Case-Studies in Mining User-Generated Reviews for Recommendation

Ruihai Dong, Michael P. O'Mahony, Kevin McCarthy and Barry Smyth

Abstract User-generated reviews are now plentiful online and they have proven to be a valuable source of real user opinions and real user experiences. In this chapter we consider recent work that seeks to extract topics, opinions, and sentiment from review text that is unstructured and often noisy. We describe and evaluate a number of practical case-studies for how such information can be used in an information filtering and recommendation context, from filtering helpful reviews to recommending useful products.

1 Introduction

User-generated reviews are now a common feature of online sites and stores. They have proven to be an important source of user opinions on products and services from books and movies to accommodation, people, and electronics. In fact, user-generated reviews and now considered by many to be a vital part of how users inform themselves, especially when it comes to purchasing behaviour. It is largely accepted that availability of reviews helps shoppers to choose [1] and increases the likelihood that they will make a buying decision [2], for example.

In this chapter we are interesting in automatically mining valuable opinion information from this plentiful but unstructured, and often noisy, source of user

R. Dong (✉)
CLARITY: Centre for Sensor Web Technologies, University College Dublin, Dublin, Ireland
e-mail: ruihai.dong@ucd.ie

M.P. O'Mahony · K. McCarthy · B. Smyth
Insight Centre for Data Analytics, University College Dublin, Dublin, Ireland
e-mail: michael.omahony@insight-centre.org

K. McCarthy
e-mail: kevin.mccarthy@insight-centre.org

B. Smyth
e-mail: barry.smyth@insight-centre.org

M.M. Gaber et al. (eds.), *Advances in Social Media Analysis*,
Studies in Computational Intelligence 602,
DOI 10.1007/978-3-319-18458-6_6

knowledge. We do this primarily by using shallow natural language processing, topic mining, and sentiment analysis techniques and demonstrate how the resulting information can be applied in a variety of information filtering and recommendation tasks. We begin by reviewing a representative sample of the state of the art in opinion mining with a particular focus on user-generated reviews. Next we describe our approach to topic extraction and sentiment analysis that is at the heart of our opinion mining method. We describe a series of case-studies to demonstrate some of the ways that the topics and opinions extracted from user-generated reviews can be applied in practice. For example, in the first case-study we look at the familiar task of classifying reviews and predicting review helpfulness [3–10] to demonstrate how an opinion-mining approach can offer some advantage over conventional alternatives. Following on, our second case-study describes a straight forward technique for recommending informative reviews to users based on our ability to accurately predict review helpfulness. Finally, in our third case-study we move from dealing with single reviews to using a collection of product reviews as a new source of product information. We describe and evaluate a product recommender that harnesses product descriptions that are formed exclusively from the opinions found in user reviews and show how this approach provides a novel basis to generate recommendations.

2 Related Work

Recent research highlights how online product reviews can influence the purchasing behaviour of users; see [1, 2]. The effect of consumer reviews on book sales on Amazon.com and Barnesandnoble.com [11] shows that the relative sales of books on a site correlates closely with positive review sentiment; although interestingly, there was insufficient evidence to conclude that retailers themselves benefit from making product reviews directly available to consumers; see also the work of [12, 13] for music and movie sales, respectively. As a result researchers have begun to focus on harnessing this type of user-generated content and there are two areas of related work particularly relevant to the research presented in this chapter: classifying reviews and extracting opinions from reviews.

2.1 Classifying User-Generated Reviews

As review volume has grown retailers recognise the need to develop ways to help users find high quality reviews for products of interest and to avoid malicious or biased reviews. This has led to a body of research focused on classifying or predicting review helpfulness, and also research on detecting so-called *spam reviews*.

Review helpfulness classification approaches, such as that proposed in [3], typically consider features related to the ratings, structural, syntactic, and semantic properties of reviews and have found ratings and review length among the most

discriminating. Reviewer expertise was found to be a useful predictor of review helpfulness in [4], confirming, in this case, the intuition that people interested in a certain genre of movies are likely to pen high quality reviews for similar genre movies. Review timeliness was also found to be important since review helpfulness declined as time went by. Furthermore, opinion sentiment has been mined from user reviews to predict ratings and helpfulness in services such as TripAdvisor [5–8].

Just as it is useful to automate the filtering of helpful reviews it is also important to identify malicious or biased reviews. These reviews can be well written and informative and so can appear to be helpful. However these reviews often adopt a biased perspective that is designed to help or hinder sales of the target product [9]. Li et al. [10] describe a machine learning approach to spam detection that is enhanced by information about the spammer's identity as part of a two-tier, co-learning approach. On a related topic, network analysis techniques are used in [14] to identify recurring spam in user-generated comments associated with YouTube videos by identifying discriminating comment *motifs* that are indicative of spambots.

2.2 Mining Opinions and Features from User-Generated Reviews

There have also been a number of efforts focused on the extraction of feature-based product descriptions from user reviews. The work in [15] is representative in this regard and describes the use of shallow natural language processing (NLP) techniques for explicit feature extraction and sentiment analysis; see also [16, 17]. The features extracted, and the techniques used, are similar to those presented in this chapter, 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. In [18], the sentiment of comparative and subjective sentences in reviews is analysed on a per-feature basis to create a semi-order of products, but the recommendation task with respect to a query product is not considered.

In this chapter we are particularly interested in product recommendation and the ability of review opinions to inform the recommendation process. Conventional recommender systems are either based on ratings or transaction data (collaborative filtering) or on fixed content representations (content-based filtering), and the idea of developing a recommendation framework based purely on noisy user-generated content remains novel in itself. The work in [19] 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 [20] has also tried to exploit user-generated content in similar ways. Likewise, reviews are leveraged to alleviate the well-known cold-start problem associated with collaborative recommenders [21]. In this 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.

3 Topic Extraction and Sentiment Analysis from User-Generated Reviews

The main focus of this chapter is how topics and sentiment mined from user-generated product reviews can be leveraged as the basis for new approaches to product filtering and recommendation. Before we describe how this topical and sentiment information can be used in practice, the approach to automatically extract topics and assign sentiment is first described; see Fig. 1 for an overview of this approach.

It is worth highlighting that the approach uses a combination of existing techniques from the literature; no novel techniques are presented. Rather, the main interest lies is the novel ways in which the extracted information can be applied to the filtering and recommendation of products as described in the case-studies that follow.

