13

From Opinions to Recommendations

Michael P. O'Mahony and Barry Smyth

Insight Centre for Data Analytics
School of Computer Science
University College Dublin, Ireland

Abstract. Traditionally, recommender systems have relied on user preference data (such as ratings) and product descriptions (such as meta-data) as primary sources of recommendation knowledge. More recently, new sources of recommendation knowledge in the form of social media information and other kinds of user-generated content have emerged as viable alternatives. For example, services such as Twitter, Facebook, Amazon and TripAdvisor provide a rich source of user opinions, positive and negative, about a multitude of products and services. They have the potential to provide recommender systems with access to the fine-grained opinions of real users based on real experiences. This chapter will explore how product opinions can be mined from such sources and can be used as the basis for recommendation tasks. We will draw on a number of concrete case-studies to provide different examples of how opinions can be extracted and used in practice.

Key words: recommender systems, opinion mining, sentiment analysis

13.1 Introduction

Traditionally, recommender systems have relied on user preference data and product descriptions as the primary sources of recommendation knowledge. For example, collaborative recommendation approaches [8, 29, 60, 64, 66] rely on the former to identify a neighbourhood of like-minded users to a target user to act as a source of product recommendations (see also Chapter 10 of this book [38]). Alternatively, content-based recommendation approaches [46, 57, 67] select products for recommendation because they are similar to those that the target user has liked in the past (see also Chapter 12 of this book [7]). These approaches have worked well when suitable sources of recommendation knowledge is available, such as user-item ratings or item meta-data, but there are many circumstances where these approaches are less successful. For example, collaborative filtering systems work well when there are large communities of active users leading to rich user profiles to drive the recommendation process. But they are less successful when dealing with new users or where there is a sparsity of preference or ratings data. Content-based techniques are effective

when rich product descriptions are available but are less successful when more limited product information can be gathered.

One approach to dealing with the shortcomings of these conventional approaches has been to develop hybrid recommender systems that attempt to combine collaborative and content-based ideas. Such hybrid approaches [10] (see also Chapter 12 [7] of this book) are able to compensate for the short comings of any individual approach in isolation have have proven to be successful in practice. This hybridization approach is of course just one strategy for improving recommender system competence. In this paper we consider an alternative by harnessing new types of recommendation knowledge that is increasingly available online.

Recently novel, alternative sources of recommendation knowledge in the form of social media information (see Chapter 11 [40] of this book) and other kinds of user-generated content have emerged. For example, services such as Twitter, Facebook, Amazon and TripAdvisor provide a rich source of user opinions, positive and negative, about a multitude of products and services. This chapter will explore how product opinions can be mined from such sources and can be used as the basis for recommendation tasks. With this in mind we describe three related case-studies to describe different ways to extract and use this type of information in a recommendation context.

13.2 Sources of Recommendation Knowledge

Recommender systems have traditionally leveraged two sources of data — ratings or meta-data — in order to generate make suggestions to a target user[1]. Different algorithms have been developed to take advantage of these different types of data, offering different advantages, disadvantages, tradeoffs and compromsies; see [8, 29, 46, 57, 60, 64, 66, 67]. Indeed, some systems combine these data sources to offer hybrid approaches [10]. In this section we will briefly outline these conventional approaches to recommendation before exploring new sources of recommendation knowledge in the form of user-generated content.

13.2.1 Collaborative Filtering

The well-known collaborative filtering style of recommender system [8, 29, 60, 64, 66] relies on ratings data provided by users. Each item is associated with a set of user ratings and each user is profiled in terms of their item ratings. Effectively a collaborative filtering system starts with a *user-item ratings matrix* in which each user-item combination can be associated with a rating; although in practice these ratings matrices tend to be extremely sparsely populated because

[1] See Chapters 11 and 12 [40, 7] of this book for other examples of recommendation knowledge.

most users only rate a tiny fraction of available items. An extensive discussion of collaborative filtering recommender systems can be found in Chapter 10 of this book [38].

As described in Chapter 14 of this book [34], ratings can be explicit (directly provided by users) or implicit (inferred from user behaviour). For example, Netflix explicitly encourages users to rate movies on a 5-star scale. On the the other hand, ratings can be also inferred by interpreting various types of user behaviours from purchasing a product (a highly positive 'rating') or selecting a link for more product detail (a moderately positive 'rating') to eliminating a product from a list (a negative 'rating'). In each case the power of collaborative filtering stems from its ability to translate these item ratings into user recommendations by identifying users (or items) with similar ratings histories. In one form of collaborative filtering, *user-based collaborative filtering*, items are suggested for the target user because they have been liked by *other users* with similar rating histories [8, 29]. Alternatively, *item-based collaborative filtering* adopts a more item-centred perspective by suggesting items for the target user that have similar ratings histories to *other items* that the target user has liked [63].

Both user-based and item-based collaborative filtering approaches generate recommendations directly from the ratings matrix. Other approaches attempt to uncover latent factors that exist within the ratings space and use these as the basis for recommendation. For example, matrix factorisation approaches seek to identify latent features that are shared between items and users. They do this by factoring the user-item ratings matrix into separate user and item matrices which map users and items to a set of k latent features respectively [39]. Then a rating for item i by user u can be predicted by computing the dot product of the u^{th} column of the user matrix and the i^{th} row of the item matrix.

13.2.2 Content-Based & Hybrid Recommendation

In contrast to the ratings-based techniques of collaborative filtering, content-based recommenders leverage item meta-data in order to make recommendations. The meta-data for an item is typically composed of a set of descriptive features, keywords, or tags (see Chapter 12 of this book [7] for an extensive discussion on tag-based recommendation). For example, in a movie recommender a movie might be represented in terms of its genre, the lead actors, the director etc. Recommendations are generated by selecting items that are similar (based on meta-data) to those that the user has liked in the past; see [46, 57]. A number of variations of content-based techniques have been proposed including case-based recommendation [67], which relies on structured feature-based item descriptions, and textual recommenders, which use less structured item descriptions.

On their own collaborative filtering and content-based techniques have a number of pros and cons. The former work well in the absence of rich item meta-data,

for example, but require large, mature communities of users with extended ratings histories. The latter are less dependent on mature user communities to get started but do require detailed meta-data, which may be difficult or expensive to acquire. Collaborative filtering approaches have trouble recommending new items until such time as they have been rated by a minimum number of users, whereas content-based techniques can recommend new items from the outset. Other challenges exist when it comes to dealing with users with unusual tastes or generating diverse and novel recommendations. In response to these pros and cons, researchers have considered various ways to combine collaborative filtering and content-based approaches [7, 10].

Ratings-data, meta-data, and other forms of item content are widely used sources of recommendation data. However, the rise of the social web and the proliferation of user-generated content in the form of user reviews provides new opportunities for recommender systems research, and in this chapter we explore how this form of content can be used as a new type of recommendation data.

13.2.3 User-Generated Content for Recommendation

The rise of the so-called social web has seen an explosion in user-generated content, from short-form status updates to long-form reviews and blog posts. This content is typically noisy and unstructured but it has the potential to act as a rich source of user opinions about products and services. If we can mine these opinions then we may be able to harness them for a new form of recommender system (see also Chapter 5 of this book [21] for a discussion on social navigation). To do this researchers have been turning their attention to developing techniques for mining user-generated content to, for example, identify opinion sentiment, identify product features, and even combining sentiment and features to generate richly opinionated product descriptions and user profiles that can be used in recommendation.

One important focus for research has been the application of sentiment analysis techniques to user-generated content [69]. Sentiment analysis encompasses different areas such as sentiment classification [9, 48, 56], which seeks to determine whether the semantic orientation of a piece of text is positive or negative (and sometimes, neutral). In [59] it was demonstrated that these sentiment classification models can be topic-dependant, domain-dependant and temporally-dependant and suggested that training with data which contains emoticons can make these models more independent. Another area within sentiment analysis is subjectivity classification [73], which classifies text as subjective (i.e. it contains author opinions) or objective (i.e. it contains factual information). Ultimately the ability to understand the polarity and perspective of an opinion (positive or negative, subjective or objective) is a key enabling technology for opinion mining.

Increasingly, product reviews provide a rich source of user opinions and it is now common practice to research our purchases by reading reviews prior to making a final buying decision. Using natural language processing, opinion mining, and sentiment analysis techniques it is now possible to mine reviews to identify features that are being discussed and the precise nature of the discussion. Accordingly we can generate a much richer picture of a product or service by understanding how users feel about certain features or by identifying entirely new features that are unlikely to appear in a regular description of the product. So, for example, by mining a hotel review we might learn that the hotel has an excellent business centre and also realise that its restaurant serves a delicious eggs benedict. Much of the initial work in this area has focused on extracting features from electronic products such as cameras or MP3 players, where the set of product features is typically more restricted, hence representing a more tractable problem, compared to other domains such as movies or books. In more recent work [33], where feature extraction and opinion mining is performed on more complex (from a feature perspective) movie reviews, the authors first attempt to identify the set of key features that authors discuss by applying clustering techniques; a Latent Dirichlet Allocation approach was found to provide the best results. While some research [31, 58] applies feature extraction techniques, such as point-wise mutual information or feature-based summarisation in a domain-independent context, it is argued in [11] that a domain-dependent approach is preferable, leading to a more precise feature set, and describe an approach based on a taxonomy of the domain product features.

