Originally published in: Cheung, David/Song, Il-Yeol/Chu, Wesley/Hu, Xiaohua/Lin, Jimmy/Li, Jiexun/Peng, Zhiyong (Eds.):
TSA '09: Proceedings of the 1st international CIKM workshop on topic-sentiment analysis for mass opinion. Hong Kong, China, 6 November 2009.
– New York: Association for Computing Machinery, 2009. Pp. 57-64.
DOI: *https://doi.org/10.1145/1651461.1651473*

Niklas Jakob
Technische Universität Darmstadt
njakob@tk.informatik.tu-darmstadt.de

Stefan Hagen Weber
Siemens AG, Corporate Technology
stefan_hagen.weber@siemens.com

Mark-Christoph Müller
Technische Universität Darmstadt
chmark@tk.informatik.tu-darmstadt.de

Iryna Gurevych
Technische Universität Darmstadt
gurevych@tk.informatik.tu-darmstadt.de

Beyond the Stars: Exploiting Free-Text User Reviews to Improve the Accuracy of Movie Recommendations

ABSTRACT

In this paper we show that the extraction of opinions from free-text reviews can improve the accuracy of movie recommendations. We present three approaches to extract movie aspects as opinion targets and use them as features for the collaborative filtering. Each of these approaches requires different amounts of manual interaction. We collected a data set of reviews with corresponding ordinal (star) ratings of several thousand movies to evaluate the different features for the collaborative filtering. We employ a state-of-the-art collaborative filtering engine for the recommendations during our evaluation and compare the performance with and without using the features representing user preferences mined from the free-text reviews provided by the users. The opinion mining based features perform significantly better than the baseline, which is based on star ratings and genre information only.

Categories and Subject Descriptors

I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*Text analysis*; H.2.8 [**Information Systems**]: Database applications—*Data Mining*

General Terms

Algorithms, Measurement, Experimentation

Keywords

Opinion Mining, Sentiment Analysis, Collaborative Filtering, Recommendation System, Multi-Relational Learning

1. INTRODUCTION

One of the key characteristics of Web 2.0 is that it allows internet users to share with other users their viewpoints and opinions about almost everything. Hearing another person's substantiated opinion can be of practical benefit when it comes to deciding whether or not to invest time, money or effort into something. This is one of the driving forces behind the increasing success of community web sites which allow registered users to write and read reviews about commercial products such as books, music, movies, or consumer electronics devices such as e.g. digital cameras or cell phones. User ratings often consist of a free-text review and an overall rating. The present paper focuses on the domain of movies, and in this domain, the overall rating often comes in the form of a *star* rating. Collected user ratings can be organized by movie and presented to users who are interested in other users' opinions about a particular movie. Increasingly often, the data is also used for the creation of personalized recommendations, in which users are proactively presented with movies which they probably like. Most recommendation systems only take into account the obligatory star ratings and some simple descriptive movie features (e.g. genre) [21, 25] and leave completely unused the wealth of information that is included in the free-text reviews.

In opinion mining, a lot of work has already been done on extracting fine-grained opinion expressions from free text [10, 18, 26]. It is consequential, therefore, to bridge the gap between opinion mining and recommendation systems and to go beyond the information conveyed by the star ratings by also exploiting free-text user reviews for recommendation. We propose to do this by employing phrase-level opinion mining on free-text movie reviews for the identification of positively and negatively opinionated user statements, and by incorporating this information into the state-of-the-art recommendation system HYRES [14]. Opinionated user statements consist of the opinion-bearing expression (e.g. an adjective like "poor" or "beautiful") and the opinion *target*, i.e. what is being commented on. Since there are no constraints to what aspect of a movie users can comment on, we have to somehow condense the information before it can be incorporated into the recommendation system. We do this by mapping each automatically extracted opinion target to one or more pre-defined descriptive categories corresponding to movie-related concepts such as "acting" or "soundtrack". We use the term *movie aspect cluster* to refer to the result of these mappings. We accumulate all opinionated statements for each movie aspect cluster and provide them to the recommendation system together with the original star rating. The recommendation system is then used as a black box for extrinsically evaluating the effect of the automatically extracted information.

The fundamental rationale of our approach is that two important types of information can be extracted from the free-text reviews: 1) The correlation of the overall star rating with the individual aspect-related opinions shows the influence on the star rating that a given movie aspect has for a user, and 2) the overall number of opinions regarding a certain movie aspect cluster reveals how important that aspect is to a user. We argue that it is desirable, e.g. to also recommend movies with only a mediocre star rating to a user, if they are rated well regarding one or more aspects which are of high interest to that user. Vice versa, a well-rated movie should not be penalized for poor ratings regarding aspects which are known to be of low importance for a given user.

The remainder of this paper is structured as follows: Section 2 reviews related work from the areas of opinion mining and recommendation systems. Section 3 describes our newly collected data set, while Section 4 provides details about the employed recommendation system. Section 5 deals with the opinion mining and opinion target mapping process. This section also discusses some different approaches for creating the required movie aspect clusters. Section 6 provides descriptions of our experiments and a discussion of the results, while Section 7 presents conclusions and future work.

