	0.6 (0.1)
Phones	14,860	257	9 (5)	0.5 (0.2)
GPS	12,115	119	24 (11)	0.6 (0.2)

Table 1: Dataset statistics.

4.1 Mining Rich Product Descriptions

The success of our approach depends on its ability to translate user-generated reviews into useful product cases. Table 1 shows the mean and standard deviation of the number of features that are extracted for each domain. On average, 9-26 features are extracted per product case, indicating that reasonably feature-rich cases are generated. Table 1 (last column) also shows the mean and standard deviation of the pairwise product cosine similarities. Again the results bode well because they show a relatively wide range of similarity values; very narrow ranges would suggest limitations in the expressiveness of extracted product representations.

4.2 Recommendation Performance

A standard *leave-one-out* approach is used in our evaluation, comparing our recommendations for each product to those produced by Amazon. Thus, for each product (referred to as the *query product*, Q) in a given domain, we generate a set of top 5 recommendations using Equation 8, varying w from 0 to 1 in steps of 0.1. This produces 22 recommendation lists for each Q, 11 each for $B1$ and $B2$, which we compare to Amazon's own recommendations for Q.

4.2.1 Ratings Benefit

We use Amazon's overall product ratings as an independent measure of product quality. The *ratings benefit* metric compares two sets of recommendations based on their ratings (Equation 9), where a ratings benefit of 0.1 means that our recommendations R enjoy an average rating score that is 10% higher that those produced by Amazon (A).

$$Benefit(R, A) = \frac{\overline{Rating(R)} - \overline{Rating(A)}}{\overline{Rating(A)}} \quad (9)$$

We also compute the *query product similarity*, the average similarity based on mined feature representations between our recommendations and the query product. This allows us to evaluate whether our techniques produce recommendations that are related to the query product and also provides a basis for comparison to Amazon's recommendations.

For each domain, Figure 3(a–c) shows $B1$ and $B2$ results for top 5 recommendations. Ratings benefit scores (left y-axis, dashed lines) for $B1$ (circles) and $B2$ (squares) against w (x-axis), along with the corresponding query product similarity values (right y-axis, solid lines), are shown. The average similarity between the query product and the Amazon recommendations is also shown, which is independent of w and so appears as a solid horizontal line in each graph.

4.2.2 Contrasting Sentiment and Similarity

At $w = 0$, Equation 8 is equivalent to a pure similarity-based approach to recommendation using cosine, because sentiment is not contributing to the overall recommendation score. For this configuration there is little or no ratings benefit; the recommendations produced have very similar average ratings to those produced by Amazon. However, the recommendations that are produced are more similar to the query product, in terms of the features mentioned in reviews, than Amazon's own recommendations. For example, in the *Phones* domain (Figure 3(b)) at $w = 0$, recommendations based on cosine have a query product similarity of 0.8 compared to 0.6 for Amazon's recommendations.

At $w = 1$, where recommendations are based solely on sentiment, we see a range of maximum positive ratings benefits (from 0.18 to 0.23) across all 3 product domains. $B2$ outperforms $B1$, except for GPS, indicating that the sentiment associated with residual (non-shared) features is important, at least for two of the three domains considered. Consider again the *Phones* domain (Figure 3(b)). At $w = 1$, we see a ratings benefit of 0.11 and 0.21 for $B1$ and $B2$, respectively. Thus, products recommended by $B2$ enjoy ratings that are 21% higher than Amazon's recommendations, an increase of almost one point on average for Amazon's 5-point scale.

However, these ratings benefits are offset by a drop in query product similarity. At $w = 1$, query product similarity falls below that of the Amazon recommendations. Thus, a tradeoff exists between ratings benefits and query product similarity.

4.2.3 Combining Similarity and Sentiment

The relative contribution of similarity and sentiment is governed by w (Equation 8). As w increases a gradual increase in ratings benefit for $B1$ and $B2$ is seen, especially at larger w, with $B2$ outperforming $B1$ except for GPS. The slope of the ratings benefit curves and the maximum benefit achieved is influenced by the ratings distribution in each domain. For example, $Phones$ and $Tablets$ have ratings distributions with relatively low means and high standard deviations. Thus, more opportunities for improved ratings exist and, indeed, the highest ratings benefits are seen for these domains (above 0.2 at $w = 1$ for $B2$).

