Noname manuscript No.
(will be inserted by the editor)

Combining Similarity and Sentiment in Opinion Mining for Product Recommendation

Ruihai Dong · Michael P. O'Mahony · Markus Schaal · Kevin McCarthy · Barry Smyth.

Received: date / Accepted: date

Abstract In the world of recommender systems, so-called content-based methods are an important approach that rely on the availability of detailed product or item descriptions to drive the recommendation process. For example, recommendations can be generated for a target user by selecting unseen products that are similar to the products that the target user has liked or purchased in the past. To do this, content-based methods must be able to compute the similarity between pairs of products (unseen products and liked products, for example) and typically this

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.

R. Dong
CLARITY: Centre for Sensor Web Technologies
School of Computer Science and Informatics
University College Dublin, Dublin, Ireland
E-mail: ruihai.dong@ucd.ie

M. P. O'Mahony
Insight Centre for Data Analytics
School of Computer Science and Informatics
University College Dublin, Dublin, Ireland
E-mail: michael.omahony@ucd.ie

M. Schaal
CLARITY: Centre for Sensor Web Technologies
School of Computer Science and Informatics
University College Dublin, Dublin, Ireland
E-mail: markus.schaal@ucd.ie

K. McCarthy
Insight Centre for Data Analytics
School of Computer Science and Informatics
University College Dublin, Dublin, Ireland
E-mail: kevin.mccarthy@ucd.ie

B. Smyth
Insight Centre for Data Analytics
School of Computer Science and Informatics
University College Dublin, Dublin, Ireland
E-mail: barry.smyth@ucd.ie

is achieved by comparing product features or other descriptive elements. The approach works well when product descriptions are readily available and when they are detailed enough to afford an effective similarity comparison. But this is not always the case. Detailed product descriptions may not be available since they can be expensive to create and maintain. In this article we consider another source of product descriptions in the form of the user-generated reviews that frequently accompany products on the web. We ask whether it is possible to mine these reviews, unstructured and noisy as they are, to produce useful product descriptions that can be used in a recommendation system. In particular we describe a novel approach to product recommendation that harnesses not only the features that can be mined from user-generated reviews but also the expressions of sentiment that are associated with these features. 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, and we demonstrate the practical benefits of this approach across a variety of Amazon product domains.

Keywords User-generated Reviews · Opinion Mining · Sentiment-based Product Recommendation

1 Introduction

Product recommender systems have, for a long time, relied on two primary sources of recommendation knowledge, either *user ratings* [55, 56, 10, 53, 29] or *product descriptions* [46, 35, 57, 5]. For example *collaborative filtering* approaches [56, 53] rely on the former to identify a neighbourhood of users who are similar to some target user to act as a source of item recommendations; basically products are selected for recommendation based on their popularity and/or ratings amongst the similar users. Alternatively, when product descriptions are available then *content-based* [46, 35] or *case-based* [57, 5] recommendation approaches can be used, selecting products for recommendation because they are similar to those that the target user has liked in the past. Each of these approaches have their own advantages and disadvantages and can often be used in concert (so-called *hybrid* recommenders [6]) for more effective recommendation.

More recently, researchers have started to consider other sources of recommendation knowledge, particularly in light of the deluge of social media information and other forms of user-generated content that has suffused modern society. For example, services like Facebook and Twitter have become a new destination for advertisers precisely because of the close connection between their users and products and brands. Simply put, these services are awash with user opinions, positive and negative, about brands, large and small, and products, far and wide. This begs the question as to whether these opinions can be usefully mined from Twitter, Facebook, and related services, to be used as the basis for recommendation tasks. For example, the work of [18] mined movie preferences and opinions from Blippr users as the basis for a conventional movie recommender system. Blippr allows users to contribute short tweet-like reviews of movies and this work constructed user profiles based on preference information that could be gleaned from these micro-reviews. The resulting system proved to be every bit as effective as more

conventional recommendation approaches and was demonstrated to be equally effective across other product categories including music, books, and applications. In a similar vein the work of [16] demonstrated how (movie) ratings information, similar to that required by collaborative filtering systems, could be extracted at scale from Twitter. Another popular approach is to use Twitter data for news recommendation by profiling people's interests based on the conversations of their social networks; see [48].

In this work we consider an alternative form of user-generated content, namely the type of product reviews that routinely accompany products on sites such as Amazon, TripAdvisor, etc. Consider, for example, the *ThinkPad X1 Carbon 14" Touchscreen Laptop*. At the time of writing its *product features*, as listed by Amazon, cover technical details such as *screen-size*, *RAM*, *processor speed*, *weight*, and *battery life*. These are the type of features that one might expect to find in a conventional content-based recommender [47]. 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 in any practical sense. However, the *ThinkPad X1 Carbon* has more than 60 reviews which encode valuable insights into a great many of its features; from its *"beautiful design"*, *"light weight"* and *"really fast bootup"* 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, as such, provide an objective basis for product comparisons.

We consider the following questions in this article. Can we use the features and opinions described above as the basis for a new type of *experiential* product recommendation, which is based on genuine user experiences? Can we use these features to generate product cases and are such cases rich enough to guide product recommendation? And what type of recommendation strategies might we use? To address these questions we describe a technique for automatically extracting opinionated product descriptions from user generated reviews and a flexible approach to recommendation that combines product similarity and feature sentiment. We also describe the results of a detailed evaluation of this approach across a variety of Amazon product domains.

The work presented in this article is based on recent work presented by the authors in [12,14]. In this earlier work we described an approach to recommendation that combined a simple overlap-based model of feature similarity with sentiment analysis and demonstrated it across 3 product domains. The current work extends this in a number of important ways. First and foremost, we describe an improved approach to combining product similarity and feature sentiment which provides much greater configuration flexibility and accommodates a wider range of similarity assessment techniques. Second, we describe two different approaches to computing feature sentiment that avoids the ad hoc treatment of so-called residual features as described in [14]. Thirdly, we explore the use of clustering techniques as a way to identify common features that might be treated as independent features using our standard opinion mining approach. And finally we evaluate these new approaches on 6 different product domains.

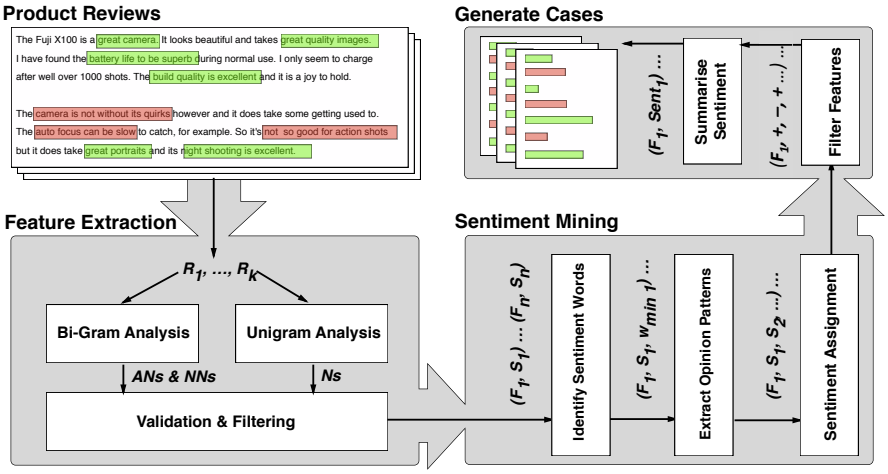


Fig. 1 Extracting experiential product cases from user-generated reviews.

2 Mining Product Experiences

The central aim of this work is to implement a practical technique for turning user-generated product reviews into rich, feature-based, experiential product cases. The features of these cases relate to topics that are discussed by reviewers and their aggregate opinions. Our intuition is that such features may provide access to a greatly expanded set of product features that would be unlikely to appear in classical catalog descriptions. Moreover, the availability of user opinions for these features provides access to a rich source of experiential information that is conspicuous by its absence from other recommendation approaches.

Our 3-step opinion mining approach is summarised in Figure 2 and its different component parts are based on the work of others in the opinion-mining literature; see for example [21,25,22,39]. (1) For a given product domain (e.g. digital cameras, printers, etc.) we use shallow NLP techniques to extract a set of candidate features from the reviews of all products in that domain; then, each particular product P in the domain is represented by a subset of these features which appear in $Reviews(P) = \{R_1, R_2, \ldots, R_k\}$, the reviews of P. (2) For each feature, F_i, we count how frequently it is associated with *positive*, *negative*, or *neutral* sentiment based on the opinions expressed in the reviews of P. (3) These features and sentiment scores are aggregated at the product level to generate a case of features and overall sentiment scores.

2.1 Extracting Review Features

We consider two basic types of features — *bi-gram* features and *single-noun* features — and use a combination of shallow NLP and statistical methods to mine them [21,25]. For the former we look for bi-grams in reviews which conform to one of two basic part-of-speech co-location patterns: (1) an adjective followed by a noun (AN) (e.g. *wide angle*); or (2) a noun followed by a noun (NN) (e.g.

video mode). These candidate features are filtered to avoid including *AN*'s that are actually opinionated single-noun features; e.g. *great flash* is really a single-noun feature, *flash*. To do this we exclude bi-grams whose adjective is a sentiment word (e.g. *excellent, terrible* etc.) in the sentiment lexicon which we use in this work[21][1].

For single-noun features we also extract a candidate set, this time of nouns, from the reviews but we validate them by eliminating nouns that are rarely associated with sentiment words as per [22]. The reason is that such nouns are unlikely to refer to product features (examples of such nouns include *month*, *friends* and *day* etc.). We calculate how frequently each feature co-occurs with a sentiment word in the same sentence, and retain a single-noun only if its frequency is greater than some fixed threshold (in this case 30%).

2.2 Evaluating Feature Sentiment

To calculate feature sentiment we use a version of the *opinion pattern mining* technique proposed in [39] for extracting opinions from unstructured product reviews. For a given feature F_i, and the corresponding review sentence S_j in review R_k, we determine whether there are any sentiment words in S_j. If there are not then this feature is labeled as *neutral*. Otherwise we identify the sentiment word w_{min} which is closest to F_i. Next we identify the part-of-speech (POS) tags for w_{min}, F_i and any words that occur between w_{min} and F_i. This POS sequence is an *opinion pattern*. For example, in the case of the bi-gram feature *screen quality* and the review sentence, *"...this tablet has excellent screen quality..."* then w_{min} is the word *"excellent"* which corresponds to an opinion pattern of *JJ-FEATURE* [39].