3.1 Topic Extraction

We consider two basic types of topics—*bi-grams* and *single nouns*—which are extracted using a combination of shallow NLP and statistical methods, primarily by combining ideas from [16, 22]. In the pre-processing step, we use OpenNLP[1] to

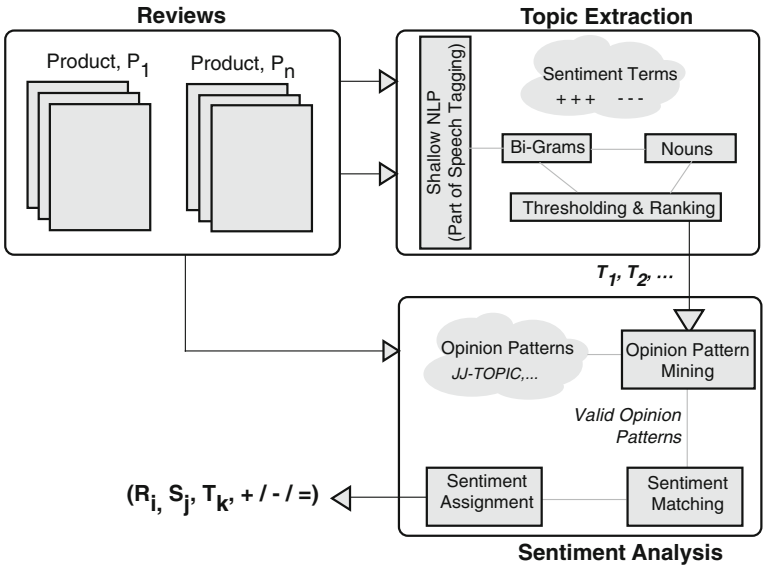

Fig. 1 System architecture for extracting topics and associated sentiment from user generated reviews

[1] OpenNLP: http://incubator.apache.org/opennlp/.

split reviews into sentences and label each term in a sentence with its appropriate part of speech, such as *NNS* (Noun, plural), *JJ* (Adjective), *VB* (Verb, base form) etc. Then, all terms in sentences are converted to lowercase and stemmed to root form, and in our method, stop words are excluded. To produce a set of bi-gram topics, all bi-grams from the global sentence set are extracted which conform to one of two basic part-of-speech co-location patterns: (1) an adjective followed by a noun (AN), such as *wide angle*; and (2) a noun followed by a noun (NN), such as *video mode*. These are candidate topics that need to be filtered to avoid including AN's that are actually opinionated single-noun topics; for example, *excellent lens* is a single-noun topic (*lens*) and not a bi-gram topic. Thus, bi-grams whose adjective is found to be a sentiment word (e.g. *excellent, good, great, lovely, terrible, horrible* etc.) are excluded using the sentiment lexicon proposed in [17].

To identify the single-noun topics we extract a candidate set of (non stop-word) nouns from the global review set. Often these single-noun candidates will not make for good topics; for example, they might include words such as *family* or *day* or *vacation*. A solution for validating such topics is proposed in [23] by eliminating those that are rarely associated with opinionated words. The intuition is that nouns that frequently occur in reviews and that are frequently associated with sentiment rich, opinion laden words are likely to be product topics that the reviewer is writing about, and therefore represent valid topics. Thus, for each candidate single-noun, how frequently it appears with nearby words from a list of sentiment words (using Hu and Liu's sentiment lexicon as above) is calculated, keeping the single-noun only if this frequency is greater than some threshold (in this case 30 %).

The result is a set of bi-gram and single-noun topics which is further filtered based on their frequency of occurrence in the review set, keeping only those topics $(T_1, \ldots, T_m)$ that occur in at least k reviews out of the total number of n reviews; by experiment, $k_{bg} = n/20$ is used for bi-gram topics and $k_{sn} = 10 \times k_{bg}$ for single noun topics.

3.2 Sentiment Analysis

To determine the sentiment of the topics in the product topic set, a method similar to the *opinion pattern mining* technique [24] is used for extracting opinions from unstructured product reviews. Once again the sentiment lexicon from [17] is used as the basis for this analysis. For a given topic T_i, and corresponding review sentence S_j from review R_k (that is the sentence in R_k that includes T_i), any sentiment words in S_j are identified. If there are none then this topic is marked as *neutral* from a sentiment perspective. If sentiment words $(w_1, w_2, \ldots)$ are present, that sentiment word (w_{min}) which has the minimum word-distance to T_i is identified.

Next the part-of-speech tags for w_{min}, T_i and any words that occur between w_{min} and T_i are determined. The POS sequence corresponds to an opinion pattern. For example, in the case of the bi-gram topic *noise reduction* and the review sentence

"*...this camera has great noise reduction...*", w_{min} is the word "*great*" which corresponds to the opinion pattern *JJ-TOPIC* as per [24].

Once an entire pass of all topics has been completed, the frequency of all opinion patterns that have been recorded is computed. A pattern is deemed to be valid (from the perspective of our ability to assign sentiment) if it occurs more than the average number of occurrences over all patterns [24]. For valid patterns sentiment is assigned based on the sentiment of w_{min} and subject to whether S_j contains any negation terms within a 4-word-distance[2] of w_{min}. If there are no such negation terms then the sentiment assigned to T_i in S_j is that of the sentiment word in the sentiment lexicon. If there is a negation word then this sentiment is reversed. If an opinion pattern is deemed not to be valid (based on its frequency) then a *neutral* sentiment is assigned to each of its occurrences within the review set.

4 Case-Study 1: Predicting Review Helpfulness

In the previous section an approach to automatically mine topics $(T_1, \ldots, T_m)$ and associated sentiment from review texts was described. Thus, each review R_i can be associated with *sentiment tuples*, $(R_i, S_j, T_k, +/-/=)$, corresponding to a sentence S_j containing topic T_k with a sentiment value positive (+), negative (−), or neutral (=). This approach forms the basis of a number of case-studies to explore how to harness user-generated reviews in various recommendation and recommendation-related tasks. To begin, in this first case-study, the task of classifying helpful reviews is examined, based on a variety of classification features, including the topical and sentiment features described above. The key question that will be explored is whether these topical and sentiment features add value relative to traditional features used in review classification.

