Text-based User-kNN: measuring user similarity based on text reviews

Maria Terzi, Matthew Rowe, Maria-Angela Ferrario, Jon Whittle

School of Computing & Communications, InfoLab21,
Lancaster University
LA1 4WA Lancaster UK
{m.terzi, m.rowe, m.ferrario, j.n.whittle}@lancaster.ac.uk

Published in User, Modeling and Adaptation (2014)

Abstract. This article reports on a modification of the user-kNN algorithm that measures the similarity between users based on the similarity of text reviews, instead of ratings. We investigate the performance of text semantic similarity measures and we evaluate our text-based user-kNN approach by comparing it to a range of ratings-based approaches in a ratings prediction task. We do so by using datasets from two different domains: movies from RottenTomatoes and Audio CDs from Amazon Products. Our results show that the text-based user-kNN algorithm performs significantly better than the ratings-based approaches in terms of accuracy measured using RMSE.

Keywords: Recommender systems, Collaborative Filtering, Text reviews, Semantic similarity measures.

Introduction

Recommender systems work by predicting how users will rate items of potential interest. A common approach is Collaborative Filtering (CF); "k-Nearest Neighbors" (user-kNN), for example, predicts a user's rating according to how similar users rated the same item [1]. User-kNN matches similar users based on the similarity of their ratings on items. We argue that ratings alone are insufficient to fully reflect the similarity between users for two reasons: a) ratings do not capture the rationale behind a user's rating, and b) there is a high probability (p=0.8) that two ratings of the same value on the same item will be given for different reasons [2]. We identify this as a potential challenge for ratings-based approaches and define it as *a similarity reflection problem*.

Existing work [3, 4] reports that measuring the similarity of users using the sentiment of their text reviews, instead of ratings, improves the accuracy of user-kNN. However, we argue that a sentiment-based approach does not fully address the similarity reflection problem since *the reasons behind a sentiment of a review remain unexploited.* In other words, the sentiment, similar to a rating, says *how much* a person liked an item, but it misses the *reason why*. For example, in the case of a movie, did the reviewer like it because of the performance of a

The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-319-08786-3_17

specific actor? Or because of the style of the director? We argue that text reviews potentially offer a substantiated opinion of a user for an item, making them an ideal source of knowledge for enhancing the recommendation process.

There is a growing body of research which aims to exploit the content of text reviews for various tasks. However, the analysis of text reviews to address the similarity reflection problem remains an under-explored area. Work in [5, 6] for example, uses text reviews to construct user preference profiles: sets of item features (such as plot or special effects in the movie domain) are extracted from the users' text reviews. These user preference profiles may then be used to measure user similarity in a user-kNN algorithm [5], or they are used to constrain CF by only using reviews similar to a user's profile when making recommendations [6]. These approaches assume that "the overall number of opinions regarding a certain item feature reveals how important that feature is to a user" [6]. An important aspect of this assumption, however, is that it generalizes the features a user finds interesting to all the items in a domain. For example, it assumes that if a user likes special effects in an action movie, s/he also likes to have special effects in a drama. Hence, such an approach does not *distinguish between user preferences across domains.*

Our previous investigation [2] indicated that users' similarity is not well reflected in rating-based approaches that only rely on users' ratings, and suggested the use of text reviews. In this paper, we present the text-based user-kNN, a modification of user-kNN algorithm,that uses text reviews to measure similarity between users, instead of using ratings.

Our text-based user-kNN applies text similarity measures directly on text reviews of co-reviewed items, instead of applying statistical similarity measures on ratings or constructing profiles of user preferences extracted from text reviews. In doing so, we attempt to form neighborhoods of users who have reviewed the same items with semantically similar reviews, while respecting the diversity of user feature preferences over items. We then identify a target user's nearest neighbor, and use their ratings to predict the target user's ratings. In an evaluation of the approach, we measure the accuracy of its predictions by comparing them to the target user's actual ratings.