Regarding query product similarity, there is little change for $w < 0.7$. But for $w > 0.7$ there is a reduction as sentiment tends to dominate during recommendation ranking. This query product similarity profile is remarkably consistent across all product domains and in all cases $B2$ better preserves query product similarity compared to $B1$.

To better understand the relative performance of $B1$ and $B2$ with respect to the Amazon baseline as w varies, we need a point of reference for the purpose of a *like-for-like* comparison. To do this we compare our techniques by fixing w at the point at which the query product similarity curve intersects with the Amazon query product similarity level and then reading the corresponding ratings benefits for $B1$ and $B2$. This is a useful reference point because it allows us to look at the ratings benefit offered by $B1$ and $B2$ when delivering recommendations that have the same query product similarity as the baseline Amazon recommendations.

Figure 2 shows these ratings benefits and corresponding w values for $B1$ and $B2$. The results clarify the positive ratings benefits that are achieved using sentiment-based recommendation without compromising query product similarity. For $Tablets$ and $Phones$ there are very significant ratings benefits, especially for $B2$ (resp. 15% and 21%). As stated above, $B1$ outperforms $B2$ for GPS, but in a relatively minor way, suggesting that the sentiment associated with residual features is not playing a significant role in this domain.

Finally, note the consistency of the w values at which the query product similarity of the sentiment-based recommendations matches that of Amazon. For each domain, $w \approx 0.9$ (for $B2$) delivers recommendations that balance query product similarity with significant ratings benefits; whether this value of w applies in general we leave to future work.

5. CONCLUSIONS

The objective of this work has been twofold: (1) to convert unstructured reviews into rich product descriptions and (2) to use these product descriptions in a recommender system that combines similarity and sentiment. Our results show clear benefits in terms of recommendation quality compared to Amazon's own recommendations. In this work, we have considered one particular approach to similarity: a cosine metric calculated over the frequency of occurrence of extracted product features. A question arises as to whether this approach indeed reflects an authentic notion of product similarity as judged by human assessment. A detailed exploration of this matter is left to future work; however, we note that preliminary assessments attest to the validity of our approach.

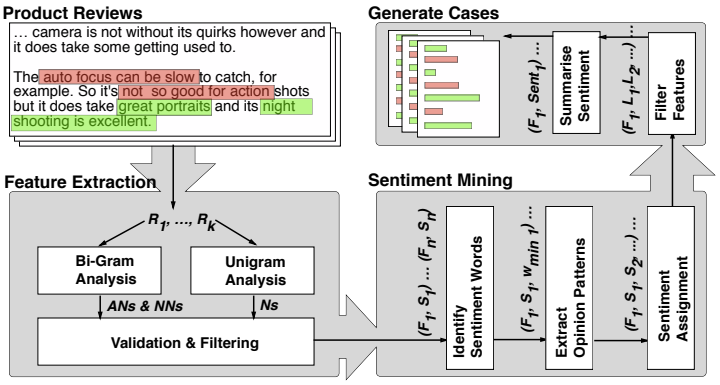

Figure 1: Extracting product cases from user reviews.

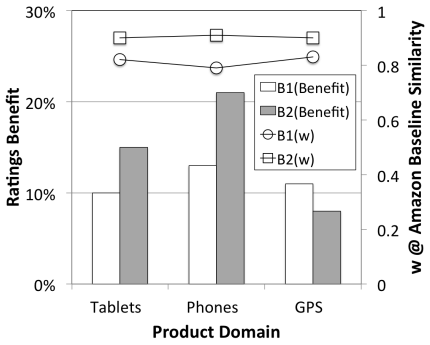

Figure 2: Ratings benefits at Amazon baseline query product similarity.