4.1 Classifying Helpful Reviews

To build a classifier for predicting review helpfulness, a supervised machine learning approach is adopted. In the data that is available to us each review has a helpfulness score that reflects the percentage of positive votes that it has received, if any. Following the approach described in [8], a review is labeled as *helpful* if and only if it has a helpfulness score in excess of 0.75. All other reviews are labeled as *unhelpful*.

To represent review instances, a standard feature-based encoding is used based on a set of 7 different types of features, including temporal information (AGE), rating

[2] In long sentences, users may comment on multiple features. Thus, we introduce a window size for negation terms to limit their scope to nearby features. Based on experiment, we set the window size to four. Moreover, we identify certain phrases (e.g. "not only") which are not considered from a sentiment perspective. We acknowledge that more sophisticated sentiment analysis techniques have been proposed, an investigation of which we leave to future work.

information (RAT), simple sentence and word counts ($SIZE$), topical coverage (TOP), sentiment information ($SENT$), readability metrics ($READ$), and content made up of the top 50 most popular topics extracted from the reviews (CNT). These different types, and the corresponding individual features are summarised in Table 1. Some of these features, such as rating, word and sentence length, date and readability have been considered in previous work [3, 4, 26] and reflect best practice in the field of review classification. However, the topical and sentiment features (explained in detail below) are novel, and the comparison of the performance of the different feature sets is intended to demonstrate the efficacy of these new features, in isolation and combination, and in comparison to classical benchmarks across a common dataset and experimental configurations.

4.2 From Topics and Sentiment to Classification Features

As described above, a set of topics ($topics(R_k) = T_1, T_2, \ldots, T_m$) and corresponding sentiment scores (*pos/neg/neutral*) is assigned to each review R_k, which can be considered in isolation and/or in aggregate as the basis for classification features. For example, information about a review's *breadth* and *depth* of topic coverage can be obtained by simply counting the number of topics contained within the review and the average word count associated with the corresponding review sentences; see Eqs. 1 and 2. Similarly, the popularity of review topics, relative to the topics across the product as a whole, is given by Eq. 3, where $rank(T_i)$ is a topic's popularity rank for the product and $UniqueTopics(R_k)$ as the set of unique topics in a review. Thus, if a review covers many popular topics then it receives a higher $TopicRank$ score than if it covers fewer rarer topics.

$$Breadth(R_k) = |topics(R_k)| \tag{1}$$

$$Depth(R_k) = \frac{\sum_{\forall T_i \epsilon topics(R_k)} len(sentence(R_k, T_i))}{Breadth(R_k)} \tag{2}$$

$$TopicRank(R_k) = \sum_{\forall T_i \epsilon UniqueTopics(R_k)} \frac{1}{rank(T_i)} \tag{3}$$

Regarding sentiment, a variety of classification features can be derived: the number of positive (*NumPos* and *NumUPos*), negative (*NumNeg* and *NumUNeg*) and neutral (*NumNeutral* and *NumUNeutral*) topics (total and unique) in a review; the rank-weighted number of positive (*WPos*), negative (*WNeg*), and neutral (*WNeutral*) topics; the relative sentiment, positive (*RelUPos*), negative (*RelUNeg*), or neutral (*RelUNeutral*), of a review's topics. These features are all summarised in Table 1 under $SENT$.

Table 1 Classification feature sets

Type	Feature	#	Description
AGE	*Age*	1	The number of days since the review was posted
RAT	*NormUserRating*	1	A normalised rating score obtained by scaling the user's rating into the interval [0, 1]
SIZE	*NumSentences*	1	The number of sentences in the review text
	NumWords	1	The total number of words in the review text
TOP	*Breadth*	1	The total number of topics mined from the review
	Depth	1	The average number of words per sentence containing a mined topic
	Redundancy	1	The total word-count of sentences that are not associated with any mined topic
	TopicRank	1	The sum of the reciprocal popularity ranks for the mined topics present; popularity ranks are calculated across the target product
SENT	*NumPos* (*Neg*, *Neutral*)	3	The number of positive, negative, and neutral topics, respectively
	Density	1	The percentage of review topics associated with non-neutral sentiment
	NumUPos (*Neg*, *Neutral*)	3	The number of *unique* topics with positive/negative/neutral sentiment
	WPos (*Neg*, *Neutral*)	3	The number of positive, negative, and neutral topics, weighted by their reciprocal popularity rank
	RelUPos (*Neg*, *Neutral*)	3	The relative proportion of unique positive/negative/neutral topics
	SignedRatingDiff	1	The value of *RelUPos* minus *NormUserRating*
	UnsignedRatingDiff	1	The absolute value of *RelUPos* minus *NormUserRating*
READ	*NumComplex*	1	The number of 'complex' words (3 or more syllables) in the review text
	SyllablesPerWord	1	The average number of syllables per word
	WordsPerSen	1	The average number of words per sentence
	GunningFogIndex	1	The number of years of formal education required to understand the review
	FleschReadingEase	1	A standard readability score on a scale from 1 (30—very difficult) to 100 (70—easy)
	KincaidGradeLevel	1	Translates FleschReadingEase into KincaidGradeLevel required (U.S. grade level)
	SMOG	1	Simple Measure of Gobbledygood (SMOG) estimates the years of education required, see [25]
CNT		50	The top 50 most frequent topics that occur in a particular product's reviews

Also considered is a measure of the relative *density* of opinionated (non-neutral sentiment) topics in a review (see Eq. 4) and a relative measure of the difference between the overall review sentiment and the user's normalized product rating, i.e. $SignedRatingDiff(R_k) = RelUPos(R_k) - NormUserRating(R_k)$; we also compute an unsigned version of this metric. The intuition behind the rating difference metrics is to note whether the user's overall rating is similar to or different from the positivity of their review content. Finally, as shown in Table 1, each review instance also encodes a vector of the top 50 most popular review topics (*CNT*), indicating whether it is present in the review or not.

$$Density(R_k) = \frac{|pos(topics(R_k))| + |neg(topics(R_k))|}{|topics(R_k)|} \tag{4}$$

4.3 Expanding Basic Features

Each of the basic features in Table 1 is calculated for a particular review. For example, the *breath* of review R_k may be 5, indicating that it covers 5 identified topics. Whether this represents a high or low value for the product in question in unclear, which may have tens or even hundreds of reviews written about it. For this reason, in addition to this basic feature value, 4 other variations are calculated as follows to reflect the distribution of its values across a particular product:

- The *mean* value for this feature across the set of reviews for the target product.
- The *standard deviation* of the values for this feature across the target product reviews.
- The *normalised* value for the feature based on the number of standard deviations above (+) or below (−) the mean.
- The *rank* of the feature value, based on a descending ordering of the feature values for the target product.