This paper's two main contributions are:

1. A text-based user-kNN approach that measures the similarity of users by applying text similarity measures directly on users' text reviews for each co-reviewed item.
2. An extensive evaluation which includes:
   a. An investigation of the performance of various text similarity measures in the text-based user-kNN approach. The investigation highlights a significant improvement of text semantic similarity measures over a simple lexical matching measure.
   b. A comparison of text-based user-kNN with a range of ratings-based approaches in a ratings prediction task. Results show that the text-based user-kNN produces a small but significant improvement over ratings-based approaches in minimizing the RMSE between the actual and the predicted ratings. Our evaluation is performed using two different datasets – a RottenTomatoes dataset and an Audio CD dataset from AmazonProductReviews. The consistently higher accuracy of the text-based user-kNN approach verifies its better performance.

The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-319-08786-3_17

The novelty of our approach over previous work lies in the way we incorporate text reviews in user-kNN. We calculate the direct similarity of text reviews to measure the similarity between users and form neighborhoods of similar users. In addition, we provide evidence of the effectiveness of our approach in predicting ratings, over various state-of-art rating based approaches using two different datasets.

Related Work

The use of text reviews as implicit feedback to improve the recommendation process is an expansive topic. In matrix factorization CF approaches, text reviews have been used to define a 'regularizer' score for the factorization model. The regularizer is assigned one of three scores depending on the methodology used: the opinion score, calculated using feature extraction and sentiment analysis of text reviews [7]; the sentiment score, calculated using only sentiment analysis on text reviews [8]; or the review-quality score, calculated based on the occurrence of features in text reviews [9]. In addition, in [10], both ratings and features extracted from text reviews are used to define a regularizer. However, the above approaches are not focused on improving the performance of neighborhood based models, such as user-kNN.

In user-kNN approaches, similar to our work, research exploiting text reviews is limited to applying sentiment analysis on text reviews [3, 4], or building user profiles of feature preferences extracted from text reviews [5, 6]. Sentiment analysis has been applied on text reviews to either reflect a user's interest in an item in terms of a binary score (like/dislike) [3], or to refine a list of rating-based CF recommendations by removing items whose review is labeled with a negative sentiment [4]. However, in such approaches [3, 4], *the reasons behind a user's rating remain unexploited.* Chen and Wang [5] investigated regression models on user text reviews to infer weighted feature preferences. They then matched users with similar weighted feature preferences to produce the item recommendations. Musat et al. [6], proposed Topic Profile CF (TPCF), a technique which builds user profiles based on extracted 'topics' from the users' aggregated text reviews. They then use the item reviews that are most similar to the user's profile to predict the user's rating for the item.

In contrast to our work, TPCF does not form neighborhoods of similar users based on their text reviews. Chen and Wang [5], focused on producing item recommendations instead of predicting ratings. Furthermore, both approaches [5, 6] are based on building user profiles with features extracted from all text reviews thus assuming that what a user likes in one domain, s/he also likes in another domain.

User preference profiles have also been used by Content Based (CB) recommendation approaches. For example, Levi et al. [11], used text reviews to infer the 'traits' or preferences of contextually similar groups in a hotel recommender and then calculate the impression a user has for a hotel. The main difference our approach with Levi et al. [11], is that we form neighborhoods of users based on their text reviews rather than exploiting the preferences of predefined groups of users. Also, we measure the direct similarity of the users' text reviews, instead of building profiles of user preferences. In doing so, we distinguish user feature preferences across domains.