2. RELATED WORK

Our work presented in this paper can be classified as opinion mining or sentiment analysis at the phrase / expression level. We aim at extracting opinion expressions including concrete targets (in our case movie aspects) and determine their sentiment polarity. This was most prominently done on product reviews in previous research [6, 10, 12, 18, 24]. The detection of the product features mentioned in the reviews is similar to the detection of movie aspects in our case, which was adressed with supervised and unsupervised methods. As we present an unsupervised approach in this work, we will focus on such methods in our analysis of the related work. The unsupervised approaches can be separated into three groups: 1.) Approaches which use pre-built knowledge bases to identify the product features [10]. Depending on the overall tasks these can be flat lists or structured resources such as taxonomies or ontologies. Such approaches typically identify the features with high precision but are prone to having a low recall, since customers are free to comment on whichever feature they consider mentionworthy in their reviews, which is usually not foreseeable in advance. 2.) Approaches which perform a statistical analysis of the review corpus [12], in some cases with additonal resources as the web [18] or a general language corpus [24]. The goal of a statistical analysis is to identify the relevant or salient terms in the review corpus. Theoretically they can extract any features mentioned but the statistical approaches also have some drawbacks as shown in [8]. 3.) A combination of the previous two variants - e.g. enriching a pre-built knowledge base by searching for certain phrase patterns or named entities in the review corpus [6] or the web. However, there are also approaches which aim at a linguistically motivated analysis of the grammatical structure of a sentence in order to identify possible opinion targets. Such methods were employed on newspaper texts [13] as well as movie reviews, for which the results were better than a statistical analysis [26].

A second aspect which we cover in this work aims at clustering the identified opinions by topics. In previous research such a clustering was employed in order to group opinions regarding the same target and sentiment orientation together [17]. Such a topic clustering can also be employed in order to separate documents from different domains and cluster the opinions regarding possible subtopics therein [22]. We perform the movie aspect clustering in order to create usable input for the recommendation system, but it can also be used to create a more useful output of an opinion mining system for the end-user by creating summaries of the reviews [3].

Various technological approaches of recommendation systems have been described and compared in detail, e.g. in [5, 19, 11]. All the described predictive models focus on a single relation type (*rates*) between two entity types (*user*, *item*). Matrix factorizations such as Singular Value Decomposition (SVD) have recently been applied to relation prediction. The maximum margin matrix factorization (MMMF) introduced in [20] is a matrix factorization approach based only on the known matrix entries. Unfortunately, the MMMF model is hardly scalable. A way to make the model more scalable is to minimize the objective by using gradient descent methods. In [21], one of the favoured approaches in the Netflix Prize[1], a simple gradient descent method was applied. Recently some unsupervised approaches [16, 15] have been proposed to deal with graph clustering problems on multi-relational domains. Lippert and Weber [14] introduced a multi-relational matrix factorization (MRMF) which is an extension of low-norm matrix factorization to multi-relational domains where the involved relation types are ususally highly correlated.

To our best knowledge there is only one approach of integrating opinion mining with a recommendation system described in the literature [1]. However, the case study presented requires users to formulate their demands in the form of a query, which is then matched to opinions uttered towards the respective aspects in other users' reviews. The present work, in contrast, strives to extract user preferences automatically from ratings and existing free-text reviews.

3. IMDB DATA SET

Although several datasets for the evaluation of recommendation systems are available (e.g. MovieLens[2], Netflix,

[1] http://www.netflixprize.com/

[2] http://www.grouplens.org/node/73

BookCrossing[3], Jester Joke[4]), they only provide numerical or star ratings and no additional free-text reviews. Since we are primarily interested in the effect of free-text reviews on recommendations, we had to create our own data set. We extracted a raw data set containing the ratings and corresponding reviews of approx. 1000 random users from the Internet Movie Database (IMDB)[5]. In the IMDB, each rating is on the scale from one to ten stars, and according to IMDB policy, free-text reviews must have at least ten lines and at most 1000 words. The IMDB website recommends a length of 200 to 500 words. Typical for fan communities, some users contribute only a few reviews, while others contribute a lot. The same applies for the movies - some are rated by many users, some only by a few. In order to enhance the collaborative effect, we removed from the raw data set all reviews regarding movies with less than ten reviews. Some statistics on the raw and the reduced data set are given in Table 1. This also drastically reduced the percentage of users with only a few reviews: In the raw dataset as many as 52% of the users wrote less than five reviews, while in the reduced dataset this share decreases to only 11%. This improved the sparseness from 0.32% to 3.82%.