After a complete pass over all features we compute the frequency of occurrence of all opinion patterns. A pattern is deemed to be valid if it occurs at least twice following the approach in [39]. For valid patterns we assign sentiment based on the sentiment of w_{min} and subject to whether S_j contains any negation terms within a 4-word-distance of w_{min}. If there are no such negation terms then the sentiment assigned to F_i in S_j is that of the sentiment word in the sentiment lexicon. Otherwise the sentiment is reversed. If an opinion pattern is deemed not to be valid (based on its frequency) then we assign a *neutral* sentiment to each of its occurrences within the review set.

2.3 Generating Experiential Product Cases

For each product P we now have a set of features $F(P) = \{F_1, ..., F_m\}$ extracted from $Reviews(P)$, and how frequently each feature F_i is associated with *positive, negative*, or *neutral* sentiment in the particular reviews in $Reviews(P)$ that discuss F_i. For the purpose of this work we only include features in a product case if they are mentioned in more than 10% of the reviews for that product. For these features we calculate an overall sentiment score as shown in Equation 1 and their popularity as per Equation 2. Then each product case, $Case(P)$, can be represented as shown

[1] The sentiment lexicon from [21] contains lists of 2,009 positive words and 4,783 negative words. There are no weights assigned to words in the sentiment lexicon; rather, all words are considered to equally reflect either positive or negative sentiment.

in Equation 3. Note, $Pos(F_i, P)$, $Neg(F_i, P)$, and $Neut(F_i, P)$ denote the number of times that feature F_i has positive, negative and neutral sentiment in the reviews for product P, respectively.

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

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

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

3 Recommending Products

Given the feature-based product representations above it is most natural to consider a content-based/case-based approach to recommendation [46,57]: to retrieve and rank recommendations based on their feature similarity to a query product. We will describe one such technique in what follows. However, the availability of feature sentiment hints at an alternative approach to recommendation in which new products can be recommended because they offer *improvements* over certain features of the query product. We will also describe just such an alternative and a hybrid technique that allows for the flexible combination of similarity and sentiment.

3.1 Similarity-Based Recommendation

In our content-based recommendation strategy, each product case is represented as a vector of features and corresponding popularity scores as per Equation 2. As such, the *value* of a feature represents its frequency in reviews as a proxy for its importance. Then we use the cosine metric to compute the similarity between the query product, Q, and candidate recommendation, C as per Equation 4.

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

Clearly this is a very simple content-based recommendation technique, but it is in-line with many conventional approaches [55,47], and serves as a useful baseline to evaluate the more sophisticated methods described below. As an aside we could have also considered a variation on the above where feature values were sentiment rather than popularity scores and, indeed, we have previously considered this in related work [14].

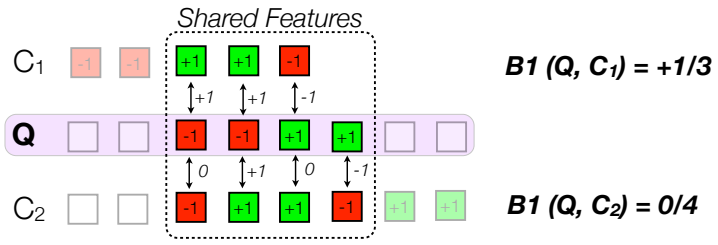


Fig. 2 Case C_1 offers better sentiment improvement than C_2 when compared to the query product Q based on shared features.

3.2 Sentiment-Enhanced Recommendation

The availability of feature sentiment suggests a very different approach to recommendation. Rather than looking for products that are *similar* to a query product, either in terms of feature popularity, as above, or feature sentiment as in [14], why not look for products that offer *better* sentiment scores than the query product?

For example, consider a user who is considering a particular digital camera. One of the features extracted from the camera's reviews is "lens quality" and let us assume that it has a popularity score of 0.2; indicating that about 20% of the reviews refer to this feature. Let us also assume an intermediate sentiment score of 0.25; indicating a weak positive sentiment. When selecting a new camera for recommendation should we, all other things being equal, look for other cameras that have lens quality mentioned in a similar proportion of reviews or that have a similar overall sentiment associated with lens quality? Or should we seek to find cameras that offer an improved sentiment score for this feature? Surely the latter makes more sense in the context of likely consumer preferences?

The starting point for this is the *better* function shown as Equation 5, which calculates a straightforward *better score* for feature F_i between query product Q and recommendation candidate C. A better score less than 0 means that the query product Q has a better sentiment score for F_i than C whereas a positive score means that C has the better sentiment score for F_i compared to Q.

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

We can then calculate an overall *better score* at the product level by aggregating the individual better scores for the product features. There are two obvious ways to do this. First, in Equation 6 we compute the average better scores across the features that are shared between Q and C. However, this approach ignores those (potentially many) features that may be unique to Q or C, so called *residual features*. For instance, in Figure 2 we see an example of the approach for two candidate recomendations, C_1 and C_2, with respect to a query case Q. In terms of their shared features, C_1 offers a better sentiment improvement than C_2 and so would be selected ahead of C_2 on sentiment grounds during recommendation.

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

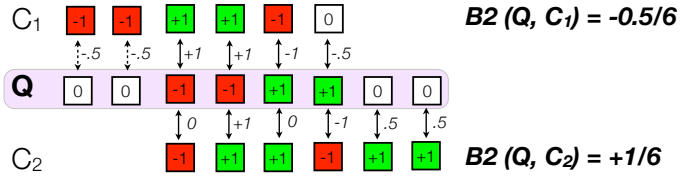


Fig. 3 Case C_2 offers better sentiment improvement than C_1 when compared to the query product Q based on shared and residual features.

A second alternative, to deal with these residual features, is to assign non-shared features a neutral sentiment score of 0 and then compute an average better score across the union of features in Q and C as in Equation 7.

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

In Figure 3 we return to our example of C_1 and C_2 above, but this time their fortunes are reversed based on a comparison of all (shared plus residual) features. This time C_2 wins out over C_1 with respect to Q.

3.3 Combining Similarity and Sentiment

The above provides two alternatives for a sentiment-based approach to recommendation, which ranks product cases in decreasing order of their better score (either $B1$ or $B2$). They prioritise recommendations 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. What is required is a way to combine similarity and sentiment during recommendation so that we can prioritise products that are similar to the query product while also being more positively reviewed.

Perhaps the simplest way to combine similarity and sentiment approaches is to use a hybrid scoring metric such as that shown in Equation 8; in this instance $Sent(Q,C)$ can be implemented as either $B1$ or $B2$ above[2]. Thus we compute an overall score for a candidate recommendation C based on a combination of C's similarity and sentiment scores with respect to Q. In what follows we will use this as our basic recommendation ranking approach, implementing versions that use $B1$ and $B2$ and varying the parameter w to control the relative influence of feature similarity and sentiment during recommendation.

$$Score(Q,C) = (1-w) \times Sim(Q,C) + w \times \left(\frac{Sent(Q,C)+1}{2} \right) \tag{8}$$

[2] The range of both $B1$ and $B2$ is [-1, +1]. Since the range of $Sim(Q,C)$ is [0, +1], $Sent(Q,C)$ is normalised to [0, +1] in Equation 8.

4 Evaluation

Thus far we have presented two core technical contributions: (1) a technique for extracting feature-based product descriptions from user-generated reviews; and (2) an approach to generating product recommendations that leverages a combination of feature similarity and review sentiment. We now describe the results of a comprehensive experiment designed to evaluate different aspects of both of these contributions using a multi-domain product dataset from Amazon. In particular, we will focus on evaluating the type of product descriptions that can be extracted, in terms of the variety of features and sentiment information, as well as assessing their suitability for recommendation based on similarity and sentiment scores. Importantly this will include an analysis of the benefits of using these approaches in a practical recommendation setting, and by comparison to Amazon's own recommendations.

4.1 Datasets

The data for this experiment was extracted from Amazon.com during October 2012. We focused on 6 different product categories: *Digital Cameras, GPS Devices, Laptops, Phones, Printers*, and *Tablets*. For each product, we extracted review texts and helpfulness information, and the top n ($n = 5$) ranked recommendations for 'related' products as suggested by Amazon[3]. In our analysis, we only considered products with at least 10 reviews; see Table 1 for dataset statistics.

Category	#Reviews	#Products	$\mu_{features}$	$\sigma_{features}$
Cameras	9,355	103	30.77	12.29
GPS	12,115	119	24.32	10.82
Laptops	12,431	314	28.60	15.21
Phones	14,860	257	9.35	5.44
Printers	24,369	233	16.89	7.60
Tablets	17,936	166	26.15	10.48

Table 1 Dataset statistics.

4.2 Mining Rich Product Descriptions

The success of the recommendation approach developed in this work depends critically on our ability to translate user-generated reviews into useful product cases; in the sense that they are rich enough, in terms of their features, to form the basis of recommendation.

4.2.1 Feature Histograms

As a starting point consider the feature histograms in Figure 4 for the cases extracted for each product type and showing the number of product cases of different

[3] In this case, related products are those as suggested by Amazon's "Customers who viewed this item also viewed these items" approach to recommendation.

sizes (that is, different numbers of features) that were produced. Generally speaking we can see that our feature mining approach is extracting rich product cases, many of which contain reasonably large numbers of features. For example, we can see that *Laptop* cases (Figure 4(c)) contain a wide range of case sizes, from small cases with very limited features sets of less than 10 to cases with as many as 70 features; the majority of cases contain somewhere between 15 and 30 features. In contrast *Phones* (Figure 4(d)) have a much narrower feature distribution; most have 5-15 features while very few have more than 20 features, indicating that users provide opinions on a narrow range of features in their reviews of phones compared to laptops.