The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-319-08786-3_17

Text-based User-kNN

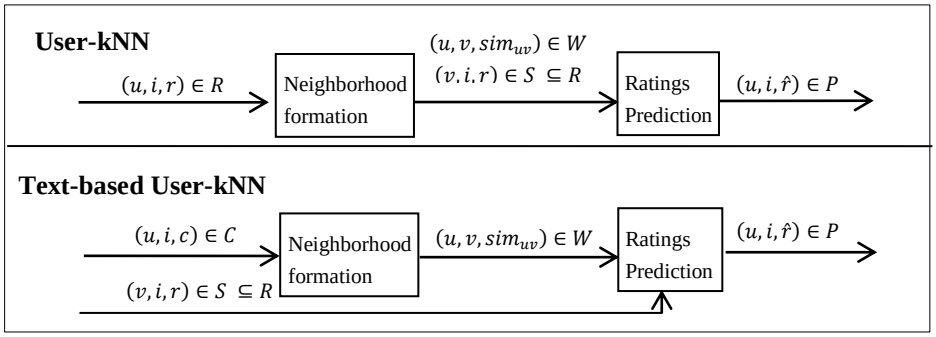

Fig. 1. User-kNN and Text-Based User-kNN

In this section we present text-based user-kNN, a modification of user-kNN which incorporates text reviews in the measurement of similarity between users, shown in Figure 1. We are given a set of users $u \in U$, a set of items $i \in I$ and a set of quadruples D, $(u,i,r,c) \in D$, with each quadruple corresponding to a review of user *u* on item *i* using the rating *r* and the content of text review *c.* We reserve special indexing letters for distinguishing users from items: for users u, v ($u,v \in U$) and for items *i* ($i \in I$). Our objective is to predict each unknown rating $\hat{r}$ of user *u* for item *i* in set *P.*

The first phase of user-kNN is the neighborhood formulation. During this phase the main goal is to measure the similarity between users and define a set of users $\in N$, who tend to review items similarly to *u* ("neighbors"). The similarity of two users is measured by applying similarity measures between their reviews on their co-reviewed items. User-kNN uses ratings to measure similarity between users. It accepts as input the set $R = \{(u,i,r): (u,i,r,c) \in D\}$, with each triple corresponding to a review of user *u*, on item *i,* using rating *r*. User-kNN calculates the similarity,sim_{uv}, between the users *u* and *v,* by applying statistical similarity measures such as Pearson, between the ratings of the two users. On the other hand, text-based user-kNN uses text reviews. In this phase text-based user-kNN accepts as input the set $C = \{(u,i,c): (u,i,r,c) \in D\}$. Each triple in C represents a review of the user *u*, on item *i* with content of the text review *c.* Text-based user-kNN measures similarity sim_{uv}, between the users *u* and *v* by applying a text similarity measure ψ over the content of the reviews of the two users' for each of their co-reviewed items (Equation 1). The measure ψ calculates the similarity between the content of two the text reviews to produce a numerical similarity score from 0 (no similarity) to +1 (strong similarity), $\psi: c \times c \rightarrow [0,1]$.

$$sim_{uv} = \frac{1}{|I_u \cap I_v|} \sum_{i \in I_u \cap I_v} \psi(c_{ui}, c_{vi}) \quad (1)$$

where $I_u \cap I_v$ is the set of co-reviewed items of user *u* and user *v*, and $\psi(c_{ui}, c_{vi})$ the text similarity measure between the content c_{ui} of the text review of user *u* for item *i* and the content of text review c_{vi} of user *v* for item *i.*

The calculated similarity scores between the users are stored in set W, $(u,v,sim_{uv},) \in W$, and are used a) to construct the set N using the k users who have the highest similarity score with user u and b) as a weight in the ratings prediction phase.

The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-319-08786-3_17

In order to estimate the unknown rating $\hat{r}_{ui}$ we resort to a set of users *v*, *N(v,k),* who have the highest similarity with user *u* and actually rated item *i* (i.e., r_{vi} is known for each user $v \in N$). Both approaches use the set S, $(v, i, r) \in S \subseteq R$, in which each triple includes the rating *r* of the user *v* for an item *i.* The number of neighbors *k* is calculated during the training phase of the model. The estimated value of $\hat{r}_{ui}$ is taken as a weighted average of the neighbors' ratings:

$$\hat{r}_{ui} = \frac{\sum_{v \in N(v,k)} sim_{uv} \times (r_{vi})}{\sum_{v \in N(v,k)} |sim_{uv}|} \quad (2)$$

where *N* is the set of *k* similar neighbors of user *u*; $r_{vi}$is the rating of the neighbor *v* for the item *i*; and sim_{uv} is the similarity between the users *u* and *v* stored in set W.

In cases where no formulation of neighborhood can be established, both user-kNN approaches use the average value of an item's ratings to predict a user's rating.

Short Text Similarity Measures.

