User Review Incorporated Collaborative Filtering For Video Recommendation System

Mr.M.Anbazhagan[1], L.Subalakshmi[2], A.Sujatha[3], L.Shalini[4]
[1]Assistant Professor
[1, 2, 3, 4] Saranathan College of Engineering

Abstract- *A Collaborative Filtering approach is proposed here which uses user's review for producing description of items that represents a consensus of users in regard with item's features. Structured metadata is helpful for representing items have been focused earlier, whereas producing better insights about content's semantics are studied in recent approaches. Semantic analysis and natural language processing based algorithms are used to decrement such as noise, personal opinions and false information problems. When compared to recommenders related with structured data, Amazon dataset evaluated here provides improvised results.*

Keywords- Recommender Systems; Collaborative Filtering; Item Representation; Unstructured Information; Sentiment Analysis;

I. INTRODUCTION

In order to develop and enhance these systems, growing efforts are necessary along with the ever hiking data availability. The problem on information overload is dealt with this tool by an ample amount of users. Two traditional mechanisms are taken into account by literature for generating such recommendations. These are content-based and collaborative. In the first approach content-based, associations on attributes are used to perform data selection, where each item has its own characteristics. In the collaborative filtering approach, based on the user's ratings, recommendations are performed by item or user associativity.

Nowadays there is an increasing effort on consideration of additional and unstructured data produced by similar or different users while consuming the content to the further side of the traditional mechanisms for generating recommendations for a single user. For instance, to decide whether a product is worth buying or to use a particular product, user reviews are a great information sources. These information sources are checked manually before the consumption by the users. This task could be smoothened with automatic techniques by incorporating analysis into filtering process which results in better recommendations. The probability of likelihood by the users is higher when users are consensus about the product's quality.

When user-provide texts are used for representing items, a set of challenges has to be dealt with. For instance, the reviews are prone to the occurrence of noise, such as misspelling, false information, and personal opinions that are valid only for the review's author. For analyzing the text, extracting and organizing relevant information about a subject there is a requirement for natural language processing tool. Finally, another challenge is, for the generation of effective recommendations according to user's preferences, how additional; data can be applied.

So as to improvise the accuracy of recommendations, this proposes a filtering approach which takes in the consensus of user's opinion. For the extraction of candidate features and personal sentiments in accordance with each feature, reviews of variety of users are processes. Representing items which contains the most relevant features is brought by the algorithm. By putting forward all the users' sentiment towards a particular feature, simulation of consensus concept is done. For instance, an average positive sentiment towards a feature could indicate that many users admit that this feature has a positive feature. Finally, CT approach based on K nearest neighbors will use this representation later on for computing the similarity of items.

The main advantage here is that, With reference to those items he found satisfactory in the past, identical user items could be recommended, where this similarity told by the common opinion regarding the quality of different aspects related to the items.

This paper is arranged as follows: in Section II we present some related works that address user reviews; in Section III we present a simple use scenario that illustrates the goals of this approach; in Section IV we detail our work; in Section V-B we present our results and in Section VI we present some conclusions and future work.

II. RELATED WORK

Item's structured metadata is used by content-based recommenders e.g. genres and casting in a movie recommender while unstructured information are explored for item characterization. Web user's reviews provide large semantic load related to the utility of terms and preferences of review's author and hence considered valuable. Such information may be related to the item as a whole (e.g. the movie was great) or to specific features of these (e.g. the actor's performance in that movie was poor).

These reviews are used to extract feelings related to inherent characteristics of an item and are used in recent works. With the information extracted from databases available in several trusted sites, Qumsiyeh and Ng [11] proposed a system capable of generating recommendations for different multimedia items. Sentiment and degree of each aspect of an item such as genres, actors and reviews are calculated by this method. Based on the previous reviews of users and ratings, Kim et al. [7] proposed a personalized search engine for movies, called Movie Mine. By the selection of existing keywords in his previous reviews, allowing the search key to be customized for each individual, user's typed query is expanded. In the system proposed by Aciar et al. [1], for the transformation of opinions about items to ontologies, which define both the skill and knowledge of the user about an item and its characteristics, text mining techniques are applied. Recommendations are considered as the ability of the consumer, his experience and how he is interested in specific characteristics (inferred from the user query) and it is made through analysis in the instances of the ontology.