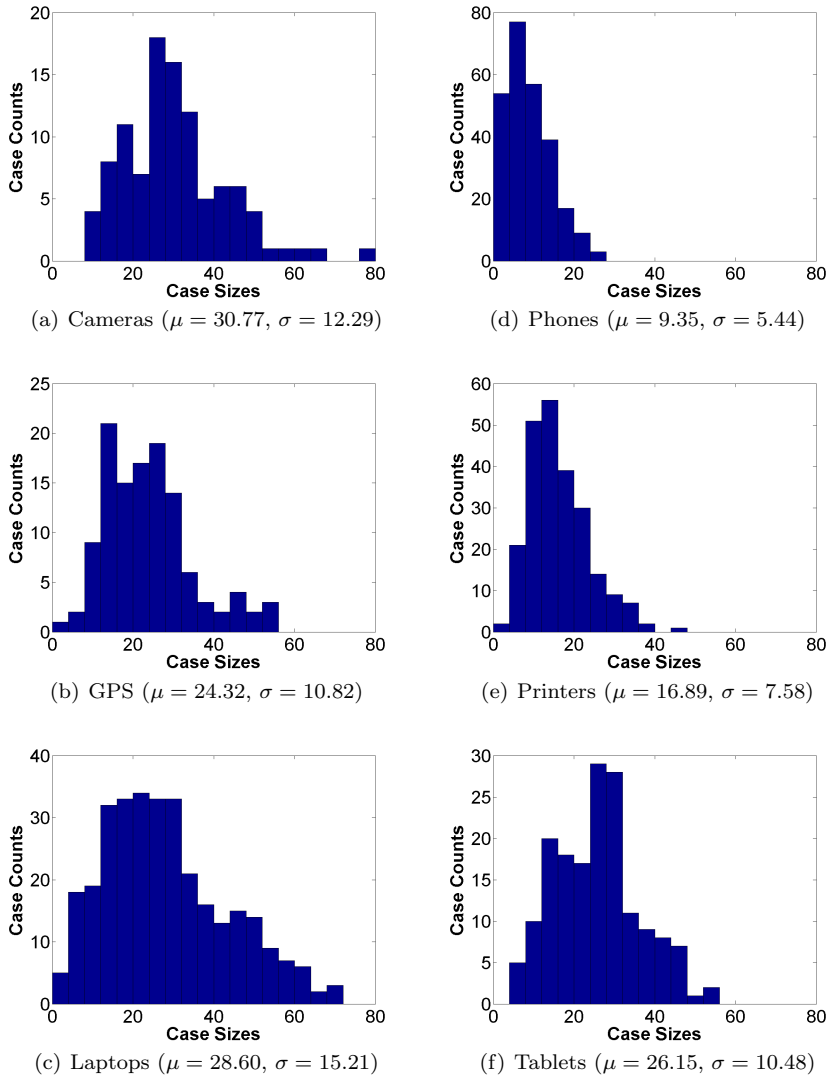
4.2.2 Product Similarity

The last two columns in Table 1 show the mean and standard deviation of the number of features that are extracted across the 6 product domains. It should be clear that we can expect to generate reasonably feature-rich cases from our review mining approach as 10-30 features are extracted per product case on average. However, this is of limited use if the variance in similarity between products in each category is low. Figure 5 shows histograms for the similarity values between all pairs of products for each of the 6 Amazon domains. Once again the results bode well because they show a wide range of possible similarity values, rather than a narrow range of similarity which may suggest limitations in the expressiveness of the extracted product representations.

4.2.3 Sentiment Heatmaps

It is also interesting to look at the different types of sentiment expressed for different features in the product categories. For example, Figure 6 shows the sentiment heatmap for the *Laptops* product category. Rows correspond to product cases and columns to their features. The sentiment of a particular feature is indicated by colour, from red (strong negative sentiment) to green (strong positive sentiment); missing features are shown in grey. (To fully appreciate this representation, the heatmap should be viewed in colour.) Both the feature columns and product rows are sorted by average sentiment.

There are a number of observations to make. First, because of the ordering of the features we can clearly see that features with the highest (leftmost) and lowest (rightmost) sentiment scores also tend to elicit the most opinions from reviewers; the leftmost and rightmost regions of the heatmap are the most densely populated. By and large there is a strong review bias towards positive or neutral opinions; there are far more green and yellow cells than red. (Similar trends are observed for the other product categories.) Some features are almost universally liked or disliked. For example, for *Laptops* the single most liked feature is *price* with *keyboard* and *screen* also featuring highly. In contrast, features such as *fan noise* and *wifi* are among the most universally disliked *Laptop* features. Across the product domains, *price* is generally the most liked feature, suggesting perhaps that modern consumer electronics pricing models are a good fit to consumer expectations, at least currently.



(a) Cameras ($\mu = 30.77$, $\sigma = 12.29$)

(d) Phones ($\mu = 9.35$, $\sigma = 5.44$)

(b) GPS ($\mu = 24.32$, $\sigma = 10.82$)

(e) Printers ($\mu = 16.89$, $\sigma = 7.58$)

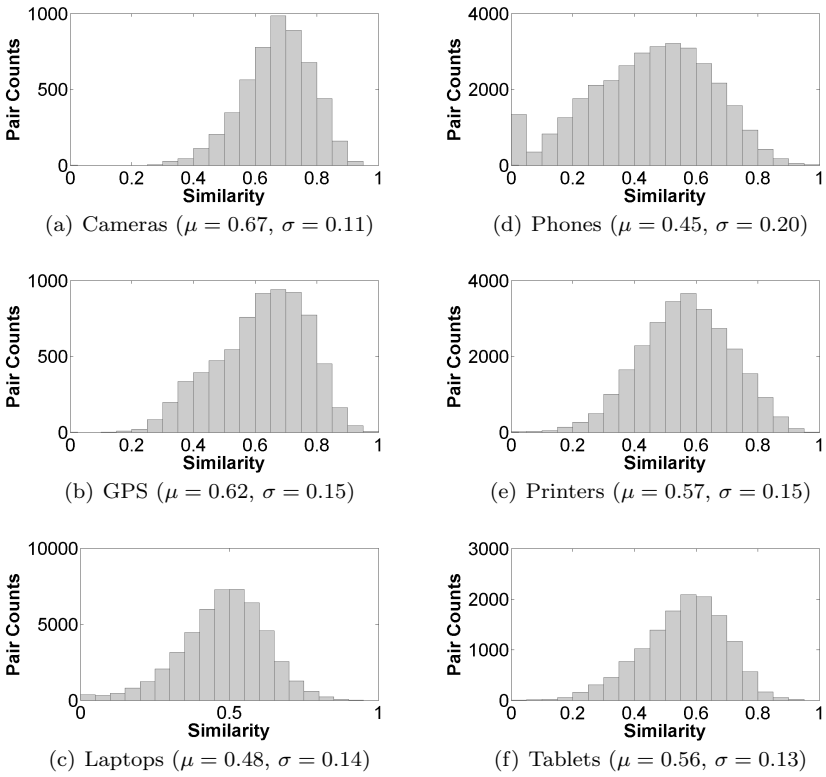
(c) Laptops ($\mu = 28.60$, $\sigma = 15.21$)

(f) Tablets ($\mu = 26.15$, $\sigma = 10.48$)

Fig. 4 Product case size histograms.

4.3 Recommendation Performance

To evaluate our recommendation approach we use a standard *leave-one-out* approach, comparing our recommendations, for each query product Q, to those produced by Amazon; as discussed previously we scraped Amazon's recommendations during our dataset collection phase. Specifically, for each query product Q in a given domain we generate a set of $n = 5$ ranked recommendations using Equation 8 instantiated with $B1$ and $B2$; we do this for each value of w from 0 to 1 in steps of 0.1. This produces 22 recommendation lists, for each Q, 11 for each of $B1$ and $B2$, which we compare to Amazon's own recommendations for Q.



(a) Cameras ($\mu = 0.67$, $\sigma = 0.11$)
(d) Phones ($\mu = 0.45$, $\sigma = 0.20$)
(b) GPS ($\mu = 0.62$, $\sigma = 0.15$)
(e) Printers ($\mu = 0.57$, $\sigma = 0.15$)
(c) Laptops ($\mu = 0.48$, $\sigma = 0.14$)
(f) Tablets ($\mu = 0.56$, $\sigma = 0.13$)

Fig. 5 Product similarity histograms.

4.3.1 Recommendation Precision

We calculate a standard precision metric to compare our recommendations to Amazon's, by calculating the percentage of our recommendations that are contained in Amazon's recommendation lists. Figure 7 presents these results averaged over all products for each of the six product domains as we vary w. We can see that lower values of w (< 0.5), where feature similarity plays a major ranking role, producing recommendation lists that include more Amazon recommendations compared to higher values of w (> 0.5) where feature sentiment plays the major role. For example, in the *Camera* domain lower values of w lead to stable precision scores of 0.4–0.5 but precision falls quickly for $w > 0.7$. This basic pattern is repeated across all six product domain, albeit with different absolute precision scores. The fact that precision is reasonably high for low values of w suggests that our similarity measure based on extracted features is proving to be useful from a recommendation standpoint as it enables us to suggest some of the same products as Amazon's own ratings-based recommender. As w increases, and feature sentiment begins to play a more influential role in recommendation ranking, both $B1$ and $B2$ start to prefer recommendation candidates that are not present in Amazon's recommendations and so precision falls.

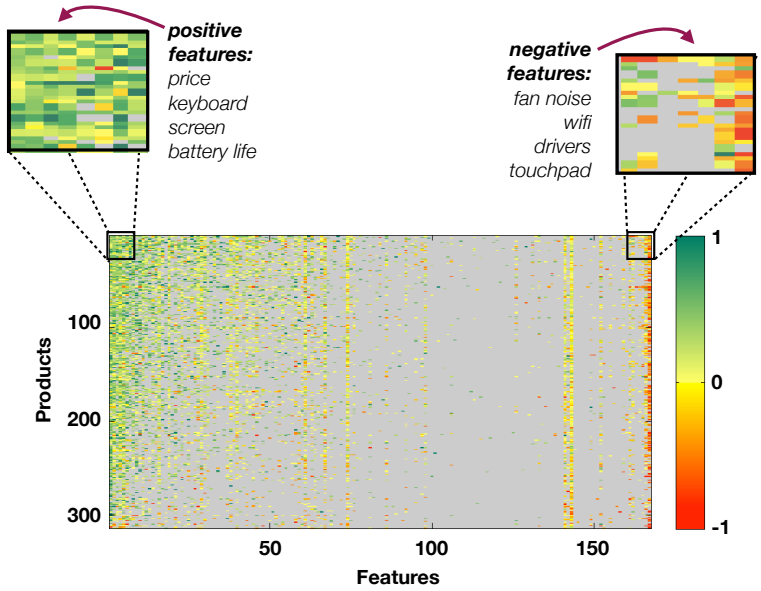


Fig. 6 Feature sentiment heatmap for the *Laptops* product category. The rows correspond to product cases and columns to features. The rows are sorted such that products which have the highest sentiment appear at the top. The columns are sorted such that the left-most features are those with highest sentiment (green colour); right-most features have lowest sentiment (red colour); missing features (i.e. features not discussed in reviews for particular product cases) are shown in grey.