The core part of our text-based user-kNN uses a text similarity measure (ψ) that can identify 'similar reviews': reviews that use semantically similar wordings to review an item. A typical approach for finding the similarity between two text segments is to use a simple lexical matching method such as 'word overlap' to produce a similarity score based on the number of words that occur in both segments. While successful to a certain degree, such lexical methods cannot identify semantic similarity. For instance, there is an obvious similarity between the text segments "The movie has an amazing storyline" and "The plot of this film is good", but most of the text similarity measures will fail in identifying any kind of connection between these texts because of the lack of lexical overlap. The semantic similarity between two words can be measured using WordNet [12], an online lexical database of English terms structured based on synsets, that is, sets of synonymous words. Synsets are connected to one another through relations such as "is-a". For example, "plot" and "storyline" nouns are in the same synset, which is connected to the "noun communication" synset by an "is-a" relationship.

We employ six word similarity measures: a simple word overlap measure; two measures based on the path length two words in WordNet; and three that use the information content (IC) of a word. All the WordNet measures we employ are publicly provided by [13]. To derive the similarity score of two text reviews (ψ) we use the average of the similarity scores (*s*) between each of their words. All stop words have been removed from the datasets using the stop word lists provided by Lewis et al.[14].

Semantic similarity measures based on path length.

Two of the measures we use in our experiment, the measure provided by Leacock and Chodorow [15] and the measure provided by Wu and Palmer [16], are based on path length of a WordNet taxonomy. A path length is equal to the count of relation links of words in the taxonomy. The lower the distance between two words, the higher the similarity between them. For example, the path length of two synonymous words is 0. The measure by Leacock

The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-319-08786-3_17

and Chodorow[15], denoted as s_{lch} , returns a similarity score based on the shortest path that connects two words and the maximum depth of the taxonomy:

$$s_{lch}(w_1, w_2) = -\log \frac{path(w_1,w_2)}{2*D} \quad (3)$$

where $path(w_1, w_2)$ is the shortest distance between the words w_1 and w_2, and *D* is a constant (e.g., the maximum depth in the WordNet taxonomy).

The similarity metric by Wu and Palmer [16], s_{wup}, is based on the depth of the two words in WordNet and that of their least common subsumer (LCS), that is, the word that is a shared ancestor of the two words. For example the LCS of the words 'car' and 'boat' would be 'vehicle'. The s_{wup}measure is determined by the equation below:

$$s_{wup}(w_1, w_2) = \frac{2*depth(\ LCS\ (w_1,w_2))}{depth(w_1)+depth(w_2)} \quad (4)$$

where $LCS(w_1, w_2)$is the LCS between the words w_1 and $w_2$and $depth(w)$ is the length of the shortest path between the root and a word w.

Semantic similarity measures based on information content.

We employ three measures that are based on the IC: Resnik [17], Lin [18] and Jiang and Conrath [19]. IC is a measure of specificity of a word. High values of IC are associated with more specific concepts of words (e.g., mouse) and lower values are more general (e.g., animal). The IC is calculated from the observed frequency counts of a word in a sense-tagged corpus: a corpus annotated with WordNet senses. The IC value of a word *w* can be quantified as a negative log likelihood of the probability of that word:

$$IC(w) = -\log p(w) \quad (5)$$

The IC-based approaches operate by default using the SemCor [20] corpus, a sense-tagged portion of the Brown Corpus.

The measure by Resnik[17], denoted as s_{res} , only considers the IC of the LCS of the two compared words:

$$s_{res}(w_1, w_2) = IC\ (LCS\ (w_1, w_2)) \quad (6)$$

where $LCS(w_1, w_2)$ is the LCS between words w_1 and w_2.

The measure introduced by Lin [18] builds on Resnik's measure by adding a normalization factor consisting of the information content of the two input words:

$$s_{lin}(w_1, w_2) = \frac{2*IC\ (LCS\ (w_1,w_2))}{IC(w_1)+IC\ (w_2)} \quad (7)$$

Finally, we use the measure introduce by Jiang and Conrath[19], s_{jnc}, determined using the following equation:

$$s_{jnc}(w_1, w_2) = \frac{1}{IC(w_1)+IC(w_2)-2*IC\ (LCS(w_1,w_2))} \quad (8)$$

The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-319-08786-3_17

where the IC of a word is defined by equation (4) and where $LCS(w_1, w_2)$ is the LCS between words w_1 and w_2.