In these previous works, reviews are used in a content-based Scenario, but some other works uses textual information in the context of collaborative filtering. An approach was proposed by Lu et al. [9] that identifies topics by clustering phrases in textual comments and assigns user sentiment for them. As an aspect rating prediction, resulting in several ratings that are combined to make a prediction on the overall rating in the review, it is done later on. A restaurant review recommender was proposed by Ganu et al. [5] that perform a user soft clustering based on topics and sentiments found in their reviews. With the regular star rating system in several scenarios, it produces text based ratings and their values are compared such as by neighborhood and latent factors models with the soft clustering recommendation model proposed, those two rating systems are compared.

The technique proposed in this paper differs from the aforementioned works since it is a collaborative filtering approach that uses user reviews and sentiment analysis to solely describe items. User's preferences are mapped in this scenario by finding their top rated items k nearest neighbors (kNN) rather than mapping users' preferences through sentiment based clustering.

III. A BASIC USE SCENARIO

For the better understanding if goals, we present a simple use scenario where it is illustrated the system's usage and benefits for the user.

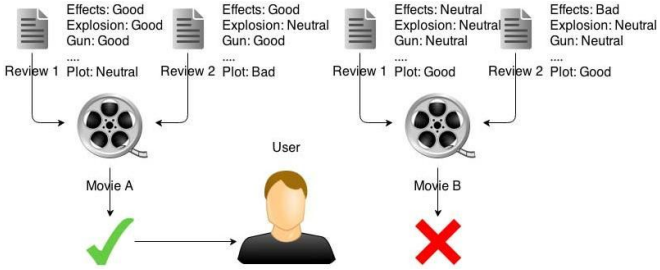

Fig. 1. An Illustrated Use Scenario

Peter enjoys watching movies that have nice effects and a lot of well performed action scenes, but doesn't care about its plot: as long as a film presents beautiful scenarios and effects, and some action such as explosions, gun shots or car chases, it could be a possible good movie for him. Recommendations are likely to be received by peter that is generated by these specific features already presented. A system in which relies on, performs the task of browsing on-line reviews to see others opinion about recommendations are given to him. The ratings peter has already given to movies that he had watched are given as input to this system and these are considered as knowledge about his preferences. A set of features with some level of appreciation (good, bad or neutral) are termed as movies which are a consensus, or average of the impressions of the reviews' authors. Figure 1 shows an example of Peter's scenario. There are two movies, A and B, each with two reviews. Movie A receives reviews that appraise its effects and action features such as explosion and gun, but also says that the plot is bad, while Movie B shows that the plot is very good but the features that Peter likes are either neutral or absent. Based on these pieces of information, the system decides that Movie A is a good recommendation for Peter, while Movie B is not.

IV. PROPOSED WORK

In order to accomplish the activity described in the above section, the approach for creating an item's representation for collaborative filtering activity with text reviews are proposed. For producing a vector-based item representation where individual position reflects a feature (plot, explosion, etc.), and it's score denotes the overall

sentiment (positive, neutral, negative) towards this, NLP tools are used.

The whole process of generating the item representations are described in this section. The natural language processing tool we use (Section IV-A) and the sentiment analysis tool (Section IV-B) are introduced which provides a score that represents sentiment of a sentence in the text. After this, research settings and how those tools are used for producing feature vectors are shown. And finally, the recommendation algorithm used for testing our approach is presented.

A. Stanford CoreNLP

A set of tools to process texts are provided by The Stanford's natural language processing tool, called Stanford CoreNLP1.Raw English text is taken as input and an entire structured analysis of most common NLP routines are produced. For each text file, it's output is a graph-form XML file with all previously set appropriate annotation. Stanford CoreNLP is a free, open-source framework composed in Java. It requires Java 1.6+ to function.

For several language analysis tools, Stanford CoreNLP is an integrated framework known as annotators. Tokenizer, sentence splitter, lemmatizer, POS tagger named entity recognizer, parser, co-reference resolution system and the sentiment analysis tool are few relevant annotators that this tool comprises of.

For our proposal, the implementations of default annotators are used that are applied in the reviews dataset. Generation of processed texts is through the usage of annotators such as Tokenizer, sentence splitter, lemmatizer, POS tagger named entity recognizer, parser and sentiment analysis tool.

B. Stanford Sentiment Analysis Tool

Investigating the isolated words most sentiment prediction algorithm compute scores and then the sum of scores are performed. Approach of Stanford CoreNLP sentiment analysis tool [13] is unique since it employs deep learning model. Here in sentence level, sentiment structure is used for providing sentiment analysis. In this sense, sentiment is computed by this tool based on the meaning that each word comprises in phrases.