Of course, as a practical matter, our objective is not necessarily to maximise this precision metric. It serves only as a superficial guide to recommendation quality relative to the Amazon baseline. But the real question is whether there is any evidence that the non-Amazon recommendations made by $B1$ and $B2$ are in any way superior to the Amazon recommendations, especially as when w increases, non-Amazon recommendations come to dominate.

4.3.2 Ratings Benefit

As an alternative to conventional precision metrics, we propose to use Amazon's overall product ratings (i.e. the average rating calculated over all reviews for each product) as an independent objective measure of product quality. Specifically, we compute a *relative benefit* metric to compare two sets of recommendations based on their ratings, as per Equation 9; e.g. a relative benefit of 0.15 means that our recommendations R enjoy an average rating score that is 15% higher that those produced by Amazon (A).

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

We also compute the average similarity between our recommendations and the current query product, using our mined feature representations; we refer to this

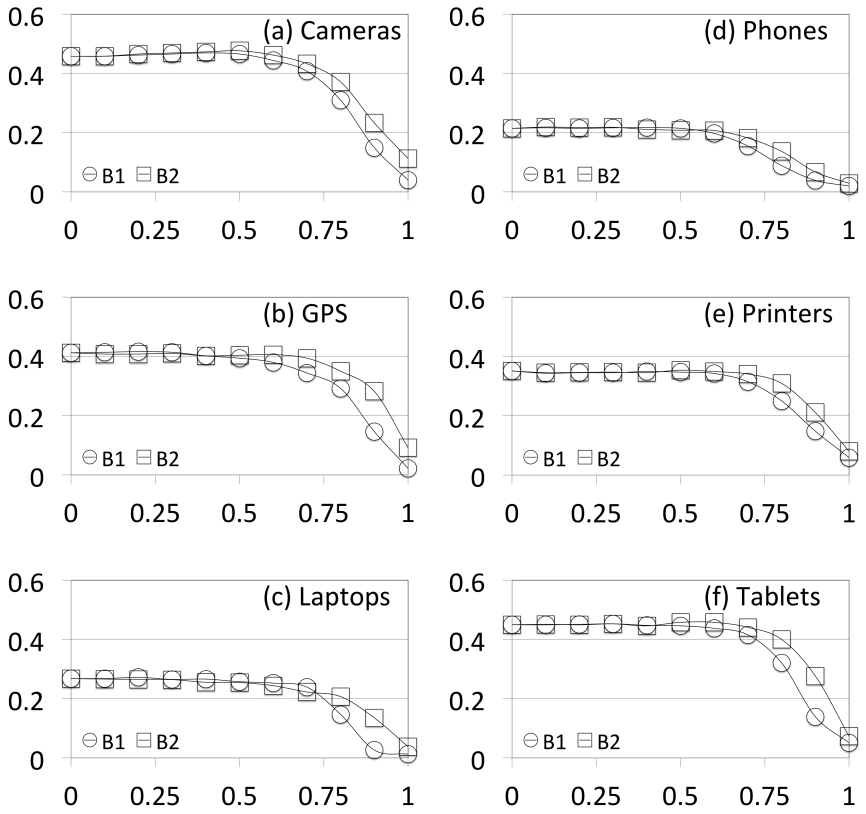


Fig. 7 Precision (y-axis) versus w (x-axis) for each product domain; $B1$ and $B2$ are presented as circles and squares on the line graphs, respectively.

as the *query product similarity*. This allows us to evaluate whether our techniques are producing recommendations that continue to be related to the query product — there is little benefit to recommending highly rated products that bear little or no resemblance to the type of product the user is looking for — and, as we shall see it also provides a basis for a more direct comparison to Amazon's own recommendations.

The results of this analysis are presented for the 6 product domains in Figure 8(a–f) for $B1$ and $B2$ when recommending $n = 5$ products. In each graph we show the benefit scores (left y-axis) for $B1$ and $B2$ (dashed lines) for varying values of w (x-axis), along with the corresponding query product similarity values calculated using Equation 4 (right y-axis, solid lines). We also show the average similarity (by Equation 4) between the query product and the Amazon recommendations, which is obviously unaffected by w and so appears as a solid horizontal line in each chart. These results allow us to examine the performance of a variety of different recommendation strategies based on the combination of mined feature similarity and user sentiment.

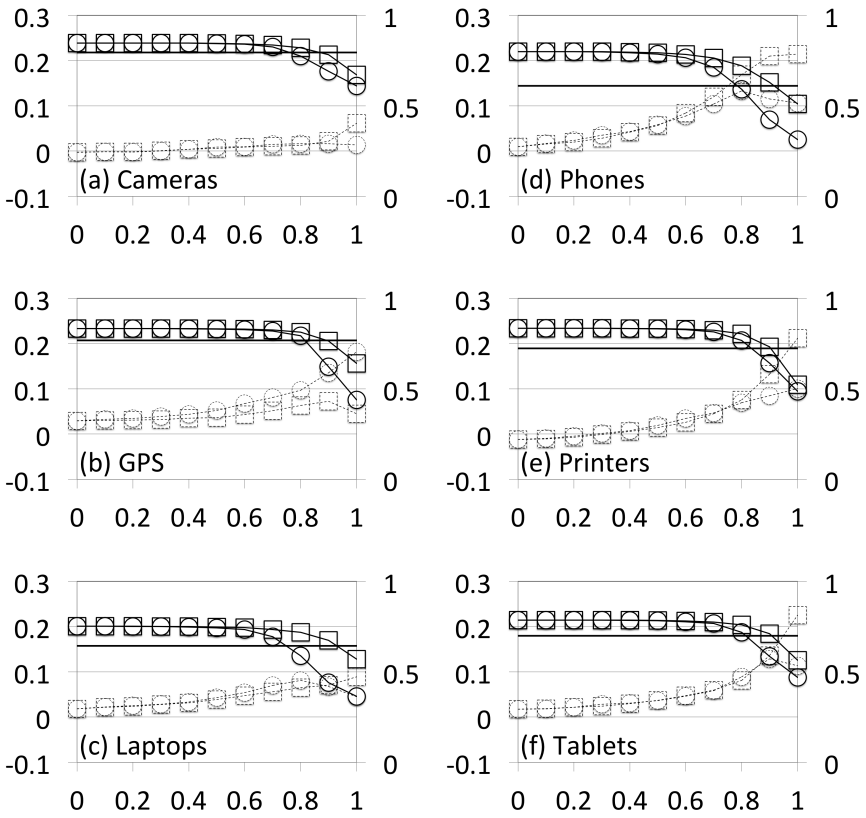


Fig. 8 Ratings benefit (left y-axis and dashed lines) and query similarity (right y-axis and solid lines) versus w (x-axis); $B1$ and $B2$ are presented as circles and squares on the line graphs respectively and the Amazon query similarity is shown as a solid horizontal line.

4.3.3 Contrasting Sentiment and Similarity

To begin with we will look at the extremes where $w = 0$ and $w = 1$. At $w = 0$ both $B1$ and $B2$ techniques are equivalent to a pure similarity-based approach to recommendation (i.e. using cosine as per Equation 4), 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 scores to those produced by Amazon – although both $B1$ and $B2$ tend to produce recommendations that 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* dataset (Figure 8(d)) at $w = 0$ we can see that $B1$ and $B2$ have a ratings benefit of approximately 0 and a query product similarity of just under 0.8 compared to approximately 0.6 for Amazon's comparable recommendations.

Another interesting configuration is at $w = 1$, which reflects an approach to recommendation that is based solely on sentiment and without any similarity component. In this configuration we can see a range of maximum positive ratings

benefits (from 0.06 to 0.23) across all 6 product domains. Further, $B2$ generally outperforms $B1$ (at this $w = 1$ setting). For example, looking again at the *Phones* dataset (Figure 8(d)), at $w = 1$ we see a ratings benefit of 0.21 for $B2$. In other words the products recommended by $B2$ enjoyed ratings that were approximately 21% higher than those products recommended by Amazon; it is worth noting that this represents on average an increase of almost one rating-scale point for Amazon's 5-point scale.

However, these ratings benefits are tempered by a drop in query product similarity. At $w = 1$, query product similarity falls to between 0.31 and 0.67, and typically below the query product similarity of the average Amazon recommendations (approximately 0.6–0.8 across the 6 product domains). Based on the similarity analysis from Figure 5 we can calibrate the extent of this drop by noting that, for $B1$ in particular, it often leads to recommendations whose average similarity to the query product is less than the average similarity between any random pair of products in a given domain. In other words there is a tradeoff between these ratings benefits and query product similarity and a likelihood that the better rated recommendations suggested by our approaches may no longer be sufficiently similar to the query product to meet the user's product needs or preferences.

4.3.4 Combining Similarity and Sentiment

By varying the value of w we can explore different combinations of similarity and sentiment during recommendation to better understand this tradeoff between query product similarity and ratings benefit. For example, as w increases we can see a gradual increase in ratings benefit for both $B1$ and $B2$, with $B2$ generally outperforming $B1$, especially for larger values of w. In some domains (e.g. *Cameras* and *Laptops*) the ratings benefit increase is more modest (< 0.1) whereas a more significant ratings benefit is observed for $GPS, Phones, Printers$, and $Tablets$. The slope of these ratings benefit curves and the maximum benefit achieved is influenced by the nature of the ratings-space in the different domains. For example, *Cameras* and *Laptops* have the highest average ratings and lowest standard deviations of ratings across the 6 domains. This suggests that there is less room for ratings improvement during recommendation. In contrast, *Phones* and *Tablets* have among the lowest average ratings and highest standard deviations and thus enjoy much greater opportunities for improved ratings.

As expected query product similarity is also influenced by w. For $w < 0.7$ we see little change in query product similarity. But for $w > 0.7$ there is a drop in query product similarity 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 $B1$.