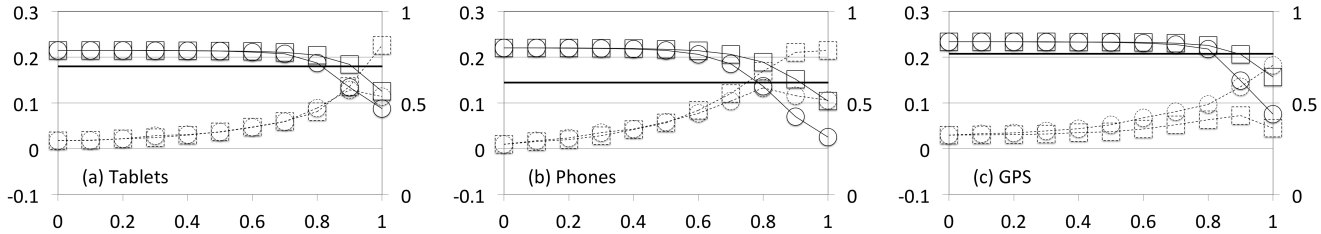

Figure 3: Ratings benefit (left y-axis, dashed lines) and query similarity (right y-axis, solid lines) versus w (x-axis); $B1$ and $B2$ are shown as circles and squares and Amazon query similarity as a solid horizontal line.

6. ACKNOWLEDGEMENTS

This work is supported by Science Foundation Ireland under grant 07/CE/I1147. The INSIGHT Centre for Data Analytics is supported by Science Foundation Ireland under Grant Number SFI/12/RC/2289.

7. REFERENCES

[1] S. Aciar, D. Zhang, S. Simoff, and J. Debenham. Informed recommender: Basing recommendations on consumer product reviews. *Intelligent Systems, IEEE*, 22(3):39–47, 2007.

[2] G. De Francisci Morales, A. Gionis, and C. Lucchese. From chatter to headlines: Harnessing the real-time web for personalized news recommendation. In *Proceedings of the fifth ACM International Conference on Web Search and Data Mining*, WSDM '12, pages 153–162, New York, NY, USA, 2012. ACM.

[3] R. Dong, M. Schaal, M. P. O'Mahony, K. McCarthy, and B. Smyth. Opinionated product recommendation. In *Case-Based Reasoning Research and Development*, volume 7969 of *Lecture Notes in Computer Science*, pages 44–58. Springer Berlin Heidelberg, 2013.

[4] R. Dong, M. Schaal, M. P. O'Mahony, and B. Smyth. Topic extraction from online reviews for classification and recommendation. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, IJCAI '13 (To appear), Menlo Park, California, 2013. AAAI Press.

[5] S. Garcia Esparza, M. P. O'Mahony, and B. Smyth. On the real-time web as a source of recommendation knowledge. In *Proceedings of the fourth ACM Conference on Recommender Systems*, RecSys '10, pages 305–308, New York, NY, USA, 2010. ACM.

[6] M. Hu and B. Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 168–177, New York, NY, USA, 2004. ACM.

[7] J. S. Justeson and S. M. Katz. Technical terminology: Some linguistic properties and an algorithm for identification in text. *Natural Language Engineering*, 1(1):9–27, 1995.

[8] S. Moghaddam and M. Ester. Opinion digger: An unsupervised opinion miner from unstructured product reviews. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pages 1825–1828, New York, NY, USA, 2010. ACM.

[9] M. Pazzani and D. Billsus. Content-based recommendation systems. In *The Adaptive Web*, volume 4321 of *Lecture Notes in Computer Science*, pages 325–341. Springer Berlin Heidelberg, 2007.

[10] A.-M. Popescu and O. Etzioni. Extracting product features and opinions from reviews. In *Natural Language Processing and Text Mining*, pages 9–28. Springer London, 2007.

[11] K. Zhang, R. Narayanan, and A. Choudhary. Voice of the customers: Mining online customer reviews for product feature-based ranking. In *Proceedings of the 3rd Workshop on Online Social Networks*, WOSN '10, Berkeley, CA, USA, 2010.