Accordingly most of the features outlined in Table 1 translate into 5 different actual features (the original plus the 4 variations) for use during classification. This is the case for every feature (30 in all) in Table 1 except for the content features (*CNT*). Thus each review instance is represented as a total set of 200 features ($(30 \times 5) + 50$ features).

4.4 Evaluation

Our hypothesis is that the topical and sentiment features will help when it comes to the automatic classification of user generated reviews, into *helpful* and *unhelpful* categories, by improving classification performance above and beyond more traditional

features (e.g. terms, ratings, readability etc.); see [3, 7]. This hypothesis is tested on real-world review data for a variety of product categories using a number of different classifiers.

4.4.1 Datasets and Methodology

The review data for this experiment was extracted from Amazon.com during October 2012; in total, 51,837 reviews for 1,384 unique products were collected. Reviews for 4 product categories—*Digital Cameras (DC), GPS Devices, Laptops, Tablets*—were considered and each was labeled as *helpful* or *unhelpful*, depending on whether their helpfulness score was above 0.75 or not, as described in Sect. 4.1. For the purpose of this experiment, all reviews included at least 5 helpfulness scores (to provide a reliable ground-truth) and the helpful and unhelpful sets were sampled so as to contain approximately the same number of reviews. Table 2 presents a summary of these data, per product type, including the average helpfulness scores across all reviews, and separately for helpful and unhelpful reviews.

Each review was processed to extract the classification features as described above. Here we are particularly interested in understanding the classification performance of different categories of features. In this case, 8 different categories are considered, *AGE, RAT, SIZE, TOP, SENT-1, SENT-2, READ, CNT*. Note, the sentiment features ($SENT$) are subdivided into into two groups *SENT-1* and *SENT-2*. The latter contains all of the sentiment features from Table 1 whereas the former excludes the ratings difference features (signed and unsigned) so that the influence of rating information (usually a powerful classification feature in its own right) within the sentiment feature-set can be better understood. Accordingly, corresponding datasets for each category (Digital Cameras, GPS Devices, Laptops and Tablets) were created in which the reviews were represented by a single set of features; for example, the *SENT-1* dataset consists of reviews (one set of reviews for each product category) represented according to the *SENT-1* features only.

For the purpose of this evaluation three commonly used classifiers were considered: *RF (Random Forest)*, *JRip* and *NB (Naïve Bayes)*, see [27]. In each case classification performance was evaluated in terms of the area under the ROC curve (AUC) using 10-fold cross validation.

Table 2 Filtered and balanced dataset statistics

Category	#Reviews	#Prod.	Avg. Helpfulness		
			Help.	Unhelp.	All
DC	3180	113	0.93	0.40	0.66
GPS Devices	2058	151	0.93	0.46	0.69
Laptops	4172	592	0.93	0.40	0.67
Tablets	6652	241	0.92	0.39	0.65

4.4.2 Results

The results are presented in Figs. 2, 3 and 5. In Figs. 2, 3 and 4 the AUC performance for each classification algorithm (RF, JRip, NB) is shown separately; each graph plots the AUC of one algorithm for the 8 different categories of classification features for each of the four different product categories (DC, GPS, Laptop, and Tablet). Figure 5 provides a direct comparison of all classification algorithms (RF, JRip, NB); here results for a classifier using all features combined are presented. AUC values in excess of 0.7 can be considered as useful from a classification performance viewpoint [28]. Overall it can be seen that RF tends to produce better classification performance across the various feature groups and product categories. Classification performance tends to be poorer for the GPS dataset compared to Laptop, Tablet, and DC.

Previous research indicates that ratings information proves to be particularly useful when it comes to evaluating review helpfulness; see [3]. It is not a surprise therefore to see our ratings-based features perform well, often achieving an AUC > 0.7 on their own. For example, in Fig. 2 an AUC of approximately 0.75 for the Laptop and Tablet datasets is achieved, compared to between 0.65 and 0.69 for GPS and DC, respectively. Other 'traditional' feature groups (AGE, SIZE, READ, and CNT) rarely achieve AUC scores > 0.7 across the product categories.

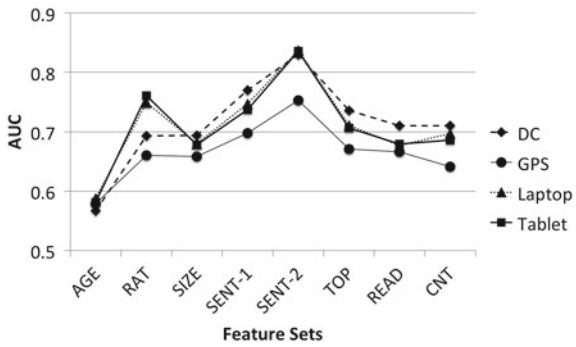

Fig. 2 Classification performance results for the RF classifier and different feature groups

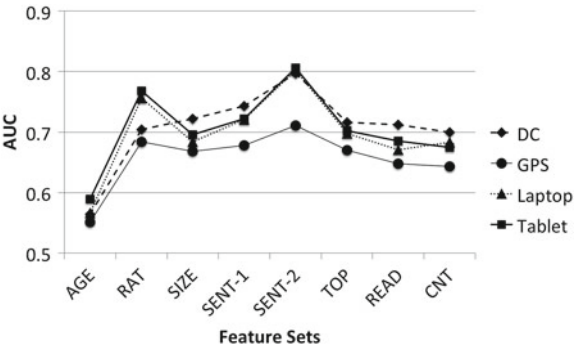

Fig. 3 Classification performance results for the JRip classifier and different feature groups

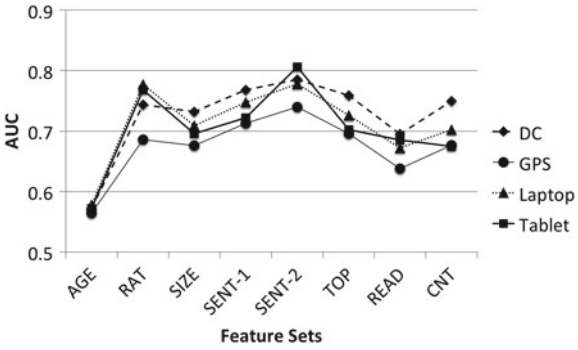

Fig. 4 Classification performance results for the NB classifier and different feature groups