Since movie features are nouns such as script and effect, and nouns are generally neutral sentiment words, using a sentiment analysis tool is a good idea that relies on sentence level scoring. Polarity assigned by checking sentiment in isolated words, would lead us to several features with a neutral (zero) value. On the basis of the sentiment in the sentence the score is assigned that contains the word. Another justification is that, with negation sentences, models could be dealt which often presents a set of negated positive words.

In this work, sentiment analysis tool is applied for producing scores for features based on the sentences' sentiments .With respect to each of the features, averages of all reviews' sentences' and sentiments are performed. In this way, an users' consensus towards a specific feature in an item is achieved.

C. Data Acquisition

For this work, video recommendation domain is chosen as an area to apply our research, as this domain consists of vast datasets and information available on the internet. In this way, for producing a representative set of items' descriptions, we have used Amazon instant video prime dataset.

The website which provides video recommendations for users is the Amazon. The Amazon dataset contains user reviews and ratings. We have removed movies that are duplicated or unidentifiable (movies without names). Each review contains a number of headers and a text body. The headers include movie ID, user ID, review date, summary, which is a one-line summary in natural language text written by the user, and a rating, which is a user-specified number ranging from 1 (awful) to 10 (excellent). The text body is the user's comments on the movie

D. Data Modeling

Productions of items' representation vectors are by applying few heuristics from the Stanford CoreNLP framework to extract the features and their corresponding sentiment from its output.

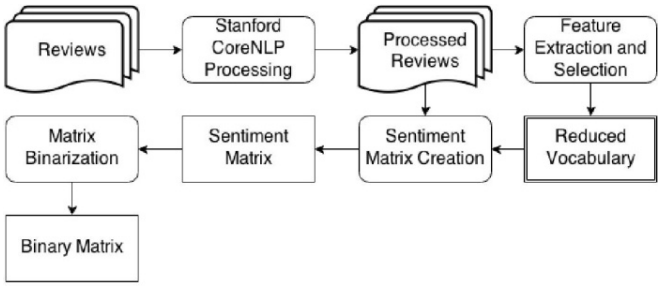

Fig. 2. The process for generating item representations.

Initially, processing of reviews used the Stanford CoreNLP tool. According to Subsection IV-A, tokenizer,

sentence splitter, lemmatizer, POS tagger, parser and the sentiment analysis tool are the used annotators. With the whole text divide into sentences, XML file was produced for each text document. Sentiment and a parser were common in each of the sentences and tokens are split from them and lemma is consisted in each of them along with its pos tag.

In the next step, for creating a set of candidate features, NLP tools' output is provided which is named as vocabulary. As explained later, the vocabulary is filtered and the dataset's items' features are there in the resulting set of terms. By the usage of tokens, features are selected, and then information retrieval and heuristics are employed for reducing the number of keywords such as stop-word removal. Movie features are often nouns such as "effects", "plot" or "direction" since one data domain is particularly dealt with. For selecting our features, only POS tagging is relied on. Candidate features are selected which have the part-of-speech tag corresponding to singular and plural nouns.

Misspelled words are possibly tagged as nouns in the case of the Stanford CoreNLP POS tagging. Given that noise is just a great number of candidate features led word would generate in our item vs. feature matrix a column where only one item have a value that differs from zero which is the item that contains by which accuracy of the recommender systems are affected by. For instance, a misspelled word. This generates an effect that, our matrix will be much more sparse that the capacity for the generation of good recommendations will become lower.

Based on an items' frequency of a particular feature, feature selection is made for the reduction of the count of candidate keys and consequently the sparsity of the resulting sentiment matrix will feed our recommender algorithm. Let F be the vocabulary and I the set of items, the item frequency IF_j of a feature f is given by Equation 1:

$$IF_f = k_{if} ,$$

Where k_{if} is equal to 1 if an item i contains that feature or 0if it doesn't.

For deciding whether the feature is maintained or removed from our vocabulary, the IFf is then compared with a threshold t. The feature should be maintained in the vocabulary for the value of the TF_f is higher than t. Four vocabularies have been produced by setting thresholds t1 = 1, t2 = 30, t3 = 100 and t4 = 200.for reducing noise, thresholds are used and consequently, the dimensionality and sparsity of the matrices. Experiments are carried out with all the four vocabularies as described in Section V.