Overall then we find that $B2$ tends to offer better ratings benefits and query product similarity than $B1$ but it is still difficult to calibrate these differences or their relationship to the Amazon baseline as w varies. We need a fixed 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 an interesting reference point because it allows us to look at the ratings benefit offered by $B1$ and $B2$

while delivering recommendations that have the same query product similarity as the baseline Amazon recommendations. For example, as shown in Figure 8(e), for *Printers* the query product similarity of $B1$ and $B2$ crossed that of Amazon at w values of 0.83 and 0.9, respectively. And at these w values they deliver ratings benefits of 8% and 14%, respectively. In other words our sentiment-based techniques are capable of delivering recommendations that are as similar to the query product as Amazon's but with a better average rating.

In Figure 9 we summarise these ratings benefits (bars) and the corresponding w values (lines) for $B1$ and $B2$. These results clarify the positive ratings benefits that are available using our sentiment-based recommendation techniques without compromising query product similarity. To evaluate the statistical significance of these results we used the Kruskall-Wallis test [30] on the raw ratings data that made up the ratings benefit results and similarity results above. As such the null hypothesis was that the 3 groups (Amazon baseline, *B1* and *B2*) of ratings (or similarity) data are all taken from the same underlying distributions. This null hypothesis was rejected at the 0.01 level of confidence, expect in the case of the *Digital Camera* product category. Next, we applied the Tukey-Kramer test to those product categories where this null hypothesis was rejected (*Laptops*, *GPS*, *Printers*, *Phones*, and *Tablets*) to perform a pairwise comparison between the 3 approaches (Amazon baseline, *B1* and *B2*), in order to determine which approaches were significantly different from the others. We found, in all cases, that both *B1* and *B2* were significant different (improved ratings benefit) from the Amazon baseline. Further, in the case of *Printers*, *Phones*, and *Tablets* we found that the differences observed between *B1* and *B2* were also statistically significant; in each case favouring *B2*, where large ratings benefits between 13% and 21% were observed for *B2* in these product categories. We also found that the differences observed across the similarity data were not statistically different. This means that our techniques were able to generate these ratings improvements without compromising on query similarity.

It is also interesting to note the consistency of the w values at which the query product similarity of the sentiment-based recommendations matches that of the Amazon recommendations, particularly for strategy $B2$ (0.87–0.93). As a practical matter this suggests that a w of about 0.9 will be sufficient to deliver recommendations that balance query product similarity with significant ratings benefits, thereby avoiding the need for domain-specific calibration.

4.4 Combining Related Features

The approach to opinion mining that we have described so far is susceptible to a proliferation of mined features because it is insensitive to the many and varied ways that people will inevitably refer to the same product features. For example, in hotel reviews North Americans will comment on the speed of the *elevators* while Europeans will talk about *lifts*. Some reviewers may refer to *customer support* while others will talk about *customer service* or, in camera reviews, *pictures* versus *photos*. Other features may in fact be subtly different but it will make sense to consider them as a single feature; for example, *touch pads* versus *mouse pads*. In all of these examples our current opinion mining technique will fail to recognise the common features.

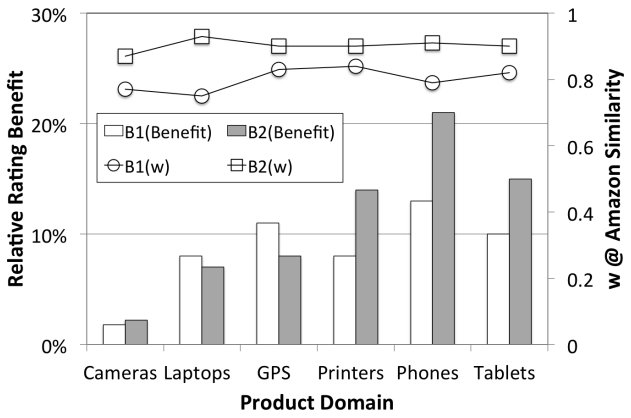


Fig. 9 Summary ratings benefits at the Amazon baseline query product similarity.

4.4.1 Feature Clustering

One way to address this is to attempt to cluster similar features together on the basis of similarities in the way that they are referred to in user generated reviews. For example, we can associate each extracted feature with a description vector that is made up of the set of terms extracted from the sentences that refer to this feature; see Equation 10 where $Sens(F_i)$ denotes the set of sentences in which feature F_i occurs and $Terms(S_k)$ denotes the set of terms contained in a sentence S_k[4]. Thus each feature F_i is associated with a set of terms and each feature can be associated with a normalized term frequency weight, w_j [36]. In this way, each feature can be compared based on the similarity of their description vectors.

$$Desc(F_i) = \bigcup_{\forall S_k \in Sens(F_i)} \{t_j : t_j \in Terms(S_k), w_j\} \tag{10}$$

Next, we can apply standard clustering techniques to these description vectors. In this experiment we use CLUTO[5] and select a standard partitional clustering algorithm. In fact we consider two experimental conditions. In the *standard* condition, the objective is to take feature synonyms into account and to cluster related features together; for example, *picture* and *shots* are synonyms of feature *image*. By experiment, setting the target number of clusters to be 35% of the total number of features extracted for each domain (i.e. such that each cluster contains approximately three related features) provided good performance in this regard. Thus, this approach allows us to consider the performance of clustering with minimal fine-tuning and a particular objective (capturing feature synonyms) in mind.

The second clustering condition, *optimized*, considers a number of clustering algorithms and cluster criterion functions available from the CLUTO toolkit, and the best performing combination for each product domain over a range of partitions with different numbers of clusters is selected. This affords us with an opportunity

[4] All terms in sentences are first converted to lowercase, stop words are removed and the remaining terms are stemmed to their root form.

[5] http://glaros.dtc.umn.edu/gkhome/views/cluto

to evaluate performance when a greater degree of fine-tuning has been carried out in order to understand the potential of this particular variation.

4.4.2 Generating Cases from Clustered Features

Using this clustering approach we can modify the case generation step of our approach. Each case is now made up of a set of clusters and each cluster is comprised of a set of features. In effect, each cluster corresponds to a type of high-level feature, such that the features it contains are related in some way. For example, we might expect to find a cluster that corresponds to the "picture quality" of a digital camera and for it to contain features such as "image resolution", "picture clarity", "night images" etc. Now, for a given cluster C_j we compute its sentiment and popularity scores in a manner similar to the way in which we compute the individual feature scores in Equations 1 and 2, except that now each cluster contains a set of features. Thus the sentiment and popularity scores are each aggregated across all of these in-cluster features, $F_1, \ldots, F_m$, as per Equations 11 and 12:

$$Sent(C_j, P) = \frac{\sum_{F_i \in C_j} Pos(F_i, P) - \sum_{F_i \in C_j} Neg(F_i, P)}{\sum_{F_i \in C_j} Pos(F_i, P) + \sum_{F_i \in C_j} Neg(F_i, P) + \sum_{F_i \in C_j} Neut(F_i, P)} \tag{11}$$

$$Pop(C_j, P) = \frac{|\{R_k \in Reviews(P) : F_1 \in R_k \vee F_2 \in R_k \vee \ldots \vee F_m \in R_k\}|}{|Reviews(P)|} \tag{12}$$

where $Pos(F_i, P)$, $Neg(F_i, P)$ and $Neut(F_i, P)$ denote the number of times feature $F_i \in C_j$ has positive, negative and neutral sentiment in the reviews for product P $\big(Reviews(P)\big)$, respectively.

4.4.3 Preliminary Results

Our aim in this section is to evaluate whether this approach to clustering features offers any meaningful improvement compared to the simpler more direct feature extraction approach. To test this we re-run the experiments above but using the cases produced from the two clustering conditions. This means that we have two new sets of ratings benefits scores, for the *standard* and *optimized* conditions, for varying values of w as per Section 4.3.4. Further, as described in Section 4.3.4, we can perform a fixed-point comparison of the techniques by comparing the ratings benefit at the point at which the query product similarity intersects with the Amazon similarity level to give a single ratings benefit value per product domain for each of the clustering conditions and for *B1* and *B2*. We compare this ratings benefit to the corresponding ratings benefit produced from Section 4.3.4 to compute a *relative clustering improvement*. For example, in Section 4.3.4 we found that $B2$ produced a relative ratings benefit of 7% for the *Laptops* domain; see Figure 9. Using clustering, the standard condition was found to deliver a corresponding relative ratings benefit of just over 10% for $B2$ in *Laptops*, or a relative clustering improvement of approximately 50%.

These results are presented in Figures 10(a) and 10(b) for the standard and optimized clustering conditions, respectively. In each graph we chart the relative

clustering improvement for $B1$ and $B2$ across the six product categories. They show varying levels of improvement due to clustering for $B1$ and $B2$ approaches, with the optimized clustering condition out-performing the standard condition. For example we can see that product domains such as *Cameras* and *Laptops* enjoy quite significant improvements, from about 20% to 170% (optimised condition), with $B2$ outperforming $B1$ in both domains. By comparison, *GPS, Printers, Phones* and *Tablets* enjoy more modest improvements with $B1$ outperforming $B2$; in fact in the standard clustering condition clustering has a negative effect on the ratings benefit for *Phones* and *Tablets* when $B2$ is used.

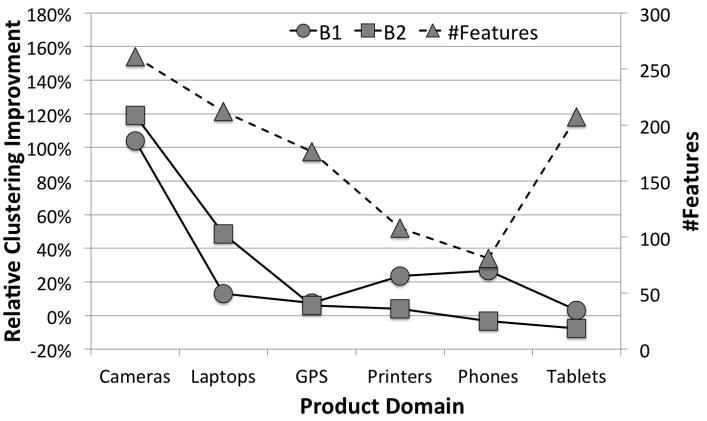
It is interesting to speculate as to why there is such a different in the benefits offered by feature clustering. In Figure 10(a) we also show total number of unique features extracted from each of the domains as a dotted line on the secondary y-axis. It is clear that there is a very strong correlation ($R_{B1} = 0.439$, $R_{B2} = 0.718$) between the number of features and the relative clustering improvement: the more features (one indicator of domain complexity) that are extracted the larger the clustering benefit. Larger numbers of mined features suggest a greater potential for clustering to bring common features together, for richer case representations, and deliver recommendation benefits as a result.