A methodology for building a recommender system by leveraging user-generated content is described in [74]. In this work, the authors propose a hybrid of a collaborative filtering and a content-based approach to recommend hotels and attractions, where the collaborative filtering component utilises the review text to compute user similarities in place of traditional preference-based similarity computations. Another early attempt to build a recommender system based on user-generated review data is described in [1]. In that work an ontology is used to extract concepts from camera reviews based on users' requests about a product; for example, "I would like to know if Sony361 is a good camera, specifically its interface and battery consumption". In this case, the features *interface* and *battery* are identified, and for each of them a score is computed according to the opinions (i.e. polarities) of other users and presented to the user. Similar ideas are presented in [2], which look at using user-generated movie reviews from IMDb in combination with movie meta-data (e.g. keywords, genres, plot outlines and synopses) as input for a movie recommender system. Their results show that user reviews provide the best source of information for movie recommendations, followed by movie genre data. Further, the authors in [71] leverage opinions mined from online reviews to enhance user preference models for use in collaborative recommender systems. Experiments indicate the approach outperforms baselines algorithms with respect to accuracy and recall.

While the research and techniques described above have focused primarily on long-form review text, recent work has also considered the analysis of short-form

reviews, such as micro-blog messages. For instance, Twitter messages are classified as positive, negative or neutral in [55] by creating two classifiers: a neutral-sentiment classifier and a polarity (negative or positive) classifier. Moreover, the effect of different attribute sets on sentiment classification for short-form and long-form reviews is compared in [6]. Results show that while classification accuracy for long-form reviews can benefit from using more complex attribute sets (for example, bigrams and POS tagging), this is not the case for short-form reviews where simpler attributes based on unigrams alone were sufficient from a performance perspective. Further, mining users' interests and hot topics from micro-blog posts have also been investigated in recent research [3, 5].

13.2.4 Review Filtering, Quality & Spam

While product reviews are undoubtedly useful from a recommendation and user profiling perspective, reviews can however vary greatly in their quality and helpfulness. For example, reviews can be biased or poorly authored, while others can be very balanced and insightful. For this reason, the ability to accurately identify helpful reviews would be a useful, albeit challenging, feature to automate. While some services are addressing this by allowing users to rate the helpfulness of each review, this type of feedback can be sparse and varied, with many reviews, particularly the more recent ones, failing to attract any feedback. Hence the need exists to develop automated approaches to classify review helpfulness.

In this regard, a significant body of work has been carried out on the classification of product review helpfulness. For example, one approach to review classification has been proposed in [37], which considered feature sets relating to the structural, lexical, syntactic, semantic and some meta-data properties of reviews. Of these features, score, review length and unigram (term distribution) were among the most discriminating. Reviewer expertise was found to be a useful predictor of review helpfulness in [44], capturing the intuition that people interested in a particular genre of movies are likely to author high quality reviews for movies within the same or related genres. Timeliness of reviews was also important, and it was shown that (movie) review helpfulness declined as time went by. The use of credibility indicators was proposed in [72] in relation to topical blog post retrieval. Some of the indicators considered were text length, the appropriate use of capitalisation and emoticons in the text, spelling errors, timeliness of posts and the regularity at which bloggers post; such indicators were found to significantly improve retrieval performance in this work. Research in relation to sentiment and opinion analysis [69] is also of interest in this regard. For example, the classification of reviews for sentiment using content-based feature sets was considered in [4], where a study based on TripAvisor reviews demonstrated the effectiveness of this approach. Additional related work can be found in [30, 52, 51].

The need to identify *malicious* or *biased* reviews has also been considered in recent times. Such reviews can be well written and informative and so appear to be helpful. However these reviews often adopt a biased perspective that is designed to help or hinder sales of a target product [43]. Thus, a number of approaches have been proposed in the literature to identify such biased reviews. For example, a machine learning approach to spam detection is described in [42] that is enhanced by information about the spammer's identify as part of a two-tier co-learning approach. On a related topic, network analysis techniques are used in [50] to identify recurring spam in user generated comments associated with YouTube videos; in this work discriminating comment *motifs* are identified that are indicative of spambots. For other work in this area, see for example [35, 36, 47, 54, 70].

In this chapter, we begin with two case-studies which focus on leveraging product reviews for recommendation. The first case-study (Section 13.3) presents a recommendation approach which is inspired by ideas from the area of information retrieval. In this approach, users and products are modelled using a bag of words approach, and user profiles act as queries against product indices to generate recommendation lists. The second case-study (Section 13.4) presents a more sophisticated approach in which user opinions expressed in reviews are mined to construct an experiential case representation for products. With this representation, products which are not only similar to, but are *better* than (from a sentiment perspective) previous liked items can be recommended to users. Finally, in the third case-study (Section 13.5), the problem of review helpfulness classification is discussed, and one approach from the literature to address this problem is described in detail.

13.3 Case Study 1 – Mining Recommendation Knowledge from Product Reviews

As mentioned above, a key issue with conventional collaborative and content-based recommenders is that oftentimes neither user ratings nor item meta-data are available in sufficient quantity to effectively drive either approach. In this case study, a third source of recommendation data — namely, user-generated content relating to products — is explored as the basis for an alternative content-based approach to recommendation. In particular, user and item profiles are constructed from product reviews and recommendations are made using traditional item representation, term weighing and similarity techniques from the area of information retrieval.

A significant challenge associated with this approach is the inherently noisy nature of product reviews. For example, while some reviews can be comprehensive and informative, others are overly brief, off-topic or biased. Nonetheless, product reviews are plentiful, and range from the long-form reviews found on sites such as TripAdvisor and Amazon to opinions expressed by users in short-

Fig. 13.1 A review of the movie *The Dark Night* from Blippr.

form on micro-blogging sites such as Twitter. In this case study, reviews from a Twitter-like service called Blippr are considered, where reviews are in the form of 160-character text posts. Figure 13.1 shows an example of a typical review posted on Blippr. In what follows, the review-based recommender proposed in [26] is described (see also [22, 25]), and an evaluation of the approach is presented which shows that comparable performance to more conventional recommendation approaches is achieved when applied to a range of product domains.

13.3.1 Review-based Recommendation Approach

In this section, the main steps of the approach, based on ideas from information retrieval, are described: (1) how users and products are represented and (2) how this representation is used for the purposes of recommendation. In addition, a benchmark approach, inspired by the collaborative filtering approach to recommendation, is described.

13.3.1.1 Index Creation Two indices, representing users and products, are created as the basis for the approach as follows.

Product Index. Consider a product P_i which is associated with a set of reviews, $Reviews(P_i) = \{r_1, ..., r_j\}$. In turn, each review r_u is made up of a set of terms, $Terms(r_u) = \{t_1, ..., t_v\}$. Thus, each product can be represented as a set of terms using a bag-of-words style approach [62] consisting of all the terms in the reviews associated with it as per Equation 13.1.

$$P_i = \{t \in r : r \in \mathit{Reviews}(P_i)\} \ . \tag{13.1}$$

In this way individual products can be viewed as documents made up of the set of terms (words) contained in their associated reviews. An index of these documents can be created such that documents (that is products) can be retrieved based on the terms that are present in their reviews. Moreover, terms that are associated with a given product can be *weighted* based on how representative or informative these terms are with respect to the product in question; here, the *term frequency–inverse document frequency* (TF-IDF) [62] and the BM25 (also referred to as *Okapi* weighting) [61] term-weighting schemes are considered. Briefly, in the case of the TD-IDF scheme (see Equation 13.2), the weight of a term t_j in a product P_i, with respect to some collection of products $\mathbf{P}$, is proportional to the frequency of occurrence of t_j in P_i (denoted by $tf(t_j, P_i)$), but inversely proportional to the frequency of occurrence of t_j in $\mathbf{P}$ overall, thereby giving preference to terms that help to discriminate P_i from the other products in the collection. For details regarding the BM25 scheme, see [26].

$$\text{TF-IDF}(P_i, t_j, \mathbf{P}) = \frac{\text{tf}(t_j, P_i)}{\sum_{t_k \in P_i} \text{tf}(t_k, P_i)} \times \text{idf}(t_j, \mathbf{P}) \ , \tag{13.2}$$

$$\text{idf}(t_j, \mathbf{P}) = \log\Big(\frac{|\mathbf{P}|}{|\{P_k \in \mathbf{P} : t_j \in P_k\}|}\Big) \ . \tag{13.3}$$

Thus a term-based index of products $\mathbf{P}$ can be created, such that each entry $\mathbf{P_{ij}}$ encodes the importance of term t_j in product P_i, where term weights are calculated according to TF-IDF (Equation 13.4) or BM25 (Equation 13.5).

$$\mathbf{P_{ij}} = \text{TF-IDF}(P_i, t_j, \mathbf{P}) \ . \tag{13.4}$$

$$\mathbf{P_{ij}} = \text{BM25}(P_i, t_j, \mathbf{P}) \ . \tag{13.5}$$

User Index. A similar approach to that above is used to create the user index. Specifically, each user U_i is represented as a document made up of the terms in their posted reviews as per Equation 13.6, where $\mathit{Reviews}(U_i)$ denotes the reviews posted by user U_i. As before, a user index, $\mathbf{U}$, consisting of all users is created, such that each entry $\mathbf{U_{ij}}$ encodes the importance of term t_j for user U_i, once again using the TF-IDF or BM25 weighting schemes as per Equations 13.7 and 13.8, respectively.