The next step is the production of sentiment matrix. The overall sentiments of the item's reviews are represented by each position in accordance with a feature .Then sentences are grouped related to them in reviews for each feature in each item. For achieving it, their sentiment scores are analyzed. Sentences are classified into five sentiment levels: Very Negative, Negative, Neutral, Positive and Very Positive by the Stanford CoreNLP sentiment analysis tool. The classification is converted into a{-2,+2} rating system and assigned as a feature score the average rating of the related sentences. The thing indicated by zero means a feature is neutral or no feature is portrayed by the item.

Our sentiment matrices have to be converted into a particular format since for item descriptions, binary matrices in the form of indexes are accepted by the recommender algorithm. To indicate that only the positive aspects of items are represented, at first we need to turn only the positive values to 1.On the other hand, Zero is assigned to all the negative and neutral sentiments. But this doesn't provide good results as indicated in the evaluation section of this paper (Section V-B): all the negative part of the sentiment matrices are missed. We divide a feature column into two columns: positive values were set as one in the first column, while negative values were set as one in the second column for incorporation of both positive and negative aspects to our matrices .In the matrices binarization process, an interval a is used for adjusting the relevance of the intensity of the sentiment level. four intervals:a1 = {-0.1, 0.1}, a2 = {-0.5, 0.5}, a3 = {-1, 1}, a1 = {-1.5, 1.5} are set and in recommender's accuracy their impact is evaluated. Figure 3 gives an example of this process, using α1.

	Feature A	Feature B
Item A	0.667	0
Item B	-0.1	1.5
Item C	0.5	-0.1

↓

	Feature A +	Feature A -	Feature B +	Feature B -
Item A	1	0	0	0
Item B	0	1	1	0
Item C	1	0	0	1

Fig. 3. Matrix binarization with α set to *{*-0.1, 0.1*}*

E. The Recommender Algorithm

A collaborative filtering algorithm based on k nearest neighbors is used in evaluating our item vs. feature sentiment

matrices where the correlations among items using their attribute vectors are computed. We adopted the MyMediaLite Recommender System Library5 [4], whose algorithm's implementation is called ItemAttributeKNN. Several algorithm implementations are contained in MyMediaLite which is an open source library regarding the CF scenario. It can be used by anyone and algorithms could be developed as it is an open source library. Many evaluation routines are also contained in MyMediaLite such as the MAE, RMSE, Precision@K and MAP among others implementation of data sampling such as cross-validation is also given.

The ItemAttributeKNN algorithm is identical to traditional item-based kNN, but instead of using rating vectors for computation of items' correlation, by measuring the distance, the similarity of items is experienced .Cosine measure is used for this work since it's the most often used measure applied to vector representations [2]and the distance is based on the angle between two instance's vectors in place of its absolute distance.

By the computation of weighted average of the ratings of an item's k nearest neighbors, prediction of an item is done by a fall-back prediction if no neighbors are found, e.g. using user and/or item biases.

V. EXPERIMENTAL EVALUATION

An experimental evaluation is presented in this section that has been conducted on the basis of our proposed items' representation technique based on users' reviews. This paper uses ItemAttributeKNN recommender algorithm, provided by the MyMediaLite Recommender Systems Library. A rating matrix and a item attribute binary matrix are taken as an input to the algorithm. A routine that allows the rating matrix to be divided in n parts is also used which makes n-fold cross-validation for testing and validation possible. A 10-fold crosses validation for each of the item attribute matrices are decided to be performed. The precision at 10 (prec@10) and the mean average precision (MAP) are chosen as evaluation metrics. We can see a brief description of these metrics in the following section, while Section V-B shows experimental results.

A. Evaluation Metrics

The prec@10 and MAP measures are used for evaluation. MyMediaLite library already has both these metrics implemented and are useful in results analysis in rank format (item recommendation instead of rating prediction) these are known to be good for evaluation as a rank of k most relevant recommendations are produced. On considering the size of the whole produced rank, precision measure is characterized. The number of relevant items returned in relation to a small sample of the rank: the k first items are measured at precision k. It can be seen as:

$$prec@k = \frac{\#(\text{retrieved items in } K)}{K}$$

This measure is used with k=10.A brute percentage of relevant items in a rank is given which is a drawback of this measure but their relative position on the rank is not considered. In this sense, many relevant items can be returned by one, but they may appear only at the lists' bottom and in user perspective nothing is interesting: only the few top results are likely to be seen by the user and the rest are ignored.