Table 1: Data Set Statistics

	Raw	Reduced
# Reviews	136710	53112
Avg. Sentences per Review	13.9	15.2
Avg. Tokens per Sentence	24.2	23.6
# Movies	41288	2731
# Users	1030	509
# Users (< 5 reviews)	541	57
Sparseness	0.32%	3.82%

4. HYRES RECOMMENDATION SYSTEM

Recommendation systems are algorithms which attempt to predict items (e.g. movies, music, books) in which a user may be interested, given some information about the item or the user's profile. Content-based algorithms only use item information. Items are recommended that are most similar to the items the current user likes. However, the item description cannot capture all relevant aspects of the item and the user's perception of it (e.g. mood). Furthermore, following content-based recommendations the user will stick to his usual preferences. Only items will be recommended that are similar to those already rated. Collaborative filtering (CF) overcomes this limitation by making use of the user's personal preferences and information, e.g. previously bought items, ratings or contacts. CF algorithms make use of the collaborative effect and recommend items that have been highly rated by likeminded users.

These two complementary recommendation approaches are combined in the hybrid and platform independent framework HYRES (HYbrid REcommendation System). This platform uses statistical machine learning methods for automatic personalized recommendations in multi-relational domains, e.g. service recommendations [2]. HYRES implements the MRMF algorithm [14]. Apart from its high accuracy, the system also exhibits a high performance even on huge data sets (e.g. the NetFlix data set). We chose HYRES as the basis for our experiments because it can easily handle more dimensions and any number of entities and relations. Furthermore, the extension of HYRES is straightforward.

4.1 Collaborative Filtering Setup

A natural way of representing relational data is the entity relationship model. Our example data set can be described as an ER diagram as depicted in Figure 1. Involved entities are *User*, *Movie*, *Genre* and *Count*, where *Count* denotes the discretized average number of given opinions of a certain type for a single movie. For example, "I like actor X" and "I dislike actor Y" are both opinions of opinion type *acting.* Relations that have to be considered in the CF model are "User rates Movie", "Movie has Genre", and for each opinion type N the n-ary relation, "User has opinion N about Movie averaged from Count opinions of that type". Several aspects had to be taken into account to transfer the ER diagram to a multi-relational CF model. The model shown in Figure 2 illustrates the full CF model. The n-ary opinion type relations had to be decomposed by reification since MRMF only handles binary relations. Each opinion type relation is modeled as an entity by itself. This corresponds semantically to split the *User*s into their different roles: the rating-role and a role for each opinion type. However, by modelling the *User*s as separate entities, the knowledge about the same identity of individual users in different roles is lost. This is compensated by introducing a new sparse relation, *sim*, between the user roles, mapping individual users on their different roles. The model contains the following five relation types modeled as bipartite adjacency matrices. 1.) The sparse matrix **rates** $\in \mathbb{R}^{u \times m}$ contains the overall star rating of users for movies (1 to 10), where u is the number of users and m is the number of movies. 2.) N sparse matrices $\mathbf{hasOp}_N \in \mathbb{R}^{u \times m}$ contain the averaged values for opinion type N of users for movies (1 to 10). 3.) The dense binary matrix **has** $\in \{0,1\}^{m \times g}$ maps movies on genres, where g denotes the numer of genres. All known genre relations are labeled with 1 whereas unknown genre affiliation is modeled as 0. 4.) N dense binary matrices $\mathbf{hasCount}_N \in \{0,1\}^{u \times c}$ map the averaged opinion on the discretized number of given opinions c. 5.) N sparse matrices $\mathbf{sim}_N \in \mathbb{R}^{u \times u}$ map the similarity between users in their different roles. Note that only the matrix diagonal is filled with the known similarity of 1 and unknown similarities are modeled as 0.

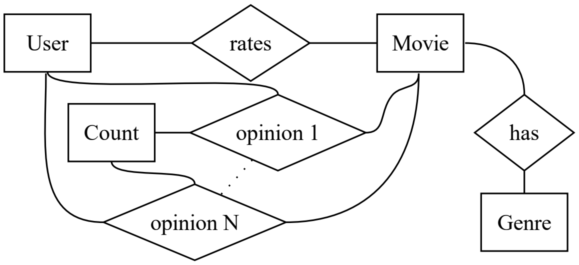

Figure 1: ER diagram of data set

The above entities and relations are supplied to the system as feature vectors. Extending the CF model, i.e. adding new features to the model, only requires manually editing

[3] http://www.informatik.uni-freiburg.de/~cziegler/BX/
[4] http://www.ieor.berkeley.edu/~goldberg/jester-data/
[5] http://www.imdb.com

the model file, which is a one-time effort, and appending the values for the new features to the existing feature vectors. In our experiments we also investigate sub-models containing only a subset of the full model. The smallest sub-model, "User rates Movie", consists of 2 entities and 1 relation, whereas the full-blown CF model for 20 opinion types results in 24 entities and 62 relations. In order to make all models comparable we abstained from optimizing free parameters for each model but fixed the free parameters to reasonable values acceptable for all models. Free parameters include learning rate, regularizer rate and the maximal number of learning epochs. For more information see [14].