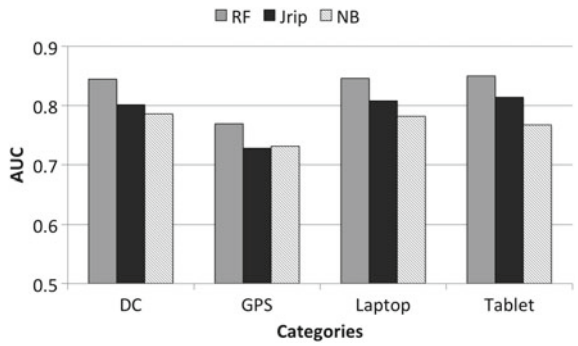

Fig. 5 Comparison of RF, JRip and NB for all features

Strong performance can be observed for the new topic and sentiment feature-sets proposed above. The *SENT-2* features consistently and significantly outperform all others, with AUC scores in excess of 0.7 for all three algorithms and across all four product categories; indeed in some cases the *SENT-2* features deliver AUC greater than 0.8 for DC, Laptop and Tablet products; see Fig. 2. The *SENT-2* feature group benefits from a combination of sentiment and ratings based features but a similar observation can be made for the sentiment-only features of *SENT-1*, which also achieve AUC greater than 0.7 for almost all classification algorithms and product categories. Likewise, the topical features (*TOP*) also deliver a strong performance with $AUC > 0.7$ for all product categories except for *GPS*.

These results bode well for a practical approach to review helpfulness prediction/classification, with or without ratings data. The additional information contained within the topical and sentiment features contributes to an uplift in classification performance, particularly with respect to more conventional features that have been traditionally used for review classification. In Fig. 5, summary classification results according to product category are presented when classifiers are trained using a combination of all feature types. Once again strong classification performance is achieved; for example, an AUC of more than 0.7 for all conditions is achieved and the *RF* classifier delivers an AUC close to 0.8 or beyond for all categories.

5 Case-Study 2: Recommending Helpful Reviews

On many e-commerce sites users are faced with having to sift through hundreds or even thousands of reviews, depending on the popularity of products. In the previous case-study we demonstrated that it is possible to accurately predict whether a given review is likely to be helpful or not. Given the review overload facing users it is worthwhile to consider taking this approach a step further: instead of classifying the helpfulness of a single review, can a review or set of reviews be identified for recommendation to a user, given their interest in a specific product? Hence in this case-study an approach to turning our review classifier into a review recommender is described.

5.1 From Helpfulness Classification to Review Recommendation

Amazon currently adopts a simple approach to review recommendation, by suggesting the most helpful positive and most helpful critical review from a review collection. Amazon collects review helpfulness feedback to support this form of review recommendation and as a criterion to rank reviews. But this approach is far from perfect. Many reviews (often a majority) have received very few or no helpfulness ratings. This is especially true for more recent reviews, which arguably may be more reliable in the case of certain product categories (e.g. hotel rooms). Moreover, if reviews are ranked by helpfulness then it is unlikely that users will see those yet to be rated, making it even less likely that they will attract ratings. It quickly becomes a case of "*the rich get richer*" for those early-rated helpful reviews.

Therefore, the motivation for this case study is to examine, in the absence of review helpfulness information, whether it is possible to make useful review recommendations. In Sect. 4 it was shown that reviews can be accurately classified as helpful or not, but what about identifying the *most* helpful review or a set of the most helpful reviews for a given product? In what follows, this question is considered by showing how the review classifier can be used to recommend helpful reviews to a user. In particular, classification confidence is used as the basis for the recommendation ranking. Thus, for a given product, the rank order of a recommended review is given by the classification confidence that the review is helpful.

5.2 Evaluation

In this experiment, review data for the 4 product categories—Digital Cameras (DC), GPS Devices, Laptops, Tablets—as described in Sect. 4.4.1 are used. For each product category, a 10 fold cross validation experimental methodology was used, such that each review for each product was associated with a classification confidence that the

review was helpful. The reviews for each product were then ranked by classification confidence and the top-ranked review was recommended; this approach is referred to as the *Pred* strategy. Recall this recommendation is made without the presence of actual helpfulness scores and relies only on ability to $predict$ whether a review will be helpful. In this experiment a random forrest (RF) classifier, based on all features described in Table 1, was used. As a simple baseline recommendation approach, a review was also selected at random (referred to as the *Rand* strategy).

The performance of these recommendation strategies can be evaluated in two ways. First, since the actual helpfulness scores of all reviews (the ground-truth) is known, the recommended review according to each strategy can be compared to the review which has the highest actual helpfulness score for each product, and averaged across all products in a given product category. Thus, the two line graphs in Fig. 6 plot the actual helpfulness of the recommended reviews (for *Pred* and *Rand*) as a percentage of the actual helpfulness of the most helpful review for each product; this is referred to as the *helpfulness ratio (HR)*. It can be seen that *Pred* significantly outperforms *Rand* delivering a helpfulness ratio of 0.9 and above compared to approximately 0.7 for *Rand*. This means that the *Pred* strategy is capable of recommending a review that has, on average, a helpfulness score which is 90 % that of the actual most helpful review.

Incidentally, very often the most helpful review has a perfect helpfulness score of 1.0 and this review is often recommended by *Pred*. In this regard, the recommendation performance of the *Pred* and *Rand* strategies can be further analysed by examining how often, on average, each strategy recommends a review for each product from among the top k reviews ranked by actual helpfulness. In Fig. 6, results for $k = 3$ are presented (as bars) for each product class. For instance, it can be seen that for Laptops *Pred* recommends a top-3 review 60 % of the time compared to only 37 % for *Rand*. Moreover, across all product categories, the *Pred* strategy recommends a top-3 review between 1.5 and 2 times as frequently as *Rand*.