$$U_i = \{t \in r : r \in \mathit{Reviews}(U_i)\} \ . \tag{13.6}$$

$$\mathbf{U_{ij}} = \text{TF-IDF}(U_i, t_j, \mathbf{U}) \ . \tag{13.7}$$

$$\mathbf{U_{ij}} = \text{BM25}(U_i, t_j, \mathbf{U}) \ . \tag{13.8}$$

13.3.1.2 Product Recommendation In the above, two types of index for use in recommendation are described: an index of users and an index of products, based on the terms in their associated reviews. This suggests the following recommendation strategies. First, a *user-based* approach can be implemented in

```
Input: Target user U_T, user index U, product index P, number of
products to retrieve n
Output: Top n product recommendations

1.  UserBasedRecommendation (U_T, U, P, n)
2.  Begin
3.      query ← U.get(U_T)          // Return term vector for U_T in U
4.      recs ← P.retrieve(query)    // Retrieve ranked list of
                                    // products from P based on query
5.      return recs.first(n)        // Return top n recommendations
6.  End
```

Fig. 13.2 User-based recommendation algorithm.

which the *target user's* profile from the user index acts as a query against the product index to produce a ranked-list of similar products (the target user's reviews are first removed from the product index to ensure that no bias is introduced into the process); see Figure 13.2. Different variations of this approach can be considered by using different weighting schemes (TF-IDF and BM25) to index and query the index[2]. Further, term stemming can be applied to the data to improve the match between query and index terms.

In addition, to provide a benchmark for the above index-based approaches, a *community-based* approach based on collaborative filtering ideas [66] can be implemented. A set of similar users (or *neighbours*) is first identified, by using the target user profile as a query on the user index, and then the preferred products of these neighbours are ranked based on their frequency of occurrence in neighbour profiles; see Figure 13.3.

13.3.2 Evaluation

The recommendation performance provided by the review-based and benchmark algorithms described above is presented. The datasets used in the evaluation are first described, followed by the metrics used to measure performance.

13.3.2.1 Datasets The evaluation is based on reviews extracted from the Blippr service, which allows users to review products from a number of different domains. Reviews (or blips) are in the form of 160-character text messages, and users must also supply an accompanying rating on a 4-point rating scale: *love it*, *like it*, *dislike it* or *hate it*. Data was collected using the Blippr API in April 2010, capturing reviews written in the English language before that date. Preprocessing of reviews is performed, such as removing stopwords, special symbols (?,*,& etc.), digits and multiple repetitions of characters in words (e.g. *goooood* is reduced to *good*). Further, only reviews which have *love it* ratings are considered

[2] Lucene (`http://lucene.apache.org`) is used in the subsequently described experiments to provide the term-weighting and querying functionality.

Input: Target user U_T, user index **U**, product index **P**, number of products to retrieve n, neighbourhood size k
Output: Top n product recommendations

```
1.  CommunityBasedRecommendation (U_T, U, M, n, k)
2.  Begin
3.      query ← U.get(U_T)              // Return term vector for U_T in U
4.      users ← U.retrieve(query)       // Get ranked list of similar users
5.      neighs ← users.first(k)         // Get the top k most similar
                                        // users as neighbours
6.      recs ← {}                       // Get all neighbours' products
7.      for each n ∈ neighs
8.          recs ← recs ∪ n.products()  // Add products from current
                                        // neighbour to recommendation set
9.      end
10.     return recs.sort(score(.,.), n) // Return top n most frequently
11. end                                 // occurring products
                                        // score(P_i, neighs) = Σ_{n ∈ neighs} occurs(P_i, n),
                                        // where occurs(P_i, n) = 1 if P_i is
                                        // present in n and 0 otherwise
```

Fig. 13.3 Community-based recommendation algorithm.

Table 13.1 Evaluation dataset statistics.

	movies	*apps*	*books*	*games*
# Products	1,080	268	313	277
# Users	542	373	120	164
# Reviews	15,121	10,910	3,003	3,472

(i.e. where users have expressed the highest sentiment toward products) since we wish to recommend products which are actually liked by users. Note that reviews which express negative sentiment could also be considered to identify products which are disliked by users and which should not be recommended; however, such an approach is not examined here.

The experiments use Blippr data relating to four product types: *movies*, *books*, *applications* (*apps*) and *games*. Products with at least three reviews and users that have authored at least five reviews are selected. See Table 13.1 for dataset statistics.

13.3.2.2 Metrics *Precision* and *recall*, which have been widely used in the field of information retrieval, are used to evaluate recommendation accuracy. These metrics have been adapted to evaluate the accuracy of a set of recommended products [64] and are defined as follows:

$$\text{Precision} = \frac{|T \cap R|}{|R|} \ , \tag{13.9}$$

$$\text{Recall} = \frac{|T \cap R|}{|T|} \ , \tag{13.10}$$

where T and R are the test and recommended sets for each user, respectively.

We also evaluate recommendation *coverage*, which measures the number of products that a recommender is capable of making recommendations for (as a percentage of the total number of products in the system) [27]. In general, the ability of an algorithm to make recommendations for large numbers of (relevant) products is a desirable system property, so as to avoid situations in which only a limited number of items (e.g. popular items) are ever capable of being recommended.

13.3.2.3 Results To evaluate the recommendation algorithms, separate product and user indices are first created for each of the four datasets according to the approach described in Section 13.3.1. The main objective is to compare the performance of the user-based approach with that of the community-based benchmark. In the case of the user-based approach, the performance of two term-weighting schemes is considered: TF-IDF and BM25. Further, to determine if term stemming has any effect on the performance of the user-based approach, versions of TF-IDF weighting with (TF-IDF+) and without stemming (TF-IDF) are compared.

For each dataset, a leave-one-out approach is used where each user in turn acts as the target user (as per Section 13.3.1) and precision and recall scores are computed for different recommendation-list sizes ranging from 5 to 30 items. Results are presented in Figure 13.4 for the *movies* and *books* datasets. The results show that there is a clear benefit for the user-based recommendation strategies compared to the community-based approaches. For example, in the case of the *books* dataset using recommendation lists of size 5, the best user-based approach enjoys a precision of 0.44. In contrast, the best performing community-based approach (CB-10), where 10 similar users are selected as the basis for recommendation, achieves a precision of 0.32.

For all datasets, TF-IDF with and without stemming provide similar results; with stemming applied, TF-IDF performs marginally better for most datasets. For the larger datasets (*movies* and *apps*), the performance provided by BM25 is very close to that of TF-IDF, but is seen to fall off for the smaller datasets (*books* and *games*); see [26] for more details.

In Figure 13.5 (left), the precision and recall provided by the user-based approach using TF-IDF are compared across the four datasets. It can be seen that the best performance is achieved for the *apps* dataset where, for example, precision and recall values of 0.54 and 0.37 are achieved, respectively, compared to values of 0.42 and 0.29 for the *books* dataset (these values correspond to recommendation lists of size 5). Also shown in this figure is the mean number of reviews (blips) per product for each dataset; it can be seen that these values correlate well with the precision ($r = 0.84$) and recall ($r = 0.83$) performance achieved for the datasets. This seems a reasonable finding, since it indicates that richer product indices (i.e. products are described by a greater number of reviews) lead to better recommendation performance. However, since the datasets used in the evaluation contain short-form reviews and relatively small numbers of users and products, further analysis is required to draw general conclusions in this regard.

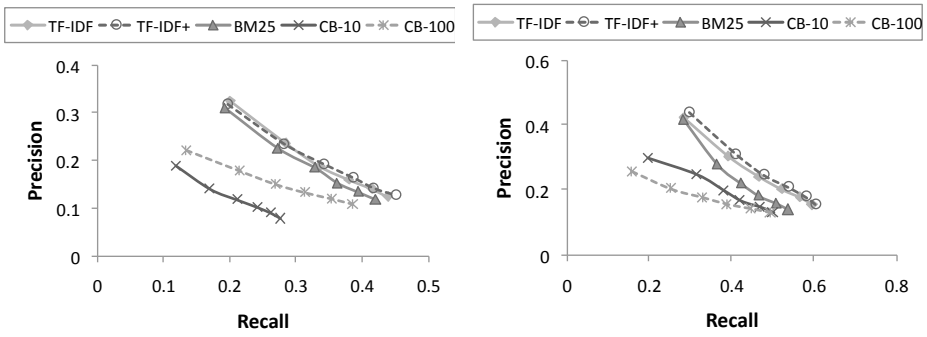


Fig. 13.4 Precision–recall for the *movies* (left) and *books* (right) datasets for user-based (TF-IDF vs. TF-IDF+ vs. BM25) and community-based (CB-10 vs. CB-100) recommendation.