The outlier is the *Tablets* domain. It has a high apparent domain complexity (approximately 200 mined features) but benefits little from clustering. Our hypothesis for why this should be the case is that this domain is an immature one because it represents a relatively new product class. It is likely, therefore, that a stable vocabulary for describing tablets has yet to emerge. For example, it is clear from some reviews that people talk about tablets as if they are large smartphones while others describe them as if there are laptops. Thus, there is a level of confusion about feature terminology and the high feature count is more an artefact of this than it is a true reflection of an inherent domain complexity. This implies that the application of feature clustering provides little benefit and hence the reduced performance observed; a rigourous test of this hypothesis is left to future work.

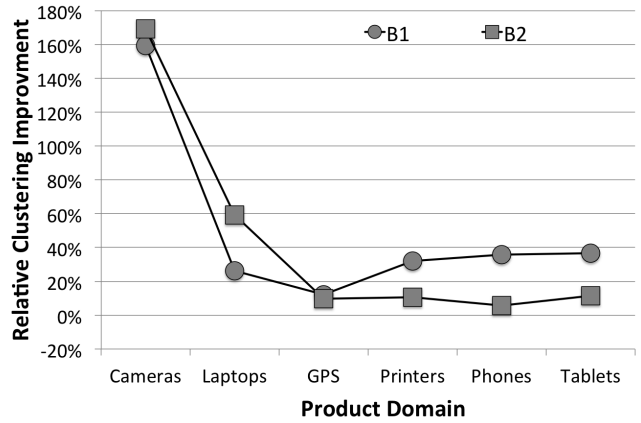
4.5 Summary Findings

The objective of this evaluation has been twofold: (1) to assess the quality of the product cases that are mined solely from product reviews; and (2) to evaluate the effectiveness of using these cases, and a combination of similarity and sentiment, during recommendation.

Regarding (1), it is clear that the product cases generated are feature rich with patterns of similar features extracted across many products. As to the quality of these features, the fact that they can be used as the basis for useful recommendations is a strong signal that they reflect meaningful product attributes; recommendations based solely on similarity share a strong overlap with those produced by Amazon, for example. More specifically, regarding (2) we have demonstrated that by combining feature similarity and sentiment we can generate recommendations that are comparable to those produced by Amazon (with respect to similarity) but enjoy higher overall user ratings, a strong independent measure of recommendation quality. Moreover our initial exploration of feature clustering suggests that there may be additional room for improved recommendations within this approach. Cer-



(a) Standard Clustering



(b) Optimized Clustering

Fig. 10 Relative clustering improvement at the Amazon baseline query product similarity.

tainly more complex domains can benefit significantly from feature clustering as a way to further enrich extracted case descriptions.

5 Related Work

The analysis of user-generated content such as online reviews has long been a fruitful research target with applications in a variety of domains. In particular the *opinions* expressed in user-generated content can provide powerful insights in many different contexts and the opinion mining literature is replete with a variety of significant challenges, mature techniques and significant results. This includes work in sub-areas such as *sentiment analysis* [31,44,8,62,63,64], *aspect-oriented opinion mining* [24,4,50,21,22,66,11], *opinion summarization* [21,32,70,58,26,45], *opin-*

ion search and retrieval [67,68,40,23], and *product review helpfulness estimation* [69,28,42,33,15] to name but a few.

For example, the task of estimating review helpfulness has received considerable attention in recent times as a way to guide users towards more helpful reviews; this is particularly important given the growth in user generated reviews online. For example, work in [42] adopts a classification-based approach to automatically classify reviews as helpful or not based on a variety of different content (e.g. review terms, readability, sentiment, ratings, etc.) and social features (e.g. reviewer history, reputation etc.). A similar approach was proposed in [27], where review ratings, length and unigram term distribution were found to be among the most discriminating features from a helpfulness perspective. Timeliness of reviews and reviewer expertise also proved to be useful predictors of movie review helpfulness [34], indicating that older reviews are less appreciated by consumers and that reviewers with an interest in and knowledge of particular movie genres are likely to author high quality reviews for similar-genre movies in the future. Recently the work of [15] proposed the automatic extraction of review features and sentiment using techniques similar to those presented in this article and demonstrate their benefit in improving helpfulness classification in addition to, or in the absence of, more conventional features; see also [2,20,43].

With the explosive growth in user-generated content, *sentiment analysis* has become a key research focus in recent times. In general, sentiment analysis has been explored at the *document-level*, *sentence-level* and the *aspect-level* [31]. At the document-level, the analysis is mainly concerned with classifying the overall polarity of a document as either positive or negative. For example, Pang et al. explored a supervised machine learning approach to classify the polarity of movie reviews, and found that a SVM-based approach using unigrams achieved best performance [44]; see also [62]. To move beyond the kind of high-level analysis provided at the document-level, researchers have also focused on analysing sentiment at the sentence-level. In this approach, sentences from documents are classified as either objective or subjective, and subjective sentences are further classified as either positive and negative; see, for example [63,41].

In aspect-oriented opinion mining [22], the focus is on extracting multi-faceted opinions that are akin to feature-based item descriptions. In the context of the work presented in this article this translates into the extraction of feature-based product descriptions from user generated reviews and other sources of opinions. For example, the work of [50] is representative in this regard and describes the use of shallow NLP techniques for explicit feature extraction and sentiment analysis; see also [21,22]. The features extracted, and the techniques used, are similar to those presented in this article, although in the case of the former there was a particular focus on the extraction of merenomic and taxonomic features to describe the *parts* and *properties* of a product. Zhang et al. [66] analyze the sentiment of comparative and subjective sentences in reviews on a per-feature basis to create a semi-ordering of products, but they do not consider the recommendation task with respect to a query product.

Building on the above work, *opinion summarisation* [70,58] is concerned with summarising the viewpoints relating to products and services obtained from multiple sources. For example, a technique to summarise and visualise opinions relating to product features mined from a set of customer reviews is presented in [21,32]. Further, work has been proposed to summarise contrasting viewpoints; see, for

example [26,45]. Work in the area *opinion retrieval*, combining research from the fields of information retrieval and sentiment analysis, is also of interest [38,67]. In this work, the objective is to identify relevant and opinionated entities; for example, locating Chinese restaurants specialising in Szechuan cuisine at reasonable prices. An example of this kind of work is the *RevMiner* application, which extracts attribute and value pairs from reviews and summarises opinions by clustering related attributes [23]. Other applications of sentiment analysis techniques include *The Stock Sonar* application [17], which mines positive and negative sentiment from news articles for stocks and visualizes sentiment together with stock price for the user; *Tweetfeel*[6], which monitors opinions contained in tweets relating to products such as movies and brands, etc; and the prediction of election outcomes by analyzing the political sentiment of tweets [61]; further applications can be found in [37,3].

In this article, the focus is on the development of novel recommendation approaches based on sentiment analysis techniques applied to user-generated content. As mentioned previously conventional recommender systems are typically based on ratings or transaction data (collaborative filtering) or on fixed content representations (content-based), and the idea of developing a recommendation framework based on noisy user-generated content remains novel in itself. The work of [18] is relevant in this regard in that it uses user-generated micro reviews as the basis for a text-based content recommender, and recently work in [9] has also tried to exploit user-generated content in similar ways. By providing a new source of recommendation knowledge user-generated reviews can enable recommender systems to operate in contexts that might otherwise be impossible. For example, reviews are leveraged to alleviate the well-known cold-start problem associated with collaborative recommenders [49]. In that work, the focus is on mining user preferences from review texts to reduce the sparsity of the user-item matrix; thereafter standard collaborative filtering algorithms are applied to the augmented user-item matrix to improve recommendation performance.

The work presented in this article clearly goes further than this body of related work in the area of recommender systems. It combines ideas from sentiment analysis, in particular aspect-oriented opinion mining, and recommender systems to demonstrate the practicality of harnessing reviews as a core knowledge source for future recommender systems. Moreover, the availability of opinion sentiment facilitates a style of recommendation that is simply not feasible with more traditional approaches. Our focus in this article has been very much on user-generated reviews driving a standalone recommender system. In reality it is more likely that this work would encourage use of user-generated reviews as part of a more holistic, hybrid approach to recommendation, one that combines catalog content with user reviews, and transactional or ratings information. How these different sources of knowledge can and should be combined for maximal benefit remains a matter for future work.

[6] http://www.tweetfeel.com/

6 Limitations and Future Work

As with any piece of research there are limitations to what we have achieved and, at the same time, opportunities for new lines of research as a result. Our broad objective with this research has been to investigate the potential for user-generated reviews to act as a new source of recommendation information. This is particularly important in many recommendation scenarios where traditional content-rich product descriptions may not be available or may be expensive to obtain. In fact this is one of the primary reasons why researchers have focused on content-free recommendation approaches such as collaborative filtering when it comes to generating recommendations in many domains. And in this paper we have provided evidence to support the hypothesis that user-generated reviews can form a rich source of product information that is suitable for the purpose of recommendation.

That being said, one of the limitations of this work is that in comparing our approaches to a baseline recommendation strategy we have had to adopt a relatively simplistic content-based recommendation strategy (see Equation 4 in Section 3.1). Briefly, we computed the similarity between two products (a query product and a recommendation product) based on the similarity of their mined features. And we used the popularity of these features in the product reviews as the feature value. In this way two products whose reviews mention price frequently are considered more similar (in terms of price) than two products where price is mentioned at very different frequencies. Clearly this is a very basic and somewhat naive approach to content similarity as we mentioned in Section 3.1. It was chosen precisely because our Amazon datasets did not provide any alternative feature-based content representations which would have been amenable to a more traditional case-based approach to product recommendation. Clearly, as part of future work, it would be appropriate to look for opportunities to consider domains where a more robust approach to product similarity could be used as a baseline. In fact since the writing of this paper we have carried out just such an evaluation using Trip Advisor data, which does provide feature-based hotel descriptions. In this Trip Advisor work (see [13]) we found very similar results to those presented here.