In summary, the above findings indicate that the helpfulness classifier can be used to recommend helpful reviews, without the need for explicit helpfulness information, and that recommendation performance compares favourably to the optimal scenario in which recommendations are based on known helpfulness information. These

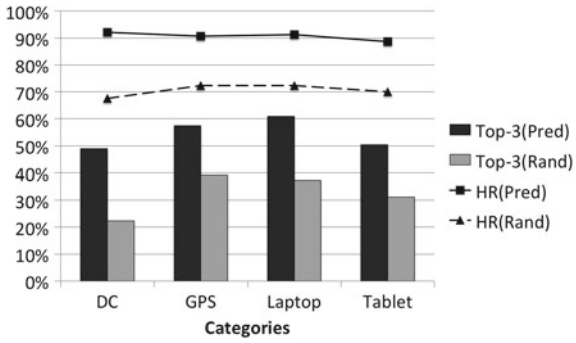

Fig. 6 The average helpfulness ratio and top-k results for *Pred* and *Rand* across all product categories

findings bode well for systems where review helpfulness is not available or is incomplete: it may still be possible to identify and recommend those reviews (new or old) which are likely to be genuinely helpful to users.

6 Case-Study 3: Mining Experiences and Recommending Products

Thus far, the focus of this chapter has been on user-generated reviews: their opinions, classification, and recommendation. In this case-study, however, the focus is changed from the reviews to the products being reviewed. After all, reviews exist because they reflect the experiences of users with real products and they are made available to users to help them chose a product for purchase. It makes sense therefore to consider whether the type of information mined from reviews, as described previously, can be aggregated at the level of individual products and used during classical product recommendation.

For instance, at the time of writing the listing for a *13" Retina MacBook Pro* on Amazon.com included a range technical features 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 system [29]. But in many domains such features are difficult to locate or are highly technical in nature, thereby limiting recommendation opportunities or making it difficult for casual consumers 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, many of which are far from technical; for example, its "*beautiful design*", its "*great video editing*" capabilities, and its "*high price*". These features capture more detail than a handful of technical (catalog) features and in this case-study these *experiential* features (and associated sentiment) are used to build alternative product descriptions for use in a product recommender; this case-study is based on a series of research papers and further detail can be found in [30–33].

6.1 From Reviews Topics to Product Features

The reviews for each product, P, are converted into a rich, feature-based description (or *product case*) using the techniques described in Sect. 3: unigram and bi-gram features are extracted from each product review and sentiment scores are assigned to these features.

Thus, for each product P we now have a set of features $F(P) = \{F_1, \ldots, F_m\}$ extracted from the reviews of P ($Reviews(P)$), 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 features which are

mentioned in $\geq 10\,\%$ of reviews for that product are only considered and overall sentiment (Eq. 5) and popularity (Eq. 6) scores are calculated; $Pos(F_i, P)$ (resp. $Neg(F_i, P)$, $Neut(F_i, P)$) denotes the number of times that feature F_i has positive (resp. negative, neutral) sentiment in the reviews for product P. The product case, $Case(P)$, is then given by Eq. 7.

$$Sent(F_i, P) = \frac{Pos(F_i, P) - Neg(F_i, P)}{Pos(F_i, P) + Neg(F_i, P) + Neut(F_i, P)} \tag{5}$$

$$Pop(F_i, P) = \frac{|\{R_k \in Reviews(P) : F_i \in R_k\}|}{|Reviews(P)|} \tag{6}$$

$$Case(P) = \{[F_i, Sent(F_i, P), Pop(F_i, P)] : F_i \in F(P)\} \tag{7}$$

6.2 Recommending Products

We will consider a *more-like-this* product recommendation setting in which the user is considering a particular product, Q, which serves as a *query product* for the purpose of recommendations, generating a set of suggestions for similar products. The above product representation leads to a content-based recommendation approach based on feature similarity to the query product. However, the availability of feature sentiment suggests another approach in which products that offer *better* quality features compared to the query product can be recommended.

6.2.1 Similarity-Based Recommendation

Each product case is represented as a vector of features, where feature *values* represent their popularity in reviews (Eq. 6) as a proxy for their importance. The cosine similarity between query product, Q, and candidate recommendation, C, is given by:

$$Sim(Q, C) = \frac{\sum_{F_i \in F(Q) \cup F(C)} Pop(F_i, Q) \times Pop(F_i, C)}{\sqrt{\sum_{F_j \in F(Q)} Pop(F_j, Q)^2} \sqrt{\sum_{F_j \in F(C)} Pop(F_j, C)^2}} \tag{8}$$

Using this approach, a set of top n recommendations are generated, ranked according to similarity with the query product [29].

6.2.2 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 9 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.

$$better(F_i, Q, C) = \frac{Sent(F_i, C) - Sent(F_i, Q)}{2} \tag{9}$$

Equation 10 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, Eq. 11 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.

$$B1(Q, C) = \frac{\sum_{F_i \in F(Q) \cap F(C)} better(F_i, Q, C)}{|F(Q) \cap F(C)|} \tag{10}$$

$$B2(Q, C) = \frac{\sum_{F_i \in F(Q) \cup F(C)} better(F_i, Q, C)}{|F(Q) \cup F(C)|} \tag{11}$$

6.2.3 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, Eq. 12 ranks recommendations based on their combined (controlled by w) similarity and sentiment with respect to Q; $Bx(Q, C)$ denotes $B1(Q, C)$ or $B2(Q, C)$, normalised to [0, 1].

$$Score(Q, C) = (1 - w)\ Sim(Q, C) + w \left(\frac{Bx(Q, C) + 1}{2} \right) \tag{12}$$

6.3 Evaluation

The above approaches are evaluated using data extracted from Amazon.com during October 2012. We considered 6 product domains in total but here present representative results for 3 domains (Table 3). For each product with ≥10 reviews, we extracted review texts, helpfulness information, and the top n (n = 5) recommendations for 'related' products as suggested by Amazon. In this case, related products

Table 3 Dataset statistics

Domain	#Reviews	#Products	$\#Features$ $\mu\ (\sigma)$	Sims $\mu\ (\sigma)$
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)

are those as suggested by Amazon's "customers who viewed this item also viewed these items" approach to recommendation.

6.3.1 Methodology and Metrics

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, a set of top-5 recommendations is generated using Eq. 12, 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 are compared to Amazon's recommendations for Q.