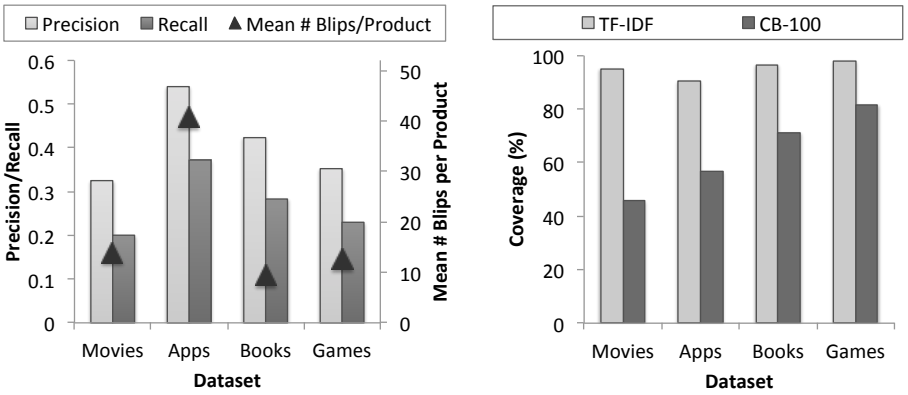


Fig. 13.5 Precision–recall (recommendation list sizes of 5) provided by user-based recommendation using TF-IDF and mean number of reviews (blips) per product vs. dataset (left) and coverage provided by recommendation approaches vs. dataset (right).

Finally, coverage performance is shown in Figure 13.5 (right). Here, trends for the user-based recommendation strategy using TF-IDF and for the best performing community-based approach using 100 nearest neighbours (CB-100) are shown. It can be seen that the user-based approach provides almost complete coverage for all datasets, well in excess of that given by the community-based approach, particularly for the larger datasets (*movies* and *apps*). This is a positive finding in respect of the utility of reviews as a source of recommendation data. It should be noted that other forms of coverage (see, for example, [27, 65]) have also been proposed; however, an analysis of such criteria is not considered here.

13.3.2.4 Discussion This case-study investigates how user-generated content can be used as a new source of recommendation knowledge. An approach is pro-

posed to represent users and products based on the terms in their associated reviews using techniques from information retrieval. An evaluation performed on short-form, and inherently noisy, reviews from a number of product domains shows promising results. The work described here is related to a growing body of research on the potential for user-generated content to provide product recommendations; for example, enriching user and item profiles by using sentiment analysis and feature extraction techniques, classification of reviews by product category to facilitate personalisation and search [23, 24], and the potential for cross-domain recommendation, where indices created using reviews from one domain are used to recommend products from other domains. For other work in this area, see [1, 2, 33].

13.4 Case Study 2 – Opinionated Recommendation

The previous case-study described an approach to leveraging the text of short-form user-generated product reviews directly for recommendation. Indeed, user-generated reviews have previously been used in a number of recommendation contexts: as part of collaborative filtering approaches to provide virtual ratings [75]; for user profiling [49]; and in content-based recommendation [20].

In this case-study we focus on the type of long-form product reviews typically found on sites like Amazon and TripAdvisor and we describe how these reviews can be used to generate complete item descriptions, in the absence of meta-data or as a complement to meta-data. Crucially, we make the point that these review-based item descriptions are *experiential* in nature — they describe the real experience of users — rather than capturing the type of technical/catalog features that are more common in conventional meta-data representations. We describe how item descriptions are created and how review content can be used to infer opinion sentiment which can be used in a novel way during recommendation. Accordingly items can be selected and ranked not only on the basis that they *have* a given feature (e.g. *Free Wifi* in a hotel), but also based on whether the *opinion* of reviewers about these features is positive or negative (see Figure 13.6).

13.4.1 From Reviews to Recommendation

An overview of the approach is presented in Figure 13.7, which highlights the core opinion mining and recommendation components involved.

Briefly, we start with a set of reviews for some product/item P and any available meta-data. The reviews are mined to identify and extract product features using some straightforward NLP techniques. Next we analyse the sentiment of these features based on the text of the reviews. The combination of features

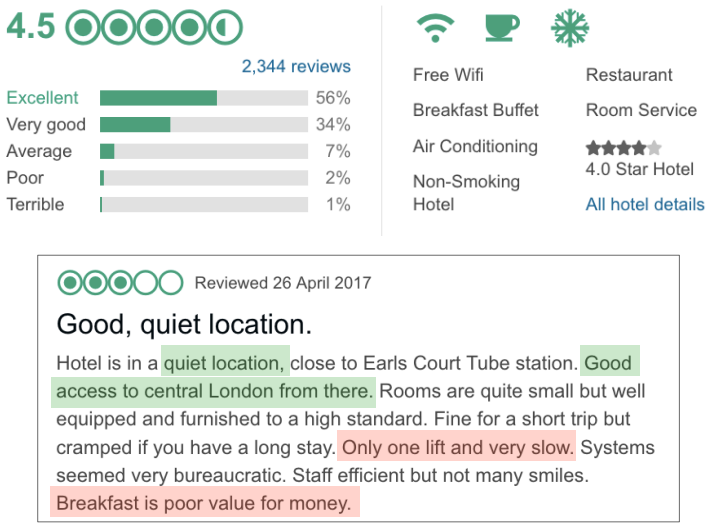


Fig. 13.6 A hotel page on TripAdvisor showing ratings and catalogue meta-data (e.g. *Free Wifi*, *Breakfast Buffet*, *Air Conditioning* etc.) for the property. Features mentioned in a sample review are also highlighted (where green and red denote positive and negative sentiment, respectively.)

and sentiment for each product, plus its meta-data (if available), is combined to produce a product/item description. Given a new user query (i.e. the current item the user is looking at), the recommendation component retrieves and ranks a set of matching items based on a combination of feature similarity and sentiment. In what follows we will describe each of these steps in more detail and provide some concluding evidence in support of the efficacy of this approach for recommendation.

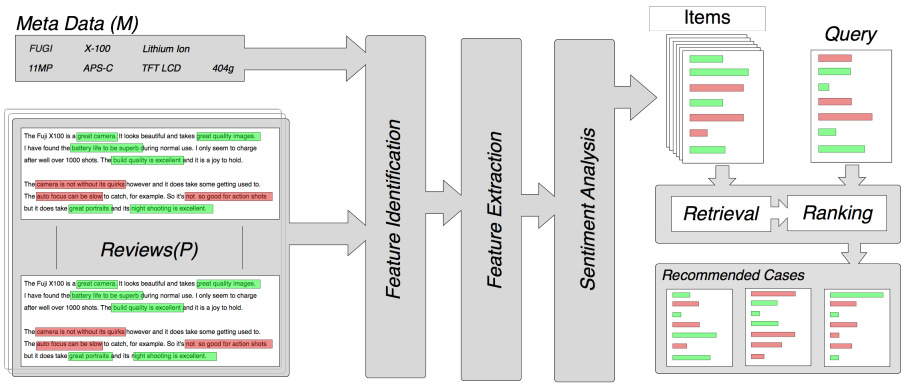


Fig. 13.7 An overview of the experiential product recommendation architecture.

13.4.2 Identifying Review Features

One straightforward way to identify candidate features is to use simple NLP methods to look for certain patterns of words. For example, bi-grams in reviews which conform to one of two basic part-of-speech co-location patterns can be considered as features — an adjective followed by a noun (AN) or a noun followed by a noun (NN). In the former case, bi-grams whose adjective is a sentiment word (e.g. *excellent, terrible* etc.) in the sentiment lexicon used in our approach [32] are excluded. Separately, single-nouns can also be considered as features after eliminating nouns that are rarely associated with sentiment words in reviews as per [32].

13.4.3 Evaluating Feature Sentiment

For each feature we evaluate its sentiment based on the sentence containing the feature within a given review. We use a modified version of the *opinion pattern mining* technique proposed by Moghaddam and Ester [45] for extracting opinions from unstructured product reviews. Once again we use the sentiment lexicon from [32] as the basis for this analysis. For a given feature F_i and corresponding review sentence S_j from review R_k, we determine whether there are any sentiment words in S_j. If there are not then this feature is marked as *neutral* from a sentiment perspective. If there are sentiment words then we identify the word w_{min} which has the minimum word-distance to F_i.

Next we determine the part-of-speech (POS) tags for w_{min}, F_i and any words that occur between w_{min} and F_i. The POS sequence corresponds to an *opinion pattern*. For example, in the case of the bi-gram feature *noise reduction* and the review sentence, *"...this camera has great noise reduction..."* then w_{min} is the word *"great"* which corresponds to an opinion pattern of *JJ-FEATURE* as per Moghaddam and Ester [45]. After a complete pass of all features over all reviews, we can compute the frequency of all recorded opinion patterns. To filter spurious opinion patterns that rarely occur, a pattern is deemed to be valid if it occurs more than the average number of occurrences over all patterns. For valid patterns we assign sentiment to F_i based on the sentiment of w_{min}, subject to whether S_j contains any negation terms within a 4-word-distance of w_{min} [31]. 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 this 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.