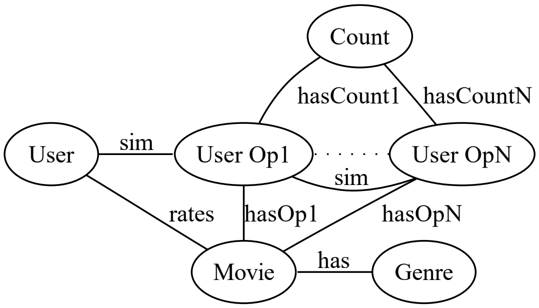

Figure 2: HYRES CF model

5. FREE-TEXT REVIEW ANALYSIS

In the following Section 5.1 we will describe the three different approaches to movie aspect identification and clustering that we tried, and in Section 5.2 we will elaborate on our opinion extraction pipeline.

5.1 Movie Aspect Identification & Clustering

As already outlined in Section 1, the set of movie aspects that a user can comment on in his or her review is in principle unconstrained. In order to integrate the opinions expressed in a given review into the recommendation system, they need to be represented in a more compact way. We do this by mapping each identified opinion target to one or more pre-defined movie aspect clusters, and by computing several overall numerical values for each cluster. The makeup of the movie aspect clusters thus has a major impact on the recommendation system, which is why we tried several ways of creating them.

5.1.1 Manual Clustering

In a first attempt, we created five medium-sized movie aspect clusters manually. For this, we read the Wikipedia article on "*Film*" and identified the following key concepts regarding film: "*acting*", "*storyline*", "*cinematography*", "*soundtrack*", and "*production*". By analyzing the corresponding articles, we then identified for each category between five and 20 pertinent terms which we considered to be potential opinion targets. Excerpts of the resulting movie aspect clusters can be found in Table 2. We intentionally left out general terms such as "*movie*" or "*film*", since opinions regarding these terms do not refer to a certain movie aspect, but express the user's opinion of the movie as a whole. This information, however, is already given by the star rating. We treated opinions regarding individual actors and directors as relating to the concepts "*acting*" and "*cinematography*", respectively. For this, we extracted 11015 actor names from the Wikipedia categories "American actors" and "American film actors", and 1171 director names from the category "American film directors".

5.1.2 Semi-automatic Clustering

The manual identification and clustering approach described above has two major disadvantages: Manually selecting terms for each of the five movie-related concepts by inspecting a resource such as Wikipedia is a very time-consuming task, and the recall can still be poor because the resource might not cover all of the terms that are used in the review corpus. We therefore also tried a semi-automatic clustering approach which used as input only the five manually defined key concepts. The semi-automatic clustering first identifies the potential movie aspects among all opinion target candidate terms. Some of these are then mapped to exactly one of the five categories, while others remain unmapped. Opinion target candidate terms are all terms in the review corpus (Table 1) after opinion word (see Section 5.2) and stop word removal. We based the clustering on the notion of *semantic relatedness* between a candidate term and a cluster's key concept. Several approaches for measuring the semantic relatedness of terms have been suggested in the past of which Explicit Semantic Analysis (ESA) on Wikipedia represents the state-of-the art in several tasks [9]. We computed the semantic relatedness of the lemmatized candidate term in the corpus to each of the five cluster key concepts. In doing so, we had to disambiguate the originally selected "*production*" to "*film-production*". Ideally, we would have used the variant "*film production*", but the implementation of the ESA algorithm available to us can only handle single words and not phrases. Each movie aspect identified was then mapped to the cluster to which it had the highest semantic relatedness score. For each of the resulting five movie aspect clusters, we only retained the 20 highest-ranked terms for our experiments. Zhuang et al. [26] report that their movie feature classes mostly contained less than 20 words. This is also the case for our manually created clusters and suggests that a cluster size of 20 seems reasonable. For reasons of space, Table 3 shows only the top 10 terms. We observe that for four of the five clusters the aspects created by the manual and the ESA approach are very similar. In total, there is an overlap of 16 movie aspects between the ESA and the manually created clusters, with the "*production*" / "*film-production*" clusters having zero overlap. This might be due to the fact that the concept "*film-production*" does not occur in Wikipedia as often as the other four concepts. Therefore the ESA algorithm also rates terms which are specific to those fewer articles as semantically highly related. This is probably also the reason why the name "Asheville" (a city) is considered to be so highly related.

5.1.3 Fully Automatic Clustering

In this approach we wanted to completely eliminate the manual effort in both the identification of key concepts in the movie domain and movie aspect clustering. Since it allows to control the number of clusters produced and since it has been successfully applied to several tasks in the past, we decided to employ Latent Dirichlet Allocation [4] for the clustering. We again removed all words in our opinion word lexicon from the corpus before clustering it. We then employed the Mallet toolkit[6] to perform the clustering on our lemmatized corpus, using Mallet's built-in stop word filtering.

The clusters created by the LDA approach (Table 4) exhibit a much finer granularity regarding the represented concepts, but this was to be expected as the number of clusters is much higher. When analyzing the terms in the clusters, one can observe that the LDA approach models the domain on different levels: On the one hand there are clusters which contain generic terms regarding the movie domain, while on the other hand there are also clusters which represent certain genres (horror, science-fiction, war) and even individual movies (James Bond, Hitchcock, Dracula). This outcome seems promising regarding the employment in the collaborative filtering, as such clusters could help to model the users' preferences on several levels of granularity. As it is common for LDA outputs, the same term can appear in several clusters (e.g. "performance", "film"). This requires a special strategy during the integration in the recommendation system (cf. Section 6.1).