Amazon's overall product ratings are used as an independent measure of product quality. The *ratings benefit* metric compares two sets of recommendations based on their ratings (Eq. 13), where a ratings benefit of 0.1 means that sentiment-based recommendations (R) enjoy an average rating score that is 10 % higher that those produced by Amazon (A).

$$Ratings\ \ Benefit(R, A) = \frac{\overline{Rating(R)} - \overline{Rating(A)}}{\overline{Rating(A)}} \tag{13}$$

The *query product similarity* is also computed, given by the average similarity (by Eq. 8) based on mined feature representations between recommendations and the query product. This allows us to evaluate whether the sentiment-based techniques produce recommendations that are related to the query product and also provides a basis for comparison to Amazon's recommendations.

6.3.2 Mining Rich Product Descriptions

The success of our approach depends on its ability to translate user-generated reviews into useful product cases. Table 3 also 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 3 (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.

6.3.3 Sentiment Versus Similarity

For each domain, Fig. 7a–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). 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.

At $w = 0$, Eq. 12 is equivalent to a pure similarity-based approach to recommendation (i.e. using cosine by Eq. 8), 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, at least in terms of the features mentioned in reviews, than Amazon's own recommendations. For example, in the Phones domain (Fig. 7b) 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, a range of maximum positive ratings benefits (from 0.18 to 0.23) can be seen 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 (Fig. 7b) at $w = 1$, where ratings benefits of 0.11 and 0.21 are achieved 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.

6.3.4 Balancing Similarity and Sentiment

The relative contribution of similarity and sentiment is governed by w (Eq. 12). 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

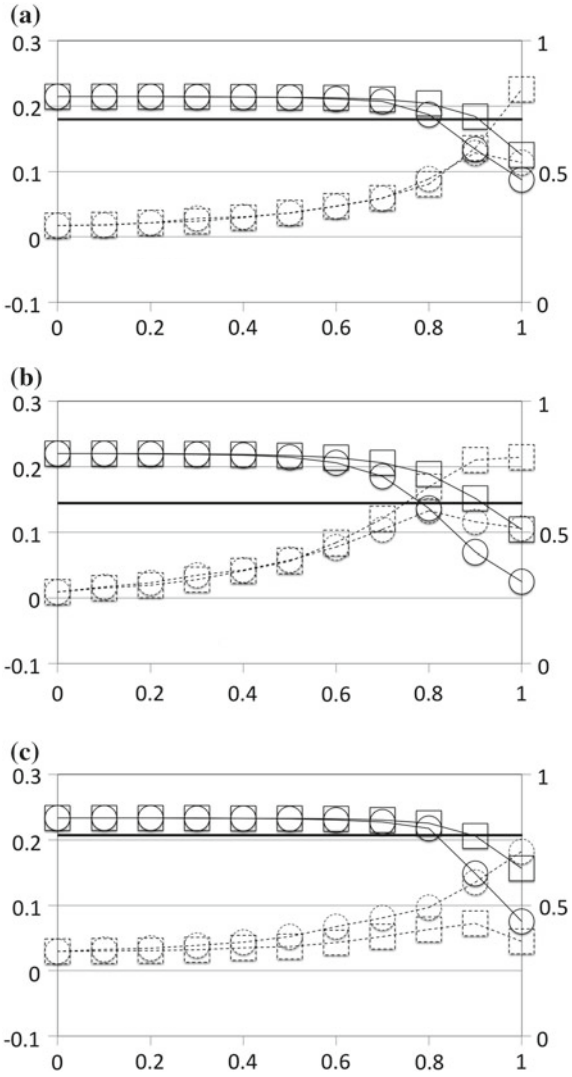

Fig. 7 Ratings benefit (*left* y-axis and *dashed* lines) and query similarity (*right* y-axis and *solid* lines) versus w (x-axis) for the Laptops (**a**), Phones (**b**) and GPS (**c**) domains. $B1$ and $B2$ are presented as *circles* and *squares* on the line graphs respectively and the Amazon query similarity is shown as a *solid horizontal line*

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, a reference point is needed for the purpose of a

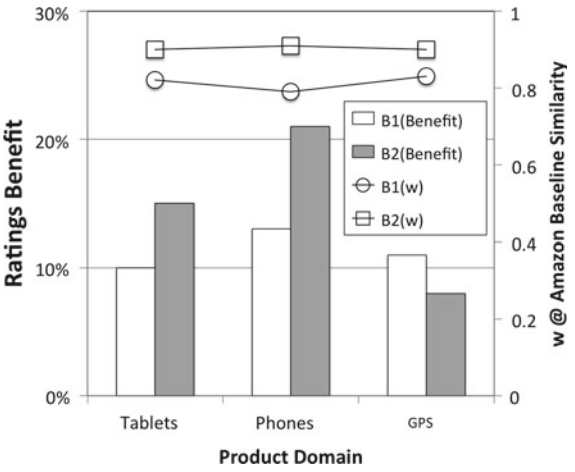

Fig. 8 Ratings benefits at Amazon baseline query product similarity

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 8 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 generalises to other domains is left to future work.

7 Conclusions

The web is awash with user-generated reviews, from the contemplative literary critiques of GoodReads to the flame wars that can sometimes engulf hotels on TripAdvisor. Reviews help consumers to choose and help online stores to convert browsers into buyers. In this chapter, a number of case-studies have been presented that focus on different ways to extract and harness the opinions contained in this valuable source of

user knowledge. Moreover, an approach to opinion mining that is well suited to user-generated reviews has been described, and a number of useful applications for the opinions that can be extracted, from the filtering and recommendation of individual reviews to a novel approach for product recommendation, have been demonstrated. In each case, the efficacy of the presented techniques have been evaluated using real-world review and product data.

Acknowledgments This work is supported by Science Foundation Ireland: through the CLARITY Centre for Sensor Web Technologies under grant number 07/CE/I1147; and through the Insight Centre for Data Analytics under grant number SFI/12/RC/2289.