With more relevance given for a ranks' early items, this problem is solved by MAP measure corresponding to the average of j queries, MAP is a measure that produces a value , a rank and a score are produced by each query that is the average of different n precision levels. Formally, let the set of relevant items for a query qj ? Q be {i1. . . imj}, and Rjk be the set of results returned from the first item until the ik item, then the MAP can be measured as [10]:

B. Evaluation Results

Structured metadata such as actors, directors, genres and writers are collected for constituting baseline for comparison. Four binary metrices are constructed with a metadata comprised in each of them, an item has a metadata if its value is 1 and an item doesn't have if its value is 0.Prec@10 and MAP for these baseline matrices are presented in Table 1.

TABLE I. THE PRECISION AT 10 AND MAP RESULTS FOR THE BASELINE METADATA MATRICES.

	prec@10	MAP
Actors	0.01862	0.02598
Directors	0.03818	0.03476
Genres	0.03622	0.0336
Writers	0.02905	0.03232

Four vocabularies along with four different thresholds are constructed for our approach: 1, 30, 100, 200.A sentiment matrix for each of these vocabularies are constructed from which five binary matrices are generated:

one in which only the positive aspects are considered, and in other four, both positive and negative aspects with a intervals. Table II and Figures 4 and 5 present all the results obtained for prec@10 and MAP for these matrices.

It is possible to note that our first approach which used only positive aspects, except for the last threshold (200) the presented results are better than our baseline but they weren't so expressive. Results produced could be much better for some of the α intervals that are set by us using both negative and positive parts of our sentiment matrix (α1, ..., α4).The results showed that the obtained results are better if the interval was tighter. In particular, for α3 results are worser than our first approach, especially for our shorter vocabularies (those with higher threshold).The tightest α1 is chosen as the optimal interval in this sense.

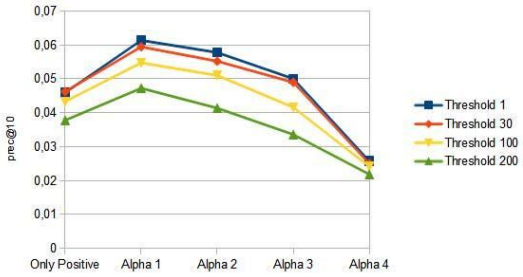

Fig. 4. Graphic comparing the precision at 10 of the proposed item attributes matrices

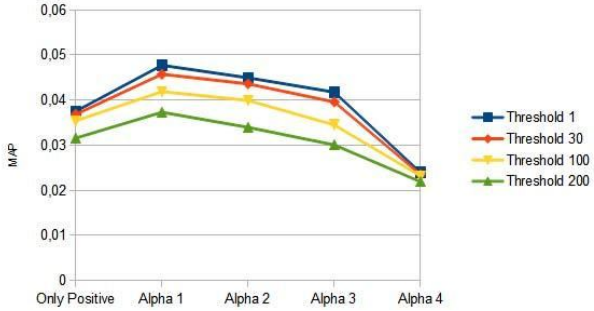

Fig. 5. Graphic comparing the MAP of the proposed item attributes matrices.

The best was achieved from the first vocabulary from all the obtained results; they have significantly higher matrix size and sparsity. The second vocabulary achieved 0.002 points less than the first in both prec@10 and MAP measures, which is a very small difference. The first vocabulary is much bigger than the other:13,000 features are contained in it other has over 3,000.Consequently,Using bigger number of features doesn't have worth, computer resources are saved by smaller dimensionality. Optimal model is selected with threshold of 30 with α1 matrix. As it can be seen in Figure 6, the results obtained by the optimal model outperform in almost twice the best baseline result. For our setting, it was worth noting that threshold 30 was optimal. Some previous evaluation would be necessary, with a different amount of items or reviews per items, For instance, we will have smaller threshold if the review set is smaller which will have fewer words. On the other hand, it can be seen that the tightest the interval α is, the less sparse is the binary matrix, which will help to increase the accuracy of the recommendation.