Table 2: Manual Cluster Excerpts (Size in Brackets)

acting (8)	**storyline** (15)	**production** (14)
actor	story	set
actress	beginning	scenery
acting	ending	costume
role	script	producer
cast	plot	crew
...	...	...

soundtrack (4)	**cinematography**(20)
music	camera angle
score	shot
song	slow-motion
soundtrack	director
	editing
	...

Table 3: Top 10 Aspect Lemmas Clustered by ESA

acting	**storyline**	**soundtrack**
acting	storyline	soundtrack
actor	storylines	song
role	character	release
actress	comic	music
filmography	reveal	album
co-star	series	track
act	story	feature
career	appear	band
television	universe	discography
theatre	villain	label

cinematography	**film-production**
cinematography	preproduction
runtime	asheville
distributor	contractees
budget	all-animated
min	high-living
film	cash-cow
edit	hit-and-miss
screenplay	star-driven
director	singer-actor
star	small-budget

5.2 Opinion Extraction

The extraction of opinions regarding individual movie aspects can be seen as an instance of opinion mining at the phrase level. On the basis of the movie aspect clusters described in Section 5.1, the remaining steps to be performed for each review are: 1) Identifying opinion-bearing words and potential movie aspects, 2) linking opinion-bearing words to potential movie aspects, 3) identifying the semantic orientation of the opinions, and 4) aggregating all opinions for each movie aspect cluster. We perform sentence splitting, tokenization, part-of-speech tagging and lemmatization, and then identify the movie aspects and the opinion bearing words in each review. For the latter task, we use the subjectivity clue lexicon from Wilson et al. [23].

In contrast to documents from other online sources of user-generated content, the reviews collected from the IMDB exhibit a rather high quality. Proper capitalization, correct grammar and a rather small number of spelling errors were evident for most of the documents inspected. We observed that the users write in a rather elaborate style, which often results in long sentences with nested clauses etc. This level of sentence complexity in the reviews rules out the use of shallow pattern-matching surface methods for linking an identified opinion word to its target. Such methods based on e.g. word distance [12] or part-of-speech patterns [24] have been used in the past. These methods have the advantage of being computationally very efficient. However, the use of syntactic parsers, while computationally more expensive, can yield more accurate structural analyses [26], which is of particular importance for more complex analyses such as negation detection. Our approach is therefore based on the syntactic analysis of the review sentences. We employ the Stanford Parser[7], which extracts typed dependencies from the grammatical relations in a sentence. In contrast to the work in [26], our approach does not require a training phase for learning relevant constituents and their syntactic relations . Instead, the extraction of movie aspects with their corresponding opinions is done on the basis of two *generic dependency relation patterns*:

The first pattern makes use of the fact that adjectives are the major means of expressing positive or negative opinions. Adjectives also make up a the largest single fraction (48%) of the subjectivity clues in the Wilson lexicon. Accordingly, we found the majority of dependency relations between opinion words and movie aspects in our corpus to be adjectival modifiers (AMOD) as in "*the beautiful soundtrack*" and nominal subjects (NSUBJ) as in "*the soundtrack is beautiful*". Such direct dependency relations are therefore used by us to extract a movie aspect with the corresponding opinion.

However, there are quite a few sentence constructions in which the relation between the opinion word and the movie aspect is established over an intermediate word. Consider the sentence: "*This is acting at its most laconic form.*" in which the word "*form*" establishes the link between the movie aspect "*acting*" (PREP) and the opinion word "*laconic*' (AMOD). Our second pattern captures these connections involving intermediate words. It also enables us to extract opinions on both aspects from sentences as "*The entire score and the atmosphere are awesome.*" in which the parser will identify the relation between "*score*" and "*awesome*" (NSUBJ) as well as the conjunction between "*score*" and "*atmosphere*"

[6] http://mallet.cs.umass.edu/

[7] http://nlp.stanford.edu/software/lex-parser.shtml

Table 4: Top 10 Movie Aspect Lemmas Clustered by LDA

woman	man	film	play	house	play	story	thriller	action	scene
life	police	make	character	gore	song	make	scene	scene	character
man	cop	time	role	remake	musical	king	plot	bond	sequel
wife	guy	watch	performance	night	music	tale	work	plot	make
father	drug	fact	actor	zombie	performance	vampire	end	sequence	series
daughter	town	story	story	dead	role	set	hitchcock	guy	humor
find	city	character	time	budget	cast	film	suspense	james	part
son	western	feel	give	director	woman	dr	room	die	tv
husband	john	lot	tom	make	stage	version	find	car	michael
young	gang	show	make	genre	screen	dracula	director	chase	plot
life	girl	good	character	cast	movie	thing	effect	character	american
world	young	film	story	john	watch	end	human	work	man
human	child	oscar	book	dvd	make	people	alien	style	make
experience	kid	star	make	make	time	time	earth	director	world
present	family	year	man	play	lot	happen	space	audience	show
people	year	win	performance	work	people	start	world	visual	country
reality	boy	picture	comic	direct	story	feel	fi	story	people
story	school	number	give	director	thing	scene	sci	camera	war
mind	high	director	actor	role	scene	make	back	filmmaker	america
society	mother	actor	time	ben	expect	point	crew	art	soldier

(CONJ). We can thus extract the opinion regarding each of the two movie aspects. This simultaneous extraction also works for two opinions being expressed for one aspect, such as in "*The characters are unbelievable and flat*".