Experimental Setup

To develop our text-based recommender system and run this evaluation we used the MyMediaLite 3.07 [21] C# library on Mono architecture. We evaluate the performance of the six text similarity measures from Section 4 on our approach compared to a range of representative ratings-based approaches using two datasets.

Datasets

Table 1. Properties of the two datasets used in our experiment

Dataset	Users	Items	Training	Validation	Test set (fold size)	Sparsity
RottenTomatoes	451	1000	40371	848	21200 (848)	86.17%
Audio CDs	53060	36381	66394	1397	34925 (1397)	99.99%

RottenTomatoes Dataset.

The Rotten Tomatoes movie review website allows two types of reviewers: critics and non-critics. Critics write movie reviews professionally. Non-critics or standard users are general members of the public. The API of the platform only offers the ability to collect reviews written from critics. To avoid any violations of the terms of the service of the platform, we only used the functionality offered by the API to construct this dataset. The RottenTomatoes dataset includes critics' reviews for the Top-100 movies for the years 2001 to 2010. Each entry in the dataset consists of a user id, a movie id, a timestamp, a rating and a short text passage. All reviews having a missing rating (30% of the reviews) or a missing text passage (0.09% of the reviews) have been removed from the dataset, resulting in a dataset of 62,365 reviews, 451 users and 1000 items.

Since our goal is to improve the accuracy of ratings prediction for the standard users, rather than critics, we carried out an experiment to investigate the divergence between standard and critic's text reviews. Using 200 random standard and 200 critic reviews for the top five movies from 2010, we carried out a statistical analysis over the two sets. Results indicated that there is a Cosine similarity of 0.85 between the term frequencies of the two sets, thus indicating the high similarity of language used by critics and standard users. The similarity between two sets of 100 random reviews written by standard users is 0.96.

Audio CDs Dataset.

The AmazonProductReviews dataset, by Jindal and Liu [24], contains user-item-rating-review quadruples on different categories of items. In this experimental evaluation, we used the category Audio CDs, since this has a reasonable number of users, items and reviews and

The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-319-08786-3_17

has been used by related work [7][9].The dataset includes 102,714 reviews, 53,060 users and 36,381 items. In this dataset ratings are in a 1 to 5 scale.

Dataset Spitting Method

A common practice in the recommender systems domain is to split the dataset into three subsets: a training set, for learning the parameters of a model; a validation set, to evaluate the model over different parameter settings to derive optimum parameters; and a test set, to assess the predictive performance of the model on held-out data and thus judge over fitting of a learnt model.

For example, the dataset used in The Netflix Prize [25] consists of three splits: a training set of 95.9% of the ratings, a validation set of 1.36% of the ratings, while the remaining 2.77% of the ratings are used to form the two almost equal size test folds. Although popular, such a splitting method does not allow for statistical significance testing of the predictive performance of a model Testing the statistical significance of an evaluation is important due to the marginal increases in performance often observed in the literature, in assessing for the chance involvement in such increases and to be more confident in any improvement we find when assessing our own method.

The modified approach we apply in this experiment uses the 1.36% of the dataset for the validation [25]. However, instead of using only two test folds, we use 25 equal size test folds. Using a small number of test sets may lead to mislabeling of significant results as insignificant [28]. Our modified setup uses 64.64% of the reviews for training, 1.36% of the reviews for validation and 1.36% for each of the 25 testing folds.

Also, we preserve time ordering when splitting the dataset: the training set's reviews appeared before those in the validation set, and both training and validation contain reviews from before each of the 25 folds. A splitting method that preserves time ordering resembles, most closely, the situation of a recommender in a real system [23]. The system 'knows' only the previous reviews at recommendation time and knows nothing about the future. Cross Validation (CV) evaluation methods such as the 5-fold CV used by [7] or the 10-fold CV used by [9], on the Amazon ProductReviews dataset, introduce bias in a model by training on future results.

Ratings-based Approaches.