Another limitation of this work relates to the choice of techniques that we have adopted for the purpose of opinion mining and sentiment analysis. The purpose of this research has always been to explore the role of user generated reviews in recommendation rather than advance the state-of-the-art in opinion mining or sentiment analysis. As such we chose to adopt tried and tested approaches to opinion mining and sentiment analysis on the grounds that if these conventional approaches worked well, which they did, then it would still provide plenty of opportunity for further improvements as new opinion mining and sentiment analysis techniques emerged. Such opportunities include: the use of domain specific sentiment lexicons for sentiment analysis [51]; alternative approaches to evaluating negative sentiment, irony, and sarcasm in reviews [60, 54]; improved techniques for mining product features [65, 1] etc.

We also made certain simplifying assumptions about the various parameters that we chose for the purpose of evaluation and testing. For example, we choose to focus on products that had at least 10 reviews and this begs the question as to whether the approaches described are particularly sensitive to the number of reviews available. It is worth noting that in choosing 10 reviews we established a

reasonably low bar for the number of reviews needed to deliver reasonable results. It is not uncommon for products to have many more than 10 reviews, for example, and so we can expect that this particular configuration setting should not hinder the application of these ideas in other domains. Nevertheless it will be interesting to consider the impact of fewer or more reviews on recommendation performance as a matter for future work.

There are of course many other exciting prospects for future work in this area. Our grand vision, as stated above, has been to establish user-generated reviewed as a reliable source of product knowledge for recommender systems and as such they can now play a role in a host of related recommendation tasks from personalization to explanation. For example, in this article we have proposed a non-personalised approach to recommendation; i.e. recommending a set of products which are similar to, and better than, a query product. An obvious extension of our approach is to consider personalised recommendations; for example, by constructing user models based on the particular features (and associated sentiment) discussed in the reviews authored by users, and to recommend products that satisfy the individual feature preferences of users. In addition, the feature-based cases extracted for products (and users) can be leveraged for the purposes of recommendation explanation, which is key to user satisfaction with and acceptance of recommender systems [19, 59]. For example, the kind of explanations afforded by our approach, based on product features which are both liked by users and which are frequently mentioned (in a positive sense) in reviews for products, provide for an intuitive and, importantly, a personalised source of explanations for users. Moreover, our work has application in the area of conversational recommender systems [7, 5, 52] where, for example, critiques based on feature sentiment in addition to feature values can be provided in order to better facilitate and enhance the exploration of the product space.

7 Conclusions

Intuitively user-generated product reviews appear to provide a rich source of recommendation raw material but to the best of our knowledge these data sources have not been used as the basis for recommendation, at least in a direct way. In this article we have described an approach to mining product descriptions from raw review texts and we have shown how this information can be used to drive a novel recommendation technique that combines aspects of product similarity and feature sentiment. In turn we have presented results from a comprehensive evaluation across 6 Amazon product domains containing more than 1,000 products and 90,000 reviews. These results point to clear benefits in terms of recommendation quality, by combining similarity and sentiment information, compared to a suitable ground-truth (Amazon's own recommendations). Importantly, these recommendations have been produced without the need for large-scale transaction/ratings data (cf. collaborative filtering approaches) or structured product knowledge or metadata (cf. conventional content-based approaches).

While our evaluation has presented results from 6 different product domains it could be argued that, since these domains all relate to consumer electronics, the generalisability of our approach may be exaggerated; perhaps it works well for consumer electronics with technical features but not perhaps for other domains.

At the time of writing we have completed a similar evaluation for hotels using TripAdvisor hotel reviews. The results are presented in [13] and it is clear that they present a very similar picture both in terms of the type of hotel cases that are produced and our ability to produce useful recommendations by combining similarity and sentiment. On this basis we can be somewhat confident that the approaches we have described in this work provide a useful new approach to case-based product recommendation.

References

1. Archak, N., Ghose, A., Ipeirotis, P.G.: Deriving the pricing power of product features by mining consumer reviews. Management Science **57**(8), 1485–1509 (2011)
2. Baccianella, S., Esuli, A., Sebastiani, F.: Multi-facet rating of product reviews. In: Advances in Information Retrieval, 31th European Conference on Information Retrieval Research (ECIR 2009), pp. 461–472. Springer, Toulouse, France (2009)
3. Bar-Haim, R., Dinur, E., Feldman, R., Fresko, M., Goldstein, G.: Identifying and following expert investors in stock microblogs. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP '11, pp. 1310–1319. Association for Computational Linguistics, Stroudsburg, PA, USA (2011). URL `http://dl.acm.org/citation.cfm?id=2145432.2145569`
4. Boiy, E., Moens, M.F.: A machine learning approach to sentiment analysis in multilingual web texts. Information Retrieval **12**(5), 526–558 (2009)
5. Bridge, D., Göker, M.H., McGinty, L., Smyth, B.: Case-based recommender systems. The Knowledge Engineering Review **20**(03), 315–320 (2005)
6. Burke, R.: Hybrid recommender systems: Survey and experiments. User Modeling and User-Adapted Interaction **12**(4), 331–370 (2002). DOI 10.1023/A:1021240730564
7. Burke, R., Hammond, K., Yound, B.: The findme approach to assisted browsing. IEEE Expert **12**(4), 32–40 (1997). DOI 10.1109/64.608186
8. Dasgupta, S., Ng, V.: Mine the easy, classify the hard: A semi-supervised approach to automatic sentiment classification. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2, ACL '09, pp. 701–709. Association for Computational Linguistics, Stroudsburg, PA, USA (2009). URL `http://dl.acm.org/citation.cfm?id=1690219.1690244`
9. De Francisci Morales, G., Gionis, A., Lucchese, C.: 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, pp. 153–162. ACM, New York, NY, USA (2012). DOI 10.1145/2124295.2124315. URL `http://doi.acm.org/10.1145/2124295.2124315`
10. Desrosiers, C., Karypis, G.: A comprehensive survey of neighborhood-based recommendation methods. In: Recommender Systems Handbook, pp. 107–144. Springer (2011)
11. Ding, X., Liu, B., Yu, P.S.: A holistic lexicon-based approach to opinion mining. In: Proceedings of the 1st ACM International Conference on Web Search and Data Mining, pp. 231–240. ACM (2008)
12. Dong, R., O'Mahony, M.P., Schaal, M., McCarthy, K., Smyth, B.: Sentimental product recommendation. In: Proceedings of the 7th ACM Conference on Recommender Systems, RecSys '13, pp. 411–414. ACM, New York, NY, USA (2013). DOI 10.1145/2507157.2507199. URL `http://doi.acm.org/10.1145/2507157.2507199`
13. Dong, R., OMahony, M.P., Smyth, B.: Further experiments in opinionated product recommendation. In: Proceedings of the 22nd International Conference on Case-Based Reasoning, ICCBR '14, pp. 110–124. Springer (2014)
14. Dong, R., Schaal, M., O'Mahony, M.P., McCarthy, K., Smyth, B.: Opinionated product recommendation. In: Proceedings of the 21st International Conference on Case-Based Reasoning, ICCBR '13, pp. 44–58. Springer Heidelberg (2013)
15. Dong, R., Schaal, M., O'Mahony, M.P., Smyth, B.: Topic extraction from online reviews for classification and recommendation. In: Proceedings of the 23rd International Joint Conference on Artificial Intelligence, IJCAI '13. AAAI Press, Menlo Park, California (2013)

16. Dooms, S., De Pessemier, T., Martens, L.: Movietweetings: a movie rating dataset collected from twitter. In: Workshop on Crowdsourcing and Human Computation for Recommender Systems, CrowdRec at RecSys, vol. 13 (2013)
17. Feldman, R., Rosenfeld, B., Bar-Haim, R., Fresko, M.: The stock sonarsentiment analysis of stocks based on a hybrid approach. In: Proceedings of the 23rd IAAI Conference (2011)
18. Garcia Esparza, S., O'Mahony, M.P., Smyth, B.: On the real-time web as a source of recommendation knowledge. In: Proceedings of the fourth ACM Conference on Recommender Systems, RecSys '10, pp. 305–308. ACM, New York, NY, USA (2010). DOI 10.1145/1864708.1864773. URL http://doi.acm.org/10.1145/1864708.1864773
19. Herlocker, J.L., Konstan, J.A., Riedl, J.: Explaining collaborative filtering recommendations. In: Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work, CSCW '00, pp. 241–250. ACM, New York, NY, USA (2000). DOI 10.1145/358916.358995. URL http://doi.acm.org/10.1145/358916.358995
20. Hsu, C.F., Khabiri, E., Caverlee, J.: Ranking comments on the social web. In: Proceedings of the 2009 IEEE International Conference on Social Computing (SocialCom-09), pp. 90–97. Vancouver, Canada (2009)
21. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '04, pp. 168–177. ACM, New York, NY, USA (2004). DOI 10.1145/1014052.1014073. URL http://doi.acm.org/10.1145/1014052.1014073
22. Hu, M., Liu, B.: Mining opinion features in customer reviews. In: Proceedings of the 19th National Conference on Artifical Intelligence, AAAI'04, pp. 755–760. AAAI Press (2004). URL http://dl.acm.org/citation.cfm?id=1597148.1597269
23. Huang, J., Etzioni, O., Zettlemoyer, L., Clark, K., Lee, C.: Revminer: An extractive interface for navigating reviews on a smartphone. In: Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology, UIST '12, pp. 3–12. ACM, New York, NY, USA (2012). DOI 10.1145/2380116.2380120. URL http://doi.acm.org/10.1145/2380116.2380120
24. Jiang, L., Yu, M., Zhou, M., Liu, X., Zhao, T.: Target-dependent twitter sentiment classification. In: ACL, pp. 151–160 (2011)
25. Justeson, J.S., Katz, S.M.: Technical terminology: Some linguistic properties and an algorithm for identification in text. Natural Language Engineering **1**(1), 9–27 (1995)
26. Kim, H.D., Zhai, C.: Generating comparative summaries of contradictory opinions in text. In: Proceedings of the 18th ACM Conference on Information and Knowledge Management, CIKM '09, pp. 385–394. ACM, New York, NY, USA (2009). DOI 10.1145/1645953.1646004. URL http://doi.acm.org/10.1145/1645953.1646004
27. Kim, S.M., Pantel, P., Chklovski, T., , Pennacchiotti, M.: Automatically assessing review helpfulness. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2006), pp. 423–430. Sydney, Australia (2006)
28. Kim, S.M., Pantel, P., Chklovski, T., Pennacchiotti, M.: Automatically assessing review helpfulness. In: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, EMNLP '06, pp. 423–430. Association for Computational Linguistics, Stroudsburg, PA, USA (2006). URL http://dl.acm.org/citation.cfm?id=1610075.1610135
29. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **42**(8), 30–37 (2009)
30. Kruskal, W.H., Wallis, W.A.: Use of ranks in one-criterion variance analysis. Journal of the American Statistical Association **47**(260), 583–621 (1952)
31. Liu, B.: Sentiment analysis and opinion mining. Synthesis Lectures on Human Language Technologies **5**(1), 1–167 (2012)
32. Liu, B., Hu, M., Cheng, J.: Opinion observer: Analyzing and comparing opinions on the web. In: Proceedings of the 14th International Conference on World Wide Web, WWW '05, pp. 342–351. ACM, New York, NY, USA (2005). DOI 10.1145/1060745.1060797. URL http://doi.acm.org/10.1145/1060745.1060797
33. Liu, J., Cao, Y., Lin, C.Y., Huang, Y., Zhou, M.: Low-quality product review detection in opinion summarization. In: EMNLP-CoNLL, pp. 334–342 (2007)
34. Liu, Y., Huang, X., An, A., Yu, X.: Modeling and predicting the helpfulness of online reviews. In: Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008), pp. 443–452. IEEE Computer Society, Pisa, Italy (2008)
35. Lops, P., De Gemmis, M., Semeraro, G.: Content-based recommender systems: State of the art and trends. In: Recommender Systems Handbook, pp. 73–105. Springer (2011)

36. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to information retrieval, vol. 1. Cambridge University Press Cambridge (2008)
37. McGlohon, M., Glance, N.S., Reiter, Z.: Star quality: Aggregating reviews to rank products and merchants. In: Proceedings of 4th International AAAI Conference on Weblogs and Social Media, ICWSM '10 (2010)
38. Mishne, G.: Multiple ranking strategies for opinion retrieval in blogs. In: Online Proceedings of TREC. Citeseer (2006)
39. Moghaddam, S., Ester, M.: 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, pp. 1825–1828. ACM, New York, NY, USA (2010). DOI 10.1145/1871437.1871739. URL http://doi.acm.org/10.1145/1871437.1871739
40. Na, S.H., Lee, Y., Nam, S.H., Lee, J.H.: Improving opinion retrieval based on query-specific sentiment lexicon. In: Advances in Information Retrieval, pp. 734–738. Springer (2009)
41. Nigam, K., Hurst, M.: Towards a robust metric of opinion. In: AAAI Spring Symposium on Exploring Attitude and Affect in Text, pp. 598–603 (2004)
42. O'Mahony, M.P., Smyth, B.: Learning to recommend helpful hotel reviews. In: Proceedings of the 3rd ACM Conference on Recommender Systems, RecSys '09. New York, NY, USA (2009)
43. O'Mahony, M.P., Smyth, B.: A classification-based review recommender. Knowledge-Based Systems **23**(4), 323–329 (2010)
44. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up?: Sentiment classification using machine learning techniques. In: Proceedings of the 2nd ACL Conference on Empirical Methods in Natural Language Processing - Volume 10, EMNLP '02, pp. 79–86. Association for Computational Linguistics, Stroudsburg, PA, USA (2002). DOI 10.3115/1118693.1118704. URL http://dx.doi.org/10.3115/1118693.1118704
45. Paul, M.J., Zhai, C., Girju, R.: Summarizing contrastive viewpoints in opinionated text. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP '10, pp. 66–76. Association for Computational Linguistics, Stroudsburg, PA, USA (2010). URL http://dl.acm.org/citation.cfm?id=1870658.1870665
46. Pazzani, M., Billsus, D.: Content-based recommendation systems. In: P. Brusilovsky, A. Kobsa, W. Nejdl (eds.) The Adaptive Web, *Lecture Notes in Computer Science*, vol. 4321, pp. 325–341. Springer Berlin Heidelberg (2007). DOI 10.1007/978-3-540-72079-9_10. URL http://dx.doi.org/10.1007/978-3-540-72079-9_10
47. Pazzani, M., Billsus, D.: Content-based recommendation systems. In: The Adaptive Web, *Lecture Notes in Computer Science*, vol. 4321, pp. 325–341. Springer Berlin Heidelberg (2007)
48. Phelan, O., McCarthy, K., Smyth, B.: Using twitter to recommend real-time topical news. In: Proceedings of the 3rd ACM conference on Recommender systems, pp. 385–388. ACM (2009)
49. Poirier, D., Tellier, I., Fessant, F., Schluth, J.: Towards text-based recommendations. In: Adaptivity, Personalization and Fusion of Heterogeneous Information, RIAO '10, pp. 136–137. Paris, France (2010). URL http://dl.acm.org/citation.cfm?id=1937055.1937089
50. Popescu, A.M., Etzioni, O.: Extracting product features and opinions from reviews. In: Natural Language Processing and Text Mining, pp. 9–28. Springer London (2007)
51. Qiu, G., Liu, B., Bu, J., Chen, C.: Expanding domain sentiment lexicon through double propagation. In: Proceedings of the 21st International Joint Conference on Artificial Intelligence, *IJCAI '09*, vol. 9, pp. 1199–1204 (2009)
52. Reilly, J., McCarthy, K., McGinty, L., Smyth, B.: Dynamic critiquing. Advances in Case-Based Reasoning pp. 37–50 (2004)
53. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: Grouplens: An open architecture for collaborative filtering of netnews. In: Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work, CSCW '94, pp. 175–186. ACM, New York, NY, USA (1994). DOI 10.1145/192844.192905. URL http://doi.acm.org/10.1145/192844.192905
54. Reyes, A., Rosso, P.: Making objective decisions from subjective data: Detecting irony in customer reviews. Decision Support Systems **53**(4), 754–760 (2012)
55. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th International Conference on World Wide Web, WWW '01, pp. 285–295. ACM, New York, NY, USA (2001). DOI 10.1145/371920.372071. URL http://doi.acm.org/10.1145/371920.372071

56. Shardanand, U., Maes, P.: Social information filtering: algorithms for automating ẅord of mouth.̈ In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 210–217. ACM Press/Addison-Wesley Publishing Co. (1995)
57. Smyth, B.: Case-based recommendation. In: P. Brusilovsky, A. Kobsa, W. Nejdl (eds.) The Adaptive Web, *Lecture Notes in Computer Science*, vol. 4321, pp. 342–376. Springer Berlin Heidelberg (2007). DOI 10.1007/978-3-540-72079-9_11. URL http://dx.doi.org/10.1007/978-3-540-72079-9_11
58. Tata, S., Di Eugenio, B.: Generating fine-grained reviews of songs from album reviews. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, ACL '10, pp. 1376–1385. Association for Computational Linguistics, Stroudsburg, PA, USA (2010). URL http://dl.acm.org/citation.cfm?id=1858681.1858821
59. Tintarev, N., Masthoff, J.: Effective explanations of recommendations: User-centered design. In: Proceedings of the 1st ACM Conference on Recommender Systems, RecSys '07, pp. 153–156. ACM, New York, NY, USA (2007). DOI 10.1145/1297231.1297259. URL http://doi.acm.org/10.1145/1297231.1297259
60. Tsur, O., Davidov, D., Rappoport, A.: Icwsm - a great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews. In: Proceedings of the International AAAI Conference on Weblogs and Social Media (2010). URL http://www.aaai.org/ocs/index.php/ICWSM/ICWSM10/paper/view/1495/1851
61. Tumasjan, A., Sprenger, T.O., Sandner, P.G., Welpe, I.M.: Predicting elections with twitter: What 140 characters reveal about political sentiment. In: Proceedings of 4th International AAAI Conference on Weblogs and Social Media, ICWSM '10 (2010)
62. Turney, P.D.: Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, ACL '02, pp. 417–424. Association for Computational Linguistics, Stroudsburg, PA, USA (2002). DOI 10.3115/1073083.1073153. URL http://dx.doi.org/10.3115/1073083.1073153
63. Wiebe, J.M., Bruce, R.F., O'Hara, T.P.: Development and use of a gold-standard data set for subjectivity classifications. In: Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics, ACL '99, pp. 246–253. Association for Computational Linguistics, Stroudsburg, PA, USA (1999). DOI 10.3115/1034678.1034721. URL http://dx.doi.org/10.3115/1034678.1034721
64. Yessenalina, A., Yue, Y., Cardie, C.: Multi-level structured models for document-level sentiment classification. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP '10, pp. 1046–1056. Association for Computational Linguistics, Stroudsburg, PA, USA (2010). URL http://dl.acm.org/citation.cfm?id=1870658.1870760
65. Zhai, Z., Liu, B., Xu, H., Jia, P.: Clustering product features for opinion mining. In: Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, WSDM '11, pp. 347–354. ACM, New York, NY, USA (2011). DOI 10.1145/1935826.1935884. URL http://doi.acm.org/10.1145/1935826.1935884
66. Zhang, K., Narayanan, R., Choudhary, A.: 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). URL http://dl.acm.org/citation.cfm?id=1863190.1863201
67. Zhang, M., Ye, X.: A generation model to unify topic relevance and lexicon-based sentiment for opinion retrieval. In: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 411–418. ACM (2008)
68. Zhang, W., Jia, L., Yu, C., Meng, W.: Improve the effectiveness of the opinion retrieval and opinion polarity classification. In: Proceedings of the 17th ACM Conference on Information and Knowledge Management, pp. 1415–1416. ACM (2008)
69. Zhang, Z., Varadarajan, B.: Utility scoring of product reviews. In: Proceedings of the 15th ACM International Conference on Information and Knowledge Management, CIKM '06, pp. 51–57. ACM, New York, NY, USA (2006). DOI 10.1145/1183614.1183626. URL http://doi.acm.org/10.1145/1183614.1183626
70. Zhuang, L., Jing, F., Zhu, X.Y.: Movie review mining and summarization. In: Proceedings of the 15th ACM International Conference on Information and Knowledge Management, CIKM '06, pp. 43–50. ACM, New York, NY, USA (2006). DOI 10.1145/1183614.1183625. URL http://doi.acm.org/10.1145/1183614.1183625