13.4.4 From Review Features to Item Descriptions

For each product P we have a set of features $F(P) = \{F_1, ..., F_m\}$ that have been either identified from the meta-data associated with P or that have been discussed in the various reviews of P, $Reviews(P)$. For each feature F_i we compute its *popularity*, which is given by the fraction of reviews it appears in (see Equation 13.11). Also, we compute the *sentiment* associated with each feature; i.e. how often it is mentioned in reviews in a positive, neutral, or negative manner (see Equation 13.12, where $Pos(F_i, P)$, $Neg(F_i, P)$, 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). In this way, each item/product can be represented as the aggregate of its features and their popularity and sentiment data as in Equation 13.13.

$$Pop(F_i, P) = \frac{|\{R_k \in Reviews(P) : F_i \in R_k\}|}{|Reviews(P)|} \ . \tag{13.11}$$

$$Sent(F_i, P) = \frac{Pos(F_i, P) - Neg(F_i, P)}{Pos(F_i, P) + Neg(F_i, P) + Neut(F_i, P)} \ . \tag{13.12}$$

$$Item(P) = \{[F_i, Sent(F_i, P), Pop(F_i, P)] : F_i \in F(P)\} \ . \tag{13.13}$$

13.4.5 Recommending Products

Unlike traditional content-based recommenders — which tend to rely exclusively on similarity in order to rank products with respect to some user profile or query — the above approach accommodates the use of feature sentiment, as well as feature similarity, during recommendation; see [13, 16]. Briefly, a candidate product C can be evaluated against a query product Q (i.e. the current product the user is looking at) according to a weighted combination of similarity and sentiment as per Equation 13.14. $Sim(Q, C)$ is a traditional similarity metric such as cosine similarity, producing a value between 0 and 1, while $Sent(Q, C)$ is a sentiment metric producing a value between -1 (negative sentiment) and +1 (positive sentiment).

$$Score(Q, C) = (1 - w) \times Sim(Q, C) + w \times \left(\frac{Sent(Q, C) + 1}{2} \right) \ . \tag{13.14}$$

13.4.5.1 Similarity Assessment For the purpose of similarity assessment a standard cosine similarity metric based on feature popularity scores can be used, as per Equation 13.15; see also, for example [57].

$$Sim(Q,C) = \frac{\sum\limits_{F_i \epsilon F(Q) \cup F(C)} Pop(F_i,Q) \times Pop(F_i,C)}{\sqrt{\sum\limits_{F_i \epsilon F(Q)} Pop(F_i,Q)^2} \times \sqrt{\sum\limits_{F_i \epsilon F(C)} Pop(F_i,C)^2}} . \quad (13.15)$$

13.4.5.2 Sentiment Assessment Sentiment is somewhat unusual in a recommendation context but its availability offers an additional way to compare products, based on a feature-by-feature sentiment comparison as per Equation 13.16. We can say that feature F_i is *better* in C than in Q if F_i in C has a higher sentiment score than it does in Q.

$$better(F_i,Q,C) = \frac{Sent(F_i,C) - Sent(F_i,Q)}{2} . \quad (13.16)$$

Accordingly we can calculate an overall better score at the product level by aggregating the individual better scores for the product features. We can do this by computing the average better scores across the *union* of features of Q and C, assigning non-shared features a neutral sentiment score of 0. This is captured in Equation 13.17; see also the work of [14] for a second variation on this scoring metric. This approach gives due consideration to the *residual* features in the query and candidate products, that is, those features that are unique to the query or candidate products.

$$Sent(Q,C) = \frac{\sum_{F_i \in F(Q) \cup F(C)} better(F_i,Q,C)}{|F(Q) \cup F(C)|} . \quad (13.17)$$

13.4.6 Evaluation

Finally in this case-study we provide some evaluation results taken from [15] to demonstrate the utility of this approach to opinion mining in recommendation.

The data for this experiment was sourced from TripAdvisor during September 2013. We focused on hotel reviews from six different cities across Europe, Asia, and the US; here, for reasons of space, we consider just two cites, London and Chicago. The data is summarised in Table 13.2, where we show the total number of reviews per city (*#Reviews*), the number of hotels per city (*#Hotels*), as well as including statistics (mean and standard deviation) on the number of features extracted from the reviews per hotel (RF). We can see that this approach to opinion mining produces product descriptions that are rich in features; on average London and Chicago hotels are represented by more than 31 and 28 features per hotel, respectively.

13.4.6.1 Methodology. To evaluate our approach to recommendation we adopt a standard *leave-one-out* methodology. For each city dataset, we treat each hotel in turn as a query case Q and generate a set of top-5 recommendations according to Equation 13.14 using different values of w (0 to 1 in increments

Table 13.2 Dataset statistics.

City	#Reviews	#Hotels	$\mu(\sigma)_{RF}$
London	62,632	717	31.8 (5.5)
Chicago	11,091	125	28.6 (5.0)

of 0.1) in order to test the impact of different combinations of similarity and sentiment; we refer to this approach as RF. Then we compare our recommendations to those produced natively by TripAdvisor (TA) using two comparison metrics. First, we calculate the average *query similarity* between each set of recommendations (RF and TA) and Q using a Jaccard similarity metric. Second, we compare the two sets of recommendations based on the TripAdvisor user ratings to calculate a *ratings benefit* as per Equation 13.18; for example, a ratings benefit of 0.1 means that our RF recommendation list enjoys an average rating score that is 10% higher that those produced by the default TripAdvisor approach (TA).

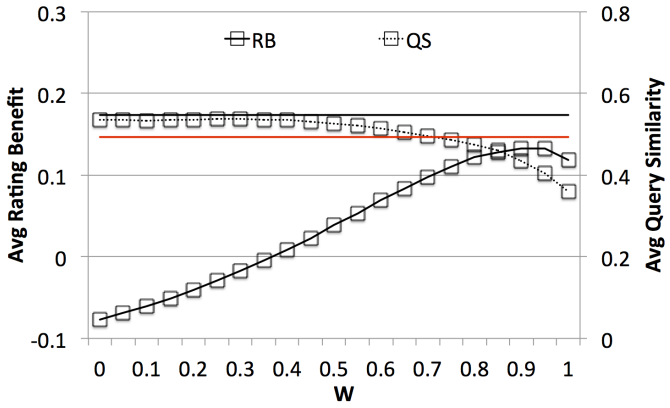
$$Ratings\ Benefit(RF, TA) = \frac{\overline{Rating(RF)} - \overline{Rating(TA)}}{\overline{Rating(TA)}}\ . \tag{13.18}$$

13.4.6.2 Results Figure 13.8 show the results for the London and Chicago hotels, graphing the average ratings benefit (RB) and average query similarity (QS) against different levels of w. Each graph also shows the average query similarity for the TA recommendations (the upper black horizontal solid line), and the region between the upper and lower horitontal lines corresponds to the region of 90% similarity; that is, query similarity scores that fall within this region are 90% as similar to the target query as the default recommendations produced by TA. The intuition here is that query similarity scores which fall below this region run the risk of compromising too much query similarity to be useful as *more-like-this* recommendations.

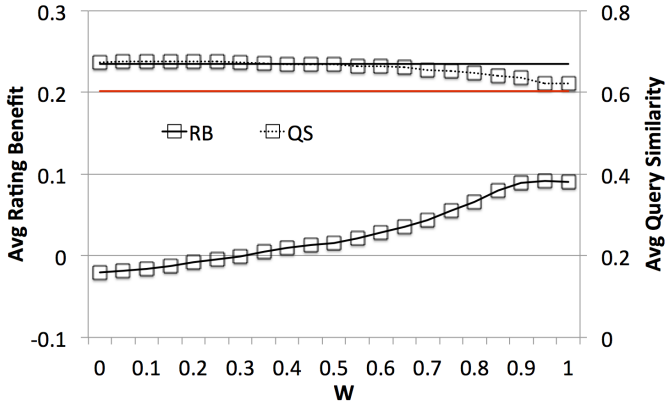
13.4.7 Results Discussion

There are a number of observations that can be made about these results. First, as w increases we can see that there is a steady increase in the average ratings benefit. In other words, as we increase the influence of sentiment in the scoring function (Equation 13.14), we tend to produce recommendations that offer better overall ratings than those produced by TA. Therefore combining similarity and sentiment in recommendation delivers a positive effect overall.

We can also see that as w increases there is a gradual drop in query similarity. In other words, as we increase the influence of sentiment (and therefore decrease the influence of similarity) in the scoring function (Equation 13.14), we tend to produce recommendations that are less similar to the target query. On the one



(a) London hotels.



(b) Chicago hotels.

Fig. 13.8 Ratings benefit (RB) and query similarity (QS) versus w, which controls the relative influence of similarity and sentiment on recommendation ranking scores (see Equation 13.14).

hand, this is a way to introduce more diversity [68] into the recommendation process with the added benefit, as above, that the resulting recommendations tend to enjoy a higher ratings benefit compared to the default TripAdvisor recommendations (TA). But on the other hand, there is the risk that too great a drop in query similarity may lead to products that are no longer deemed to be relevant by the end-user. For this reason, we have (somewhat arbitrarily) chosen to prefer query similarities that remain within 90% of those produced by TA.

We can usefully compare the recommendation approaches by noting the average ratings benefit available at the value of w for which the query similarity of a given approach crosses this 90% (TA) query similarity threshold. For example, in Figure 13.8, for London hotels we can see that the query similarity for the RF

approach falls below the 90% threshold at about $w = 0.7$ and this corresponds to a ratings benefit of 0.1. In the case of Chicago query similarity never dips below this 90% threshold and a maximum ratings benefit of just under 0.1 at $w = 0.9$. Thus, we can conclude that our approach is capable of providing recommendations which enjoy higher ratings compared to those provided by TA, which maintaining a high degree of similarity to the user query.