The task of detecting negation during the opinion extraction is also done by analyzing the dependency parser output. If we find a direct negation relation to an opinion word, we invert the polarity, i.e. the positive or negative orientation, of that opinion. In the case of a relation with an intermediate word, we check for a negation relation to the opinion word or the intermediate word. Figure 3 illustrates the possible dependency relation paths which our approach uses to extract pairs of opinion words and movie aspects, and to do the negation detection.

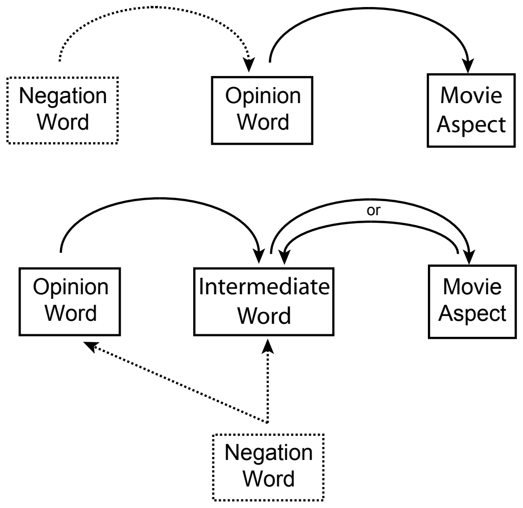

Figure 3: Possible Dependency Relations

6. EXPERIMENTS

Our experiments examine the usefulness of extracting opinions about movie aspects and using them as additional features for recommendation based on collaborative filtering. Our recommendation system will predict the overall star rating of movies for given users. We perform a ten-fold cross validation on the dataset of ratings and reviews as described in Section 3. For each testing fold, we calculate the root mean square error (RMSE) between the actual star ratings and our predictions, which is in turn averaged over the ten folds. We extrinsically evaluate the three different clustering approaches described in Section 5.1.

6.1 Experimental Setup

Our results are created by always using the overall star ratings of the users regarding the movies as the most basic feature. Since we also want to evaluate the usefulness of the extracted opinions against other features typically used for collaborative filtering in the movie domain, we created an extended baseline which also uses the genre information of the rated movie. The genre information was extracted from the IMDB as well. Note that the IMDB allows a movie to belong to more than one genre.

In our first non-baseline experiments (`⋆-Rating + Opin. Ratings`), we extract and accumulate all expressions for each movie aspect cluster from each review and average the identified opinion polarities (positive or negative orientation) in order to extract one overall polarity value for each cluster. We noticed, however, that this approach loses some relevant information, i.e. the *amount* of opinions uttered about that cluster. As described in Section 1, this information could be useful for the collaborative filtering, since it can reveal how important a certain movie aspect cluster is for a user. In our second set of non-baseline experiments (`⋆-Rating + Opin. Ratings + Num. Opinions`), we therefore also incorporated this number. Inspired by the reviewers' comments we plan to run experiments with two additional baseline configurations. See our website[8] for details.

6.2 Results & Discussion

The results of our experiments are shown in Table 5. For each setup the contributing features and the RMSE along with the corresponding 95% confidence intervals are given.

[8] http://www.ukp.tu-darmstadt.de/people/njakob/TSA09

Table 5: Results of Setups (smaller RMSE is better)

Setup	Features	RMSE (95%CI)
Baseline	⋆-Rating	$1.8526^{+0.0060}_{-0.0060}$
	+ Genre	$1.8319^{+0.0058}_{-0.0058}$
Manual	⋆-Rating + Opin. Ratings	$1.8225^{+0.0064}_{-0.0073}$
	+ Num. Opinions	$1.8221^{+0.0060}_{-0.0061}$
	+ Genre	$1.8090^{+0.0069}_{-0.0068}$
ESA	⋆-Rating + Opin. Ratings	$1.8269^{+0.0065}_{-0.0062}$
	+ Num. Opinions	$1.8243^{+0.0063}_{-0.0069}$
	+ Genre	$1.8080^{+0.0063}_{-0.0064}$
LDA	⋆-Rating + Opin. Ratings	$1.8230^{+0.0072}_{-0.0072}$
	+ Num. Opinions	$1.8139^{+0.0066}_{-0.0072}$
	+ Genre	$1.8073^{+0.0073}_{-0.0080}$

The first two rows contain the results of our baseline configuration in which we use the star ratings as a feature or both the star rating and the genre information. We observe that incorporating the genre information significantly reduces the RMSE. This was to be expected, as the genre information has been successfully employed in previous research. When analyzing the results of the three configurations which use the ratings extracted from the opinions (five clusters for "Manual" and "ESA", 20 for "LDA"), we observe that this additional feature reduces the RMSE in all approaches. When comparing the results of star rating plus genre information as features with the star rating plus opinion rating as features, we observe that the predictions when using the extracted opinion ratings are always better regardless of which clustering approach is used.