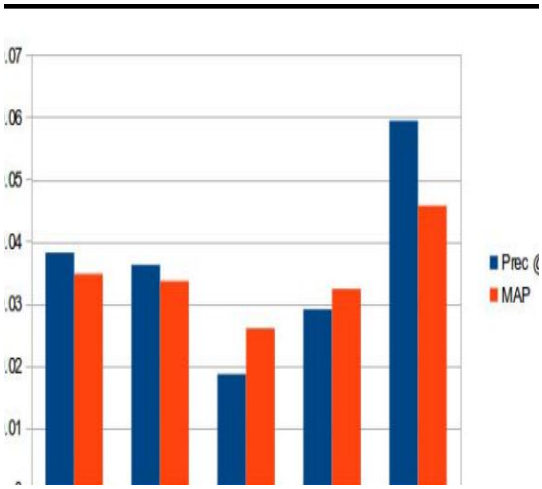

Fig. 6. Graphic comparing the precision at 10 and MAP of the most significant result of our approach

VI. CONCLUSION AND FUTURE WORK

In this paper, a collaborative filtering approach is proposed where item descriptions are produced using user's reviews that represent a consensus of users regarding items' features. On the basis of structured metadata, technique's description was presented, along with an experimental evaluation that compared the proposal with baselines. It was found that baseline was outperformed by our proposed approach from our evaluation but still improvements can be made in the results. Our experiment's drawback is that sensitive number reviews are not there for some items – some had one or two, while others didn't have reviews at all. In our future work, different review bases are used for suppressing this problem. Another troubling drawback is the usage of binary aspect of the recommender algorithm. Much of the semantics carried are lost by the transformation of sentiment matrix to binary matrix. Our future work is also left for extending this algorithm's implementation such that it accepts multivalued matrices as each item attribute's weight.

REFERENCES

[1] S. Aciar; D. Zhang; S. Simoff and J. Debenham. Informed Recommender : Basing Recommendations on

Consumer Product Reviews. *IEEE Intelligent Systems*, v. 22, n. 3, p. 39–47, 2007.

[2] G. Adomavicius and A. Tuzhilin. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. *IEEE Transactions on Knowledge and Data Engineering*,v. 17, n. 6, p. 734–749, 2005.

[3] C. C. Aggarwal and C. Zhai. A Survey of Text Clustering Algorithms.In: C. C. Aggarwal and C. Zhai, eds. *Mining Text Data*, Springer US, p. 77-128, 2012.

[4] Z. Gantner; S. Rendle; C. Freudenthaler and L. Schmidt-Thieme. MyMediaLite: A Free Recommender System Library. In: *Proceedings of the 5th ACM Conference on Recommender Systems (RecSys 2011)*, 2011, p. 305-308.

[5] G. Ganu; Y. Kakodkar and A. Marian. Improving the Quality of Predictions Using Textual Information in Online User Reviews. *Information Systems*, v. 38, n. 1, p. 1-15, 2013.

[6] M. Hu and B. Liu. Mining and Summarizing Customer Reviews. In:*Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2004, p. 168-177.

[7] H. W. Kim; K. Han; M. Y. Yi; J. Cho and J. Hong. MovieMine: personalized movie content search by utilizing user comments. *IEEE Transactions on Consumer Electronics*, v. 58, n. 4, p. 1416-1424, 2012.

[8] P. Lops; M. Gemmis and G. Semeraro. Content-based Recommender Systems: State of the Art and Trends. In: F. Ricci; L. Rokach; B. Shapira and P. B. Kantor, eds. *Recommender Systems Handbook*, Springer US, p. 73-105, 2011.

[9] Y. Lu; C. Zhai and N. Sundaresan. Rated Aspect Summarization of Short Comments. In: *Proceedings of the 18th International Conference on World Wide Web*, 2009, p. 131-140.

[10] C. D. Manning; P. Raghavan and H. Sch¨utze. *Introduction to Information Retrieval*, Cambridge University Press, 2008.

[11] R. Qumsiyeh and Y.-K. Ng. Predicting the Ratings of Multimedia Items for Making Personalized Recommendations. In: *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2012, p. 475-484.

[12] F. Ricci; L. Rokach and B. Shapira, Introduction to Recommender Systems Handbook. In: F. Ricci; L. Rokach; B. Shapira and P. B. Kantor, eds. *Recommender Systems Handbook*, Springer US, p. 1-35, 2011.

[13] R. Socher; A. Perelygin; J. Wu; J. Chuang; C. D. Manning; A. Y. Ng and C. Potts. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 2013, p. 1631– 1642.

[14] R. Socher; J. Bauer; C. D. Manning and A. Y. Ng, Parsing With Compositional Vector Grammars. In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, 2013. toutanova2003

[15] K. Toutanova; D. Klein; C. D. Manning and Y. Singer. Feature-rich Part-of-speech Tagging with a Cyclic Dependency Network. In: *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, 2003, p. 173–180. 209