13.5 Case Study 3 – Review Helpfulness Classification

In the above case-studies, two approaches are described which leverage product reviews for recommendation purposes. However, not all reviews are equally informative and comprehensive, and hence the need to rank reviews for products and to filter less helpful reviews — both to validate the data used as input to recommender systems and to facilitate users to navigate though the thousands of reviews that are often available for popular products. In order to address the issue of information overload in the review space, sites such as TripAdvisor and Amazon allow users to provide manual feedback on review helpfulness; for example, by allowing other consumers to post comments about reviews, to report abuse in cases where review content is considered inappropriate and to indicate whether reviews are found to be helpful or not. While such approaches are of benefit to consumers in highlighting the most helpful reviews, they depend on the willingness of the community at large to contribute feedback and there is no guarantee that all reviews will receive feedback in sufficient quantities to provide a robust signal to consumers. Thus, the community has sought to address this problem by proposing *automated* approaches to classify review helpfulness and a significant body of work has been carried out in this area in recent times; see, for example, [17, 28, 30, 44, 53]. In this case-study, one approach from the literature [37] to automatically classify the helpfulness of reviews is described.

13.5.1 Classifying Review Helpfulness

In the approach presented in [37], the problem of classifying the helpfulness of review is formulated as a supervised classification task. Each review instance is represented by a number of feature sets and the ground truth is given by the review helpfulness as per Equation 13.19:

$$h(r \in R) = \frac{rating_{+}(r)}{rating_{+}(r) + rating_{-}(r)} \ , \tag{13.19}$$

where $rating_{+}(r)$ and $rating_{-}(r)$ are the number of helpful and unhelpful (manually provided) votes for review r, respectively. Thus, once each review in the training set is translated into a feature-based instance representation, a model is

learned which is then applied to classify the helpfulness, $h(r_t)$, of an unseen review instance, r_t. The feature sets and classification approach used are described in the following sections.

13.5.1.1 Feature Sets Review instances consist of feature sets derived from distinct categories which are mined from individual reviews and from the wider community reviewing activity. The following feature sets, which are hypothesised to be predictive of review helpfulness, are considered in [37]:

- *Structural features* capture aspects of the review structure and formatting and include features such as review length, the number of sentences in the review, the mean sentence length, the percentage of sentences with questions, the number of exclamation marks contained in the review, and the number of HTML bold tags `<b>` and line breaks `<br>` in the review body.
- *Lexical features* concern the occurrence of words in reviews; in this case, the TF-IDF statistic of each unigram and bigram occurring in a review are calculated.
- *Syntactic features* capture the linguistic properties of a review by calculating the percentages of tokens that are open-class, nouns, verbs, verbs conjugated in the first person, adjectives or adverbs.
- *Semantic features* capture the intuition that helpful reviews are likely to contain critiques of particular product features (e.g. *capacity* and *zoom* in the case of MP3 players and digital camera products); thus, the number of lexical matches that occur for each product feature and the number of sentiment words in a review are calculated.
- *Meta-data features*, in contrast to the above feature sets, are based on knowledge that is independent of the review text; in this regard, two features that are related to the rating scores that often accompany the review text are considered – namely, the rating score assigned by the reviewer and the absolute difference between this score and the mean rating score assigned by all reviewers.

13.5.1.2 Ranking Reviews Given the availability of training instances and once a classifier is trained, a set of reviews R for a given product can then be ranked in descending order of $h(r)$, $r \in R$. SVM regression [18] using a radial basis function (RBF) kernel is used in [37] as this combination was found to provide optimal performance.

13.5.2 Evaluation

In this section, the datasets used in the evaluation are first described, followed by a description of the evaluation methodology and metrics used. A summary of the key findings of the classification approach is then presented.

Table 13.3 Evaluation dataset statistics (source [37]).

	MP3 Players	*Digital Cameras*
Total Products	736	1,066
Total Reviews	11,374	14,467
Average Reviews/Product	15.4	13.6
Min/Max Reviews/Product	1 / 375	1 / 168

13.5.2.1 Datasets Evaluation datasets consisting of reviews for all products from two product categories, *MP3 Players* and *Digital Cameras*, were sourced from Amazon. Following pre-processing (which included the removal of duplicate reviews, reviews for duplicate products, and reviews for which less than five helpful and unhelpful votes were available), two evaluation datasets were created; see Table 13.3 for statistics relating to these datasets.

13.5.2.2 Methodology and Metrics For each dataset, 10% of products were withheld in order to determine the optimal SVM kernel (RBF) and to tune kernel parameters. Thereafter, the remaining 90% of products were randomly divided into 10 sets, and a 10-fold cross-validation approach was applied to rank (as per Section 13.5.1) the reviews for each product in the test folds. Thus, a ranking for each product's review set is learned, which is compared to a ground truth ranking based on actual helpfulness votes extracted from Amazon.com. Spearman rank correlation is used to compare the learned and actual rankings for each product. Moreover, since in the course of ranking, the absolute helpfulness scores for reviews are learned by the classifier, Pearson correlation is also used to compare these absolute scores to ground truth scores obtained from Amazon.com.

13.5.2.3 Results Results are shown in Table 13.4 for different combinations of features drawn from the subset of features which provides best performance; these features are review length (LEN), unigrams (UGR) and rating score (STR1). When used in isolation, these features provide similar performance. For both datasets, the best performing pair of features is the combination of review length and rating score. As can be seen, the combination of review length, unigrams and rating score features is optimal, achieving Spearman rank corrections of 0.656 and 0.595 for the *MP3 Players* and *Digital Cameras* datasets, respectively.

It is interesting to note the differences between the Spearman rank and Pearson correlation results; in all instances, the quality of the review rankings produced by the classifier (given by Spearman rank correlation) exceeded the performance of the classifier when learning absolute helpfulness scores (given by Pearson correlation). For example, in the case of the *MP3 Players* dataset, Spearman rank and Pearson correlations of 0.656 and 0.476 are seen using a combination of all three features, respectively. Given that learning the absolute helpfulness scores of reviews is a more difficult task, this finding is not surprising; moreover, the results also indicate that accurate rankings can be learned without learning the absolute helpfulness scores of reviews perfectly.

For further details on the evaluation and a discussion on the performance of various other feature combinations, see [37].

Table 13.4 Evaluation results (source [37]).

Feature Combinations	*MP3 Players*		*Digital Cameras*	
	Spearman†	*Pearson*†	*Spearman*†	*Pearson*†
LEN	0.575 ± 0.037	0.391 ± 0.038	0.521 ± 0.029	0.357 ± 0.029
UGR	0.593 ± 0.036	0.398 ± 0.038	0.499 ± 0.025	0.328 ± 0.029
STR1	0.589 ± 0.034	0.326 ± 0.038	0.507 ± 0.029	0.266 ± 0.030
UGR+STR1	0.644 ± 0.033	0.436 ± 0.038	0.490 ± 0.032	0.324 ± 0.032
LEN+UGR	0.582 ± 0.036	0.401 ± 0.038	0.553 ± 0.028	0.394 ± 0.029
LEN+STR1	0.652 ± 0.033	0.470 ± 0.038	0.577 ± 0.029	0.423 ± 0.031
LEN+UGR+STR1	**0.656 ± 0.033**	**0.476 ± 0.038**	**0.595 ± 0.028**	**0.442 ± 0.031**

LEN=*Length*; UGR=*Unigram*; STR=*Stars*

†95% confidence bounds are calculated using 10-fold cross-validation.

13.5.2.4 Discussion User-generated reviews have become an important source of knowledge for consumers and are known to play an active role in decision making in many domains. However, given the thousands of reviews which can often accrue for popular products on sites such as Amazon and TripAdvisor, a new challenge arises — namely, how best to facilitate users to rapidly and effectively identify the most useful reviews. Hence the need for automatic approaches to identify review helpfulness to assist users by, for example, filtering less informative or comprehensive reviews from the user's view. The case study presented in this chapter highlights one approach in a significant body of work carried out in this area; further work on this problem can be found in [17, 30, 44, 51, 53].

13.6 Conclusions

Today, product reviews have become an important part of our online experience, assisting consumers to make informed choices and providing key insights to retailers about their product offerings. For example, Lee et al. report that 84 percent of Americans are influenced by online reviews when they are making purchase decisions [41]; see also [12, 76]. Further, many companies have now recognised that consumer reviews represent a new and important communication channel with their consumers, and they have begun monitoring online consumer reviews as a crucial source of product feedback [19]. Moreover, companies can predict their performance or sales according to this online feedback; for example, Duan et al. used Yahoo movie reviews and box office returns to examine the

persuasive and awareness effects of online user reviews on the daily box office performance [19].

Increasingly, researchers are also leveraging product reviews for the purposes of user profiling and recommendation. In particular, reviews often capture detailed and nuanced user opinions for different kinds of products and services, and thus represent a plentiful, albeit noisy and unstructured, alternative source of recommendation knowledge to replace or complement the more conventional data sources such as product ratings and meta-data. In this chapter, we have presented two case-studies which describe particular approaches in which review data can be successfully leveraged for recommendation. Moreover, we have described an approach to estimate review quality in order to help users cope with the volume and variability of review content. Given the prevalence of user-generated content online and the valuable insights it provides to both consumers and retailers alike, this area of research presents many exciting opportunities for the future.