When comparing the setups based on opinion mining regarding pairs of identical features, we observe that the results of the ESA-based approach are always slightly worse than the approach based on the manual clustering. Apparently the slightly bigger clusters of the ESA approach and the fact that terms in the clusters definitely occur in the corpus as well compensate for the lack of detecting opinions about artists or directors. The ESA approach seems to be a reasonable option if the cluster topics can be defined manually, but the effort of filling the clusters by hand as well is not desired. The LDA-based approach performs slightly worse than the manual approach when using the `⋆-Rating + Opin. Ratings` features. However, when including the number of opinions and the genre information, it is consistently better than the other configurations and significantly better than the baseline. Ultimately, the information regarding the number of extracted opinions is beneficial regarding the predictions in all configurations. Apparently this feature introduces relevant information which allows the collaborative filtering to e.g. model how important a certain aspect cluster is to a user.

In our last experimental setup, we wanted to verify whether the features extracted by the opinion mining are complementary or redundant in combination with the genre information. As shown in the last row of each clustering-based approach, the results improve in all configurations when combining the opinion ratings with the genre information. We can therefore conclude that the opinions extracted about the movie aspects are a useful feature to improve the recommendations of the collaborative filtering. The confidence intervals indicate that all improvements with respect to the `⋆-Rating + Genre` baseline are statistically significant with at least $p < 0.05$.

Most important for the users' acceptance of the recommendation system is the proper prediction of the items the user is most intereseted in, maximizing true positives and avoiding false positives. Recommendation systems can be seen as supervised classifiers mapping the input features to two classes: *likes* and *dislikes*. In order to evaluate our models with respect to this consideration we re-interpreted all given ratings: ratings smaller than the global average (6.997) were labeled as *dislikes* and ratings above labeled accordingly as *likes*. Now we can compare the models in terms of receiver operating characteristics (ROC) as well as the area under the ROC curve (AUC) [7]. We calculated the AUC values for the LDA approach with all features as it yielded the best results regarding RMSE. The `⋆-Rating + Genre` baseline is improved by approximately 1.18%: $AUC_{LDA} = 0.9072 > 0.8967 = AUC_{baseline}$ (higher AUC is better).

7. CONCLUSIONS AND FUTURE WORK

In this paper we have shown how the extraction of opinions from free-text movie reviews can be used as features for a recommendation system to improve the prediction accuracy. The information extracted from the users' opinions can be employed in combination with structured information about the movies which in turn leads to better results. Our results show that the LDA-based movie aspect extraction and clustering approach yields the best results while the candidate extraction and clustering work fully automatic. We see the main difference between the LDA-based and the other two approaches in the number and the granularity of the clusters extracted. We can conclude that the larger number and fine-grained clusters provide a broader / better representation of the topics in the corpus and are therefore beneficial for the recommendation accuracy. However in future work we might investigate whether a disambiguation of movie aspects that occur in more than one LDA cluster can lead to even better results, since the opinion extraction would be more exact then. The results we obtained with the ESA-based approach are promising, but we observed that the ability to only calculate the semantic similarity between single word terms limits the detection of e.g. actors as semantically related to the "*acting*" category. If we can overcome this limitation, we could improve the detection of opinions regarding some categories, which could in turn lead to better recommendations.

The elaborate style of the majority of the reviews, in combination with complex sentence structures lead to a frequent use of anaphora in the documents. By resolving the anaphora we might increase the recall of the extracted opinions.

The representation of the opinions for the collaborative filtering might be improved by analyzing the positive and the negative opinions separately: In our current setup the recommendation system only receives the overall number of opinions regarding a certain aspect cluster. The differentiation between positive and negative opinions could lead to a more exact representation of the review.

8. ACKNOWLEDGMENTS

The project was funded by means of the German Federal Ministry of Economy and Technology under the promotional reference "01MQ07012". The authors take the responsibility for the contents. The information in this document is proprietary to the following THESEUS consortium members funded by means of the German Federal Ministry of Economy and Technology: Technische Universität Darmstadt, Siemens AG. The information in this document is provided "as is", and no guarantee or warranty is given that the information is fit for any particular purpose. The above referenced consortium members shall have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials subject to any liability which is mandatory due to applicable law. Copyright 2009 by Technische Universität Darmstadt, Siemens AG. All rights reserved.

9. REFERENCES

[1] S. Aciar, D. Zhang, S. Simoff, and J. Debenham. Informed recommender: Basing recommendations on consumer product reviews. *IEEE Intelligent Systems*, 22(3):39–47, May / June 2007.