A common practice when evaluating the benefits of a modified ratings-based recommendation approach by incorporating text reviews is to compare the modified approach to the original ratings-based approach. For example, the TBCF approach [6] and the text reviews clustering approach to produce recommendations [5] were compared to a non-personalized baseline, and the Opinion-BMF [7] approach was compared to its ratings equivalent.

In this study, in addition to the ratings equivalent (user-kNN), we compare our approach to a range of ratings-based approaches organized into three categories:

a) Baseline: approaches that make no use of personalized information such as UserItemAverage, which makes ratings predictions based on the average rating value of an item, plus a regularized user and item bias.

The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-319-08786-3_17

b) Memory-based Neighborhood algorithms: We employ the rating equivalent of our approach user-kNN, and the Item-based k-Nearest Neighbors (item-kNN), which forms neighborhoods of similar items. We use both methods with Cosine and Pearson Correlation Coefficients similarity measures [1].

c) Matrix Factorization methods: approaches based on low-dimensional factor models. In this category we use SVD++ and BMF. SVD++ incorporates both the standard Singular Value Decomposition (SVD), representing users by their own factor representation, and the asymmetric SVD model, representing users as a bag of item vectors. We also use BMF – the standard MF method with explicit user and item biases [26].

Training the user-kNN approaches.

We trained all the approaches on the training set and then validated their performance on the validation set. During this procedure we observed that ratings-based user-kNN approaches required a different size of neighborhood (k) than the text-based user-kNN approaches to achieve their best performance. The user-kNN approaches on ratings tend to produce the lowest RMSE when using 100 or 200 neighbors (k=100, k=200), while the text-based user-kNN approaches performed better when using only the single most similar neighbor (k=1). In other words, the text-based approaches perform better when using the most proximate user in terms of sharing similar views about items, or when using a weighted average of the ratings of a large amount of users.

Intuitively, this is similar to how a person would ask for a recommendation in a real life scenario: a person interested in getting a recommendation for a restaurant will probably ask the one person whom s/he trusts most when choosing a restaurant, i.e., the one that s/he shares similar tastes and views on restaurants with. Otherwise, the person would crowdsource many opinions using social networking sites, reviewing websites, or asking people from the offline environment to get a large amount of opinions and make a final decision on which recommendation to follow. In the future, we aim to further explore this observation.

Results and Discussion

All results are reported on the test folds, which were excluded from the training process. For each of the test folds, we calculated the RMSE between the actual ratings and the predictions and averaged this over the 25 testing folds. All significant values reported were calculated using a Sign Test [22], as suggested by [23] due to its simplicity and lack of assumptions over the distribution of cases over the 25 testing folds.

The results of our evaluation, reported in Table 2, indicate that out text-based user-kNN approach performs consistently and significantly better than the ratings-based approaches over the two datasets. In the RottenTomatoes dataset, the best performing than the best of the rating based approaches item-kNN with cosine similarity which achieved a RMSE of 0.1466 between the actual and the predicted ratings.

The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-319-08786-3_17

Table 2. Mean RMSE of text and rating-based approaches over the 25 test folds for the RottenTomatoes and Audio CDs datasets (lower is better).

Rating scale	RottenTomatoes 0.0 to 1.0	Audio CDs 1.0 to 5.0
Text–based user-kNN		
Leacock and Chodorow	0.1478	1.1094
Wu and Palmer	0.1472	1.1094
Resnik	**0.1461**	1.1093
Lin	0.1469	**1.1092**
Jiang and Conrath	0.1467	**1.1092**
Word Overlap	0.1462	1.1101
Rating-based approaches		
Pearson user-kNN	0.1485	**1.1190**
Cosine user-kNN	**0.1473**	1.1263
Pearson item-kNN	0.1473	1.1130
Cosine item-kNN	**0.1466**	1.1156
UserItemAverage	0.1483	1.1398
SVD ++	**0.1467**	**1.1099**
BMF	0.1476	1.1105

In the Audio CDs dataset, the best performing text-based approaches were those using the Lin and Jiang & Conrath similarity measures. They achieved a RMSE of 1.1092, significantly better ($p<0.0001$) than the RMSE of 1.1190 of the user-kNN with Cosine similarity. In addition, it is significantly better ($p<0.0001$) than the best of the rating based approaches, SVD++, which achieved a RMSE of 1.1099. The better performance of the text-based user-kNN approach over the ratings-based user-kNN approaches and over the two datasets confirms our hypothesis that measuring similarity based on text reviews can help to overcome similarity reflection problems.