13.7 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. Aciar, S., Zhang, D., Simoff, S., Debenham, J.: Recommender system based on consumer product reviews. In: Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence (WI-IATW '06). pp. 719–723. IEEE Computer Society, Washington, DC, USA (2006)
2. Ahn, S., Shi, C.K.: Exploring movie recommendation system using cultural metadata. In: Pan, Z., Cheok, A.D., Mller, W., Rhalibi, A. (eds.) Transactions on Edutainment II, Lecture Notes in Computer Science, vol. 5660, pp. 119–134. Springer Berlin Heidelberg (2009), http://dx.doi.org/10.1007/978-3-642-03270-7_9
3. Angel, A., Koudas, N., Sarkas, N., Srivastava, D.: What's on the grapevine? In: Proceedings of the 35th SIGMOD International Conference on Management of Data (SIGMOD '09). pp. 1047–1050. ACM, New York, NY, USA (2009)
4. 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)
5. Banerjee, N., Chakraborty, D., Dasgupta, K., Mittal, S., Joshi, A., Nagar, S., Rai, A., Madan, S.: User interests in social media sites: An exploration with micro-blogs. In: Proceeding of the 18th ACM Conference on Information and Knowledge Management (CIKM '09). pp. 1823–1826. ACM, New York, NY, USA (2009)
6. Bermingham, A., Smeaton, A.F.: Classifying sentiment in microblogs: Is brevity an advantage? In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management (CIKM '10). pp. 1833–1836. ACM, New York, NY, USA (2010), http://doi.acm.org/10.1145/1871437.1871741

7. Bogers, T.: Tag-based recommendation. In: Brusilovsky, P., He, D. (eds.) Social Information Access, LNCS, vol. 10100, p. in this volume. Springer, Heidelberg (2017)
8. Breese, J.S., Heckerman, D., Kadie, C.M.: Empirical analysis of predictive algorithms for collaborative filtering. In: Cooper, G.F., Moral, S. (eds.) Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI '98). pp. 43–52. Morgan Kaufmann (1998)
9. Brew, A., Greene, D., Cunningham, P.: Using crowdsourcing and active learning to track sentiment in online media. In: Proceedings of the 19th European Conference on Artificial Intelligence (ECAI '10). pp. 145–150. IOS Press, Amsterdam, The Netherlands, The Netherlands (2010), http://portal.acm.org/citation.cfm?id=1860967.1860997
10. Burke, R.: Hybrid recommender systems: Survey and experiments. User Modeling and User-Adapted Interaction 12(4), 331–370 (2002)
11. Cruz, F.L., Troyano, J.A., Enríquez, F., Ortega, F.J., Vallejo, C.G.: A knowledge-rich approach to feature-based opinion extraction from product reviews. In: Proceedings of the 2nd International Workshop on Search and Mining User-generated Contents (SMUC '10). pp. 13–20. ACM, New York, NY, USA (2010), http://doi.acm.org/10.1145/1871985.1871990
12. Dhar, V., Chang, E.A.: Does chatter matter? The impact of user-generated content on music sales. Journal of Interactive Marketing 23(4), 300–307 (2009), http://www.sciencedirect.com/science/article/pii/S1094996809000723
13. 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. pp. 411–414. RecSys '13, ACM, New York, NY, USA (2013), http://doi.acm.org/10.1145/2507157.2507199
14. Dong, R., O'Mahony, M.P., Schaal, M., McCarthy, K., Smyth, B.: Combining similarity and sentiment in opinion mining for product recommendation. Journal of Intelligent Information Systems pp. 1–28 (2015), http://dx.doi.org/10.1007/s10844-015-0379-y
15. Dong, R., O'Mahony, M.P., Smyth, B.: Further experiments in opinionated product recommendation. In: Proceedings of the 22nd International Conference on Case-Based Reasoning. pp. 110–124. ICCBR '14, Springer (2014)
16. Dong, R., Schaal, M., O'Mahony, M.P., McCarthy, K., Smyth, B.: Opinionated product recommendation. In: Case-Based Reasoning Research and Development, Lecture Notes in Computer Science, vol. 7969, pp. 44–58. Springer Berlin Heidelberg (2013)
17. 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 2013). pp. 1310–1316 (2013)
18. Drucker, H., Burges, C.J.C., Kaufman, L., Smola, A.J., Vapnik, V.N.: Support vector regression machines. In: Advances in Neural Information Processing Systems 9 (NIPS 1996). pp. 155–161. MIT Press (1996)
19. Dwyer, P.: Measuring the value of electronic word of mouth and its impact in consumer communities. Journal of Interactive Marketing 21(2), 63–79 (2007)
20. Esparza, S.G., O'Mahony, M.P., Smyth, B.: Effective product recommendation using the real-time web. In: Bramer, M., Petridis, M., Hopgood, A. (eds.) Research and Development in Intelligent Systems XXVII, pp. 5–18. Springer London (2011), http://dx.doi.org/10.1007/978-0-85729-130-1_1
21. Farzan, R., Brusilovsky, P.: Social navigation. In: Brusilovsky, P., He, D. (eds.) Social Information Access, p. in this volume. LNCS, Springer, Heidelberg (2017)
22. Garcia Esparza, S., O'Mahony, M.P., Smyth, B.: On the real-time web as a source of recommendation knowledge. In: Proceedings of the 4th ACM Conference on Recommender Systems (RecSys 10). pp. 305–308. ACM, New York, NY, USA (2010)
23. Garcia Esparza, S., O'Mahony, M.P., Smyth, B.: Towards tagging and categorization for micro-blogs. In: Proceedings of the 21st Irish Conference on Artificial Intelligence and Cognitive Science (AICS 2010). pp. 122–131 (2010)
24. Garcia Esparza, S., O'Mahony, M.P., Smyth, B.: Further experiments in micro-blog categorization. In: Proceedings of the 22nd Irish Conference on Artificial Intelligence and Cognitive Science (AICS 2011). pp. 156–165 (2011)

25. Garcia Esparza, S., O'Mahony, M.P., Smyth, B.: A multi-criteria evaluation of a user-generated content based recommender system. In: Proceedings of the 3rd Workshop on Recommender Systems and the Social Web, 5th ACM Conference on Recommender Systems (RSWEB 2011) (2011)
26. Garcia Esparza, S., O'Mahony, M.P., Smyth, B.: Mining the real-time web: A novel approach to product recommendation. Knowledge Based Systems 29, 3–11 (2012)
27. Ge, M., Delgado-Battenfeld, C., Jannach, D.: Beyond accuracy: Evaluating recommender systems by coverage and serendipity. In: Proceedings of the Fourth ACM Conference on Recommender Systems. pp. 257–260. RecSys '10, ACM, New York, NY, USA (2010), http://doi.acm.org/10.1145/1864708.1864761
28. Ghose, A., Ipeirotis, P.G.: Designing novel review ranking systems: Predicting the usefulness and impact of reviews. In: Proceedings of the Ninth International Conference on Electronic Commerce. pp. 303–310. ICEC '07, ACM, New York, NY, USA (2007), http://doi.acm.org/10.1145/1282100.1282158
29. Herlocker, J.L., Konstan, J.A., Borchers, A., Riedl, J.: An algorithmic framework for performing collaborative filtering. In: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '99). pp. 230–237. ACM, New York, NY, USA (1999)
30. Hsu, C.F., Khabiri, E., Caverlee, J.: Ranking comments on the social web. In: Proceedings of the International Conference on Computational Science and Engineering (CSE'09). vol. 4, pp. 90–97. IEEE (2009)
31. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '04). pp. 168–177. ACM, New York, NY, USA (2004)
32. Hu, M., Liu, B.: Mining opinion features in customer reviews. In: Proceedings of the 19th National Conference on Artificial Intelligence. pp. 755–760. AAAI '04, AAAI Press (2004), http://dl.acm.org/citation.cfm?id=1597148.1597269
33. Jakob, N., Weber, S.H., Müller, M.C., Gurevych, I.: Beyond the stars: Exploiting free-text user reviews to improve the accuracy of movie recommendations. In: Proceeding of the 1st International CIKM Workshop on Topic-sentiment Analysis for Mass Opinion (TSA '09). pp. 57–64. ACM, New York, NY, USA (2009), http://doi.acm.org/10.1145/1651461.1651473
34. Jannach, D., Lerche, L., Zanker, M.: Recommending based on implicit feedback. In: Brusilovsky, P., He, D. (eds.) Social Information Access, LNCS, vol. 10100, p. in this volume. Springer, Heidelberg (2017)
35. Jindal, N., Liu, B.: Opinion spam and analysis. In: Proceedings of the 2008 International Conference on Web Search and Data Mining. pp. 219–230. WSDM '08, ACM, New York, NY, USA (2008), http://doi.acm.org/10.1145/1341531.1341560
36. Jindal, N., Liu, B., Lim, E.P.: Finding unusual review patterns using unexpected rules. In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management. pp. 1549–1552. CIKM '10, ACM, New York, NY, USA (2010), http://doi.acm.org/10.1145/1871437.1871669
37. 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), http://portal.acm.org/citation.cfm?id=1610075.1610135
38. Kluver, D., Ekstrand, M., Konstan, J.: Collaborative filtering. In: Brusilovsky, P., He, D. (eds.) Social Information Access, p. in this volume. LNCS, Springer, Heidelberg (2017)
39. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer 42(8), 30–37 (August 2009), http://dx.doi.org/10.1109/MC.2009.263
40. Lee, D., Brusilovsky, P.: Social link-based recommendations. In: Brusilovsky, P., He, D. (eds.) Social Information Access, p. in this volume. LNCS, Springer, Heidelberg (2017)
41. Lee, J., Park, D.H., Han, I.: The different effects of online consumer reviews on consumers' purchase intentions depending on trust in online shopping mall: An advertising perspective.