[2] N. Bhatti and S. H. Weber. *Handbook of Research on Social Dimensions of Semantic Technologies and Web Services*, chapter Semantic Visualization to Support Knowledge Discovery in Multi-Relational Service Communities, pages 281–303. IGI Global, 2009.

[3] S. Blair-Goldensohn, K. Hannan, R. McDonald, T. Neylon, G. Reis, and J. Reynar. Building a sentiment summarizer for local service reviews. In *Proceedings of the WWW2008 Workshop: NLP in the Information Explosion Era*, Beijing, China, April 2008.

[4] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, March 2003.

[5] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pages 43–52, Madison, WI, USA, 1998.

[6] X. Cheng and F. Xu. Fine-grained opinion topic and polarity identification. In *Proceedings of LREC*, pages 2710–2714, Marrekech, Morocco, May 2008.

[7] T. Fawcett. Roc graphs: Notes and practical considerations for researchers. Technical report, HP Laboratories, 2004.

[8] L. Ferreira, N. Jakob, and I. Gurevych. A comparative study of feature extraction algorithms in customer reviews. In *Proceedings of ICSC*, pages 144–151, Santa Clara, CA, USA, August 2008.

[9] E. Gabrilovich and S. Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of IJCAI*, pages 1606–1611, Hyderabad, India, January 2007.

[10] M. Gamon, A. Aue, S. Corston-Oliver, and E. Ringger. Pulse: Mining customer opinions from free text. In *Advances in Intelligent Data Analysis VI*, volume 3646 of *Lecture Notes in Computer Science*, pages 121–132. Springer Berlin, August 2005.

[11] J. L. Herlocker, J. A. Konstan, L. G. Terveen, John, and T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 22:5–53, 2004.

[12] M. Hu and B. Liu. Mining and summarizing customer reviews. In *Proceedings of KDD-04*, pages 168–177, Seattle, WA, USA, August 2004.

[13] S.-M. Kim and E. Hovy. Extracting opinions, opinion holders, and topics expressed in online news media text. In *Proceedings of the ACL Workshop on Sentiment and Subjectivity in Text*, pages 1–8, Sydney, Australia, July 2006.

[14] C. Lippert, S. H. Weber, Y. Huang, V. Tresp, M. Schubert, and H.-P. Kriegel. Relation-prediction in multi-relational domains using matrix-factorization. In *Proceedings of the NIPS 2008 Workshop: Structured Input - Structured Output*, Vancouver, Canada, December 2008.

[15] B. Long, X. Wu, Z. M. Zhang, and P. S. Yu. Unsupervised learning on k-partite graphs. In *Proceedings of the KDD 2006*, pages 317–326, Philadelphia, PA, USA, 2006.

[16] B. Long, Z. M. Zhang, X. Wu, and P. S. Yu. Spectral clustering for multi-type relational data. In *Proceedings of the ICML 2006*, pages 585–592, Pittsburgh, PA, USA, 2006.

[17] Y. Lu and C. Zhai. Opinion integration through semi-supervised topic modeling. In *Proceedings of WWW '08*, pages 121–130, Beijing, China, April 2008.

[18] A.-M. Popescu and O. Etzioni. Extracting product features and opinions from reviews. In *Proceedings of HLT and EMNLP*, pages 339–346, Vancouver, Canada, October 2005.

[19] J. B. Schafer, D. Frankowski, J. L. Herlocker, and S. Sen. Collaborative filtering recommender systems. In *The Adaptive Web*, volume 4321 of *Lecture Notes in Computer Science*, pages 291–324. Springer, 2007.

[20] N. Srebro, J. D. M. Rennie, and T. S. Jaakkola. Maximum-margin matrix factorization. In *Advances in Neural Information Processing Systems 17*, pages 1329–1336, 2005.

[21] G. Takacs, I. Pilaszy, B. Nemeth, and D. Tikk. On the gravity recommendation system. In *Proceedings of KDD Cup Workshop at SIGKDD'07, 13th ACM Int. Conf. on Knowledge Discovery and Data Mining*, pages 22–30, San Jose, CA, USA, 2007.

[22] I. Titov and R. McDonald. Modeling online reviews with multi-grain topic models. In *Proceedings of WWW '08*, pages 111–120, Beijing, China, April 2008.

[23] T. Wilson, J. Wiebe, and P. Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of HLT and EMNLP*, pages 347–354, Vancouver, Canada, October 2005.

[24] J. Yi, T. Nasukawa, R. Bunescu, and W. Niblack. Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques. In *Proceedings of ICDM*, pages 427–434, Melbourne, FL, USA, December 2003.

[25] K. Yu, S. Yu, and V. Tresp. Multi-label informed latent semantic indexing. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 258–265, Salvador, Brazil, 2005. ACM Press.

[26] L. Zhuang, F. Jing, and X.-Y. Zhu. Movie review mining and summarization. In *Proceedings of CIKM*, pages 43–50, Arlington, VA, USA, November 2006.