References

1. Hu, N., Liu, L., Zhang, J.: Inf. Technol. Manag. **9**, 201 (2008)
2. Zhu, F., Zhang, X.M.: J. Market. **74**(2), 133 (2010)
3. Kim, S.M., Pantel, P., Chklovski, T., Pennacchiotti, M. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2006), pp. 423–430. Sydney, Australia (2006)
4. Liu, Y., Huang, X., An, A., Yu, X. In: Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008), pp. 443–452. IEEE Computer Society, Pisa, Italy (2008)
5. Baccianella, S., Esuli, A., Sebastiani, F. In: Advances in Information Retrieval, 31th European Conference on Information Retrieval Research (ECIR 2009), pp. 461–472. Springer, Toulouse, France (2009)
6. Hsu, C.F., Khabiri, E., Caverlee, J. In: Proceedings of the 2009 IEEE International Conference on Social Computing (SocialCom 2009), pp. 90–97. Vancouver, Canada (2009)
7. O'Mahony, M.P., Cunningham, P., Smyth, B. In: Proceedings of the 20th Irish Conference on Artificial Intelligence and Cognitive Science (AICS 2009), pp. 244–253. Dublin, Ireland (2009)
8. O'Mahony, M.P., Smyth, B. In: Proceedings of the 3rd ACM Conference on Recommender Systems, RecSys '09. New York (2009)
9. Lim, E.P., Nguyen, V.A., Jindal, N., Liu, B., Lauw, H.W. In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management, CIKM 2010, pp. 939–948. ACM, New York (2010). doi:10.1145/1871437.1871557. http://doi.acm.org/10.1145/1871437.1871557
10. Li, F., Huang, M., Yang, Y., Zhu, X. In: Proceedings of the 22nd International Joint Conference on Artificial Intelligence—Volume Volume Three, IJCAI 2011, pp. 2488–2493. AAAI Press, San Jose (2011). doi:10.5591/978-1-57735-516-8/IJCAI11-414. http://dx.doi.org/10.5591/978-1-57735-516-8/IJCAI11-414
11. Chevalier, J.A., Dina Mayzlin, D.: J. Market. Res. **43**(3), 345 (2006)
12. Dhar, V., Chang, E.A.: J. Interact. Market. **23**(4), 300 (2009)
13. Dellarocas, C., Zhang, M., Awad, N.F.: J. Interact. Market. **21**(4), 23 (2007)
14. O'Callaghan, D., Harrigan, M., Carthy, J., Cunningham, P. In: ICWSM (2012)
15. Popescu, A.M., Etzioni, O. In: Natural Language Processing and Text Mining, pp. 9–28. Springer, London, 2007. doi:10.1007/978-1-84628-754-12. http://dx.doi.org/10.1007/978-1-84628-754-1_2
16. Hu, M., Liu, B. In: Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '04, pp. 168–177. ACM, New York (2004). doi:10.1145/1014052.1014073. http://doi.acm.org/10.1145/1014052.1014073
17. Hu, M., Liu, B. In: Proceedings of the 19th National Conference on Artifical Intelligence, AAAI'04, pp. 755–760. AAAI Press, San Jose (2004). http://dl.acm.org/citation.cfm?id=1597148.1597269

18. Zhang, K., Narayanan, R., Choudhary, A. In: Proceedings of the 3rd Workshop on Online Social Networks, WOSN '10. Berkeley (2010). http://dl.acm.org/citation.cfm?id=1863190.1863201
19. Garcia Esparza, S., O'Mahony, M.P., Smyth, B. In: Proceedings of the 4th ACM Conference on Recommender Systems, RecSys '10, pp. 305–308. ACM, New York (2010). doi:10.1145/1864708.1864773. http://doi.acm.org/10.1145/1864708.1864773
20. De Francisci Morales, G., Gionis, A., Lucchese, C. In: Proceedings of the 5th ACM International Conference on Web Search and Data Mining, WSDM '12, pp. 153–162. ACM, New York (2012). doi:10.1145/2124295.2124315. http://doi.acm.org/10.1145/2124295.2124315
21. Poirier, D., Tellier, I., Fessant, F., Schluth, J. In: Adaptivity, Personalization and Fusion of Heterogeneous Information, RIAO '10, pp. 136–137. Paris, France (2010). doi:http://dl.acm.org/citation.cfm?id=1937055.1937089
22. Justeson, J.S., Katz, S.M.: Nat. Lang. Eng. **1**(1), 9 (1995)
23. Qiu, G., Liu, B., Bu, J., Chen, C. In: Proceedings of the 21st International Joint Conference on Artifical Intelligence, IJCAI 2009, pp. 1199–1204. Morgan Kaufmann Publishers Inc., San Francisco (2009). http://dl.acm.org/citation.cfm?id=1661445.1661637
24. Moghaddam, S., Ester, M. In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management, CIKM '10, pp. 1825–1828. ACM, New York (2010). doi:10.1145/1871437.1871739. http://doi.acm.org/10.1145/1871437.1871739
25. DuBay, W. Impact Information, pp. 1–76 (2004)
26. O'Mahony, M.P., Smyth, B. In: Adaptivity, Personalization and Fusion of Heterogeneous Information, RIAO 2010, pp. 164–167. Paris, France (2010). http://dl.acm.org/citation.cfm?id=1937055.1937097
27. Witten, I., Frank, E. Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, San Diego (2005)
28. Streiner, D., Cairney, J. The Canadian Journal of Psychiatry/La revue Canadienne de Psychiatrie (2007)
29. Pazzani, M., Billsus, D. In: The Adaptive Web, Lecture Notes in Computer Science, vol. 4321, pp. 325–341. Springer, Berlin, Heidelberg (2007). doi:10.1007/978-3-540-72079-910. http://dx.doi.org/10.1007/978-3-540-72079-9_10
30. Dong, R., Schaal, M., O'Mahony, M.P., McCarthy, K., Smyth, B. In: Proceedings of the 20th International Conference on Case-Based Reasoning (2012), ICCBR '12, pp. 62–76
31. Dong, R., Schaal, M., O'Mahony, M.P., McCarthy, K., Smyth, B. In: Proceedings of the 21st International Conference on Case-Based Reasoning, ICCBR '13, vol. 7969, pp. 44–58. Springer, Heidelberg (2013)
32. Dong, R., Schaal, M., O'Mahony, M.P., Smyth, B. In: Proceedings of the 23rd International Joint Conference on Artificial Intelligence, IJCAI '13, pp. 1310–1316. AAAI Press, Menlo Park, California (2013)
33. Dong, R., O'Mahony, M.P., Schaal, M., McCarthy, K., Smyth, B. In: Proceedings of the 7th ACM Conference on Recommender Systems, RecSys '13, pp. 411–414. ACM, New York (2013). doi:10.1145/2507157.2507199. http://doi.acm.org/10.1145/2507157.2507199