Internet Research 21(2), 187–206 (2011), http://dblp.uni-trier.de/db/journals/intr/intr21.html#LeePH11
42. Li, F., Huang, M., Yang, Y., Zhu, X.: Learning to identify review spam. In: Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence. pp. 2488–2493. IJCAI'11, AAAI Press (2011), http://dx.doi.org/10.5591/978-1-57735-516-8/IJCAI11-414
43. Lim, E.P., Nguyen, V.A., Jindal, N., Liu, B., Lauw, H.W.: Detecting product review spammers using rating behaviors. In: Proceedings of the 19th ACM international conference on Information and Knowledge Management. pp. 939–948. CIKM '10, ACM, New York, NY, USA (2010), http://doi.acm.org/10.1145/1871437.1871557
44. Liu, Y., Huang, X., An, A., Yu, X.: Modeling and predicting the helpfulness of online reviews. In: Proceedings of the 2008 Eighth IEEE International Conference on Data Mining (ICDM 2008). pp. 443–452. IEEE Computer Society, Pisa, Italy (2008)
45. 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. pp. 1825–1828. CIKM '10, ACM, New York, NY, USA (2010), http://doi.acm.org/10.1145/1871437.1871739
46. Mooney, R.J., Roy, L.: Content-based book recommending using learning for text categorization. In: Proceedings of the Fifth ACM Conference on Digital Libraries (DL '00). pp. 195–204. ACM, New York, NY, USA (2000)
47. Mukherjee, A., Liu, B., Glance, N.: Spotting fake reviewer groups in consumer reviews. In: Proceedings of the 21st International Conference on World Wide Web. pp. 191–200. WWW '12, ACM, New York, NY, USA (2012), http://doi.acm.org/10.1145/2187836.2187863
48. Mullen, T., Collier, N.: Sentiment analysis using support vector machines with diverse information dources. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '04). pp. 412–418 (2004)
49. Musat, C.C., Liang, Y., Faltings, B.: Recommendation using textual opinions. In: Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI 13). pp. 2684–2690. AAAI Press, Menlo Park, California (2013)
50. O'Callaghan, D., Harrigan, M., Carthy, J., Cunningham, P.: Network analysis of recurring youtube spam campaigns. In: Proceedings of the Sixth International Conference on Weblogs and Social Media (ICWSM 12). pp. 531–534 (2012)
51. O'Mahony, M.P., Cunningham, P., Smyth, B.: An assessment of machine learning techniques for review recommendation. In: Coyle, L., Freyne, J. (eds.) Artificial Intelligence and Cognitive Science, Lecture Notes in Computer Science, vol. 6206, pp. 241–250. Springer Berlin Heidelberg (2010), http://dx.doi.org/10.1007/978-3-642-17080-5_26
52. O'Mahony, M.P., Smyth, B.: Learning to recommend helpful hotel reviews. In: Proceedings of the Third ACM Conference on Recommender Systems (RecSys '09). pp. 305–308. ACM, New York, NY, USA (2009), http://doi.acm.org/10.1145/1639714.1639774
53. O'Mahony, M.P., Smyth, B.: A classification-based review recommender. Knowledge Based Systems 23(4), 323–329 (2010)
54. Ott, M., Choi, Y., Cardie, C., Hancock, J.T.: Finding deceptive opinion spam by any stretch of the imagination. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1. pp. 309–319. HLT '11, Association for Computational Linguistics, Stroudsburg, PA, USA (2011), http://dl.acm.org/citation.cfm?id=2002472.2002512
55. Pandey, V., Iyer, C.K.: Sentiment analysis of microblogs (2009), http://www.stanford.edu/class/cs229/proj2009/PandeyIyer.pdf, technical Report, Stanford University, http://www.stanford.edu/class/cs229/proj2009/PandeyIyer.pdf (Accessed: November 2010)
56. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up?: Sentiment classification using machine learning techniques. In: Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP '02). pp. 79–86. Association for Computational Linguistics, Morristown, NJ, USA (2002)

57. Pazzani, M.J., Billsus, D.: Content-based recommendation systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) The Adaptive Web: Methods and Strategies of Web Personalization, pp. 325–341. Springer-Verlag, Berlin, Heidelberg (2007)
58. Popescu, A.M., Etzioni, O.: Extracting product features and opinions from reviews. In: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT '05). pp. 339–346. Association for Computational Linguistics, Morristown, NJ, USA (2005)
59. Read, J.: Using emoticons to reduce dependency in machine learning techniques for sentiment classification. In: Proceedings of the ACL Student Research Workshop (ACL '05). pp. 43–48. Association for Computational Linguistics, Morristown, NJ, USA (2005), http://portal.acm.org/citation.cfm?id=1628960.1628969
60. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: GroupLens: An open architecture for collaborative filtering of netnews. In: Proceedings of the ACM Conference on Computer-Supported Cooperative Work (CSCW '94). pp. 175–186. Chapel Hill, North Carolina, USA (August 1994)
61. Robertson, S., Walker, S., Jones, S., Hancock-Beaulieu, M., Gatford, M.: Okapi at TREC-3. In: Text REtrieval Conference (TREC). pp. 109–126 (1996)
62. Salton, G., McGill, M.J.: Introduction to Modern Information Retrieval. McGraw-Hill, Inc. (1986)
63. Sarwar, B.M., Karypis, G., Konstan, J.A., Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th International World Wide Web Conference (WWW '01). pp. 285–295. Hong Kong (May 2001)
64. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Analysis of recommendation algorithms for e-commerce. In: Proceedings of the 2nd ACM Conference on Electronic Commerce (EC '00). pp. 158–167. ACM, Minneapolis, Minnesota, USA (October 17-20 2000)
65. Shani, G., Gunawardana, A.: Evaluating recommendation systems. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.) Recommender Systems Handbook, pp. 257–297. Springer US (2011), http://dx.doi.org/10.1007/978-0-387-85820-3_8
66. Shardanand, U., Maes, P.: Social information filtering: Algorithms for automating "word of mouth". In: Proceedings of the SIGCHI Conference on Human factors in Computing Systems (CHI '95). pp. 210–217. ACM Press/Addison-Wesley Publishing Co. (1995)
67. Smyth, B.: Case-based recommendation. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) The Adaptive Web: Methods and Strategies of Web Personalization, pp. 342–376. Springer-Verlag, Berlin, Heidelberg (2007)
68. Smyth, B., McClave, P.: Similarity vs. diversity. In: Aha, D.W., Watson, I. (eds.) Case-Based Reasoning Research and Development, Lecture Notes in Computer Science, vol. 2080, pp. 347–361. Springer Berlin Heidelberg (2001), http://dx.doi.org/10.1007/3-540-44593-5_25
69. Tang, H., Tan, S., Cheng, X.: A survey on sentiment detection of reviews. Expert Systems With Applications 36(7), 10760–10773 (2009)
70. Wang, G., Xie, S., Liu, B., Yu, P.S.: Review graph based online store review spammer detection. In: Proceedings of the 2011 IEEE 11th International Conference on Data Mining. pp. 1242–1247. ICDM '11, IEEE Computer Society, Washington, DC, USA (2011), http://dx.doi.org/10.1109/ICDM.2011.124
71. Wang, W., Wang, H.: Opinion-enhanced collaborative filtering for recommender systems through sentiment analysis. New Review of Hypermedia and Multimedia 21(3-4), 278–300 (2015)
72. Weerkamp, W., de Rijke, M.: Credibility improves topical blog post retrieval. In: Proceedings of the Association for Computational Linguistics with the Human Language Technology Conference (ACL-08:HLT). pp. 923–931. Columbus, Ohio, USA (2008)
73. Wiebe, J., Riloff, E.: Creating subjective and objective sentence classifiers from unannotated texts. In: Proceedings of the 6th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing '05). pp. 486–497. Mexico City, Mexico (2005)

74. Wietsma, R.T.A., Ricci, F.: Product reviews in mobile decision aid systems. In: Proceedings of Pervasive Mobile Interaction Devices (PERMID 2005) - Mobile Devices as Pervasive User Interfaces and Interaction Devices - Workshop in conjunction with: The 3rd International Conference on Pervasive Computing (PERVASIVE 2005). pp. 15–18. Munich, Germany (2005)
75. Zhang, W., Ding, G., Chen, L., Li, C., Zhang, C.: Generating virtual ratings from Chinese reviews to augment online recommendations. ACM Transactions on Intelligent Systems and Technology 4(1), 9:1–9:17 (Feb 2013), http://doi.acm.org/10.1145/2414425.2414434
76. Zhu, F., Zhang, X.M.: Impact of online consumer reviews on sales: The moderating role of product and consumer characteristics. Journal of Marketing 74(2), 133–148 (2010)