Moreover, text-based user-kNN with semantic similarity measures, particularly those using the IC, performed better than those using the simple lexical overlap. This provides some evidence of improvement when measuring text similarity using semantic similarity measures. This is also in agreement with the superior performance of IC measures in a paraphrase detection task [27] over the path based measures and other approaches including Latent Semantic Analysis (LSA).

Although the improvements of RMSE we obtain may seem small, they are significant. In addition, Koren [26] provides evidence that even a small improvement in a rating prediction error can affect the ordering of items and have significant impact on the quality of the top few presented recommendations and thus the overall performance of the recommender system.

Conclusion and Future Work

Related work has suggested using text reviews to overcome the similarity reflection problems of user-kNN by incorporating text reviews in the measurement of similarity. The

The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-319-08786-3_17

suggested approaches use the sentiment of text reviews instead of ratings [3,4] or build user profiles of aggregated feature preferences extracted from text reviews [5, 6]. We argue that using the sentiment of a text review does not overcome completely similarity reflection problems since the reasons behind a rating remain unexploited. In addition, building user profiles by aggregating the feature preferences does not respect the diversity of the users' feature preferences across items.

To overcome the above limitations, we proposed text-based user-kNN: an approach that measures the direct semantic similarity of users' text reviews on co-reviewed items to form neighborhoods of similar users and minimize RMSE in a ratings prediction task. To measure the similarity between text reviews we investigate five semantic similarity measures based on WordNet, and a simple lexical word overlap measure, through their application in text-based user-kNN. We evaluate its performance by comparing it to BMF, SVD++, user-kNN and item-kNN with Cosine and Pearson correlation and UserItemAverage baseline, on the RottenTomatoes and Audio CDs datasets. Our results show that the text-based methods produce consistently and significantly lower RMSE than the rating-based approaches over the two datasets used in this experiment. In addition, we have shown that a text-based user-kNN that uses semantics similarity measures to calculate the similarity of text reviews performs better than when using a simple lexical word overlap measure.

In our future work, we will carry out an evaluation with other text-based approaches in an items prediction task to investigate how significant our approach is to users. In addition, in the future we will investigate other techniques to further enhance the measurement of similarity between text reviews such as sentiment analysis and evaluate different combinations of text, sentiment and ratings similarities. Furthermore, we would like to investigate the use of Linked Data to identify hidden similarity between entities found in text reviews to improve the similarity reflection between users.

References

1. Herlocker, J., Konstan, J., Borchers, J.A., Riedl, J.: An Algorithmic Framework for Performing Collaborative Filtering. In: Proceedings of the 1999 Conference on Research and Development in Information Retrieval (1999)
2. Terzi, M., Ferrario, M., Whittle, J.: Free Text In User Reviews: Their Role In Recommender Systems. In: Proceedings of the 3rd ACM RecSys'10 Workshop on Recommender Systems and the Social Web, p. 45-48. ACM, Chicago, US (2011)
3. Leung, C.W.K., Chan, S.C.F., Chung, F.: Integrating collaborative filtering and sentiment analysis: A rating inference approach. In: Proceedings of the ECAI 2006 Workshop on Recommender Systems, pp. 62–66, Riva del Garda, Italy (2006)
4. Zhang, W., Ding, G., Chen, L., Li, C.: Augmenting Chinese Online Video Recommendations by Using Virtual Ratings Predicted by Review Sentiment Classification. In: Proc. Of the IEEE ICDM Workshops. IEEE Computer Society, Washington, DC (2010)
5. Chen, L., Wang, F.: Preference-based Clustering Reviews for Augmenting e-Commerce Recommendation. In: Knowledge-Based Systems (2013)
6. Musat, C. C., Liang, Y., Faltings, B.: Recommendation using textual opinions. In: Proceedings of the 23rd IJCAI pp. 2684-2690. AAAI Press (2013)
7. Pero, Š., Horváth, T.: Opinion-Driven Matrix Factorization for Rating Prediction. In: User Modeling, Adaptation, and Personalization, pp. 1-13. Springer, Heidelberg (2013)

The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-319-08786-3_17

8. Singh, V.K., Mukherjee, M., Mehta, G.K.: Combining collaborative filtering and sentiment classification for improved movie recommendations. In: MIWAI 2011. LNCS, vol. 7080, pp. 38–50. Springer, Heidelberg (2011)
9. Raghavan, S., Gunasekar, S., Ghosh, J.: Review quality aware collaborative filtering. In: Proceedings of the 6th ACM conference on RecSys, pp. 123-130. ACM, Chicago (2011)
10. McAuley,J., Leskovec, J.:Hidden factors and hidden topics: understanding rating dimensions with review text. In: Proceedings of the 7th ACM RecSys. ACM (2013)
11. Levi, A., Mokryn, O., Diot, C., Taft, N.: Finding a needle in a haystack of reviews: cold start context-based hotel recommender system. In: Proc. RecSys 2012, pp. 115–122. ACM, New York (2012)
12. Fellbaum, C.:WordNet: An Electronic Lexical Database , Mit Press (1998)
13. Pedersen, T., Patwardhan, S., Michelizzi, J.: WordNet: Similarity - Measuring the Relatedness of Concepts. In: Proc. of AAAI, pp. 1024–1025. AAAI, Menlo Park (2004)
14. Lewis, D. D., Yang, Y., Rose, T. G., and Li, F. Rcv1: A new benchmark collection for text categorization research. J. Mach. Learn. Res. 5 , pp. 361–397. (2004)
15. Leacock, C., Chodorow, M.: Combining local context and WordNet similarity for word sense identification. In : C.Fellbaum (Ed.), pp. 305–332. MIT Press (1998)
16. Wu, Z., Palmer, M.: Verb semantics and lexical selection. In: 32nd. Annual Meeting of the Association for Computational Linguistics, pp. 133–138 (1994)
17. Resnik, P.: Using information content to evaluate semantic similarity in a taxonomy. In: Proceedings of IJCAI, pp. 448–453 (1995)
18. Lin, D.: An information theoretic definition of similarity. In: Proceedings of the 15th IICML. Morgan Kaufmann, San Francisco. (1998)
19. Jiang, J. J., & Conrath, D. W.: Semantic similarity based on corpus statistics and lexical taxonomy. In: ROCLING X , Academia Sinica. Tapei, Taiwan (1997)
20. Miller, G. A., Leacock, C., Tengi, R., Bunker, R. T.: A semantic concordance. In Proceedings of the workshop on HLT , pp. 303–308, Stroudsburg, PA, USA (1993)
21. Gantner, Z., Rendle, S., Freudenthaler, C., Schmidt-Thieme, L.: Mymedialite: a free recommender system library. In: Proceedings of the 5th ACM Conference on Recommender Systems, pp. 305–308. ACM, New York (2011)
22. Demsar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research, 7, pp. 1–30. (2006)
23. Shani, G., Gunawardana, A.: Evaluating recommendation systems. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.), Recommender Systems Handbook. (2011)
24. Jindal, N. and B. Liu. Opinion spam and analysis. In Proceedings of the Conference on Web Search and Web Data Mining (2008)
25. Bennet, J., Lanning, S.:"The Netflix Prize", KDD Cup and Workshop. (2007)
26. Koren, Y.: Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: Proceedings of the 14th ACM SIGKDD, pp.426–434. ACM, New York (2008).
27. Mohler, M., Mihalcea, R.:Text-to-Text Semantic Similarity for Automatic Short Answer Grading. In: EC-ACL 2009, Athens, Greece, pp. 567–575. (2009)
28. Gunawardana, A., Shani, G.: A survey of accuracy evaluation metrics of recommendation tasks. J. Mach. Learn. Res. 10, pp.2935–2962. (2009)

The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-319-08786-3_17