Will I like it? – Providing Product Overviews based on Opinion Excerpts

Silviu Homoceanu, Michael Loster, Christoph Lofi, Wolf-Tilo Balke
Institut für Informationssysteme
Technische Universität Braunschweig
Braunschweig, Germany
silviu@ifis.cs.tu-bs.de, m-r.loster@tu-braunschweig.de, lofi@ifis.cs.tu-bs.de, balke@ifis.cs.tu-bs.de

Abstract — **With the growing popularity and acceptance of e-commerce platforms, users face an ever increasing burden in actually choosing the right product from the plethora of online offers. Thus, techniques for personalization and shopping assistance are in high demand by users, as well as by shopping platforms themselves. For a pleasant and successful shopping experience, users should be empowered to easily decide on which products to buy with high confidence. However, especially for entertainment goods like e.g. movies, books, or music, this task is very challenging. Unfortunately, todays approaches for dealing with this challenge (like e.g. recommender systems) suffer severe drawbacks: recommender systems are completely opaque, i.e. the recommendation is hard to justify semantically. User reviews could help users to form an opinion of recommended items, but with several thousand reviews available for e.g. a given popular movie, it is very challenging for users to find representative reviews. In this paper, we propose a novel technique for automatically analyzing user reviews using advanced opinion mining techniques. The results of this analysis are then used to group reviews by their semantics, i.e. by their contained opinions and point-of-views. Furthermore, the relevant paragraphs with respect to each opinion is extracted and presented to the user. These extracts can easily be digested by users to allow them a quick and diverse forming of opinion, and thus increasing their confidence in their decision, and their overall customer satisfaction.**

Human Computer Interaction; Knowledge Management; E-Commerce; Opinion-Mining.

I. INTRODUCTION

In recent years, the amount of products available on online e-commerce platforms has increased tremendously. In particular, this trend can be observed in the area of entertainment goods (like e.g. movies, books, music, computer games, etc.). Modern online platforms enable customers to buy or rent a vast selection of mainstream as well as long-tail titles, far beyond the capabilities of brick and mortar stores. Due to this vast selection of titles and the convenient shopping experience compared to physical shops (e.g. product browsing can be performed at home, goods are directly delivered digitally or via mail), these platforms enjoy an ever increasing popularity. However, this freedom comes at a price: users are often overwhelmed by sheer amount of offers and have a hard time deciding on a certain product.

At the same time, users of web-based platforms loose the benefits of professional sales staff. While in a decent video rental store, the staff might have provided the user with helpful insights (e.g. "You will probably like this movie; it has top-notch special effects, and the plot is very intriguing."), most online platforms only offer remarkably little help to their users. This problem is for many product categories further aggravated by the absence of useful metadata. For example, many movie-on-demand websites offer just simple metadata like title, year of release, an actor list, or a brief plot summary. This information usually does not allow for inferring the movies quality.

This situation poses a severe challenge for user and e-commerce platforms alike: users have a hard time finding and deciding on suitable titles, and often lack confidence in their shopping decision due to their offered items. This customer insecurity can have a strong negative impact on e-commerce platform's revenue: users who can't decide which items to buy will buy less. This effect is commonly known as choice-overload, and is even worse for subscription-based business models like e.g. Netflix[1], where users pay a fixed monthly fee which allows them to rent a given number of movies: as soon as a user rented all titles he knew upfront and felt confident with, the likelihood of canceling the subscription grows tremendously [1]. Thus, one of the platforms most important tasks is to keep the user interested in new titles, and offer him the tools for familiarizing himself with the platform's offers in an authentic fashion. But also more traditional platforms, like e.g. Amazon[2] can suffer from decreased sales volumes if users feel insecure [1].

Of course, these problems have been addressed to a certain degree by the research community as well as content platforms themselves. For example, bigger platforms like e.g. Amazon or Netflix employ recommender systems to suggest titles to their users based on their preferences [2,3]. Most recommender systems are based on collaborative filtering, i.e. users provide ratings for the titles they buy or rent, and the feedback of similar users with similar interests is used to recommend additional titles (e.g. "users who liked movie X also like movie Y"). While many recommender systems show convincing results [4] with respect to recommendation accuracy, users often have problems to understand before buying or renting why a recommended item is supposed to be a good choice, and often are unsure whether they will like the item or not. This issue is mainly due to the fact that there is no information available why a user should

[1] http://www.netflix.com/
[2] http://www.amazon.com/

like an item; users have to blindly trust the algorithms underlying the recommender system.

User-provided *reviews* have been shown to be a potent tool for helping users to decide on a certain product, and have become increasingly popular with the growing integration of Web 2.0 techniques in e-commerce platforms. They provide subjective summaries of the reviewed item, and discuss positive and negative aspects of its features. Reading reviews from different point-of-views, showing different opinions thus can be a valuable tool for boosting the confidence with respect to buying or not buying an item. This effect has also been recognized by many state-of-the-art platforms: for example, Amazon encourages users to provide reviews of recently bought products, which are then directly incorporated into the product's web shop page. Also, companies like e.g. the Internet Movie Database IMDB[3] base their whole business model solely on collecting and aggregating meta-data on movies, with user ratings and especially user reviews being a core part of the data sets. Unfortunately, the increasing popularity of user-generated reviews also hampers their originally intended benefit: more popular titles like e.g. the 2009 movie release "Avatar" has over 2,700 user reviews on IMDB. While this number of reviews contains an impressive wealth of subjective impressions and discussions on the movies merits and shortcomings, users trying to decide if they should rent this particular movie will hardly be able to read and distill even just a fraction of these reviews. Furthermore, with the growing importance of mobile e-commerce, traditional review-based platforms are further challenged due to the small available screen estate of mobile devices.

Therefore, additional *semantic* processing of the plethora of user reviews is required to generate concise and meaningful summaries of the contained sentiments and information. Using this information, all reviews can be *semantically clustered* for feature-based browsing. Additionally, this semantic approach also allows for selecting just short representative *review excerpts* for display to the user, each showcasing a different and informative set of opinions. This ensures that the user needs to only read a very small amount of text in order to obtain a complete and representative overview of the strengths and weaknesses of the chosen product. Furthermore, as these excerpts are very short, this approach is perfectly suited for mobile devices.

For example, for the title "Avatar", nearly all authors of the over 2,700 reviews positively mention the impressive special effects and stunning visuals, while at the same time, a significant number of users negatively criticize the predictive and uninspired plot as well as the shallow characters. In addition to offering short and concise text fragments contain the full spectrum of relevant opinions, short summaries per movie highlighting the strong and weak points of a movie.

However, extracting summaries and excerpts of strong and weak points from reviews is far from trivial as individual reviews usually focus only on a subset of features. In this paper, we propose a workflow for *automatically* extracting feature-based opinions and generating summaries of positive and negative aspects mentioned in user reviews. We will base our work on notable results from the area of *opinion mining*, which we will heavily modify to accommodate for the peculiar properties of the domain of movie reviews. The results of this analysis will be used to generate short *polarity profiles* (i.e. a list of features and the general user opinion for the features) for each movie. Additionally, we will automatically extract short and concise excerpts just containing the relevant passages of a review text. Finally, we will evaluate our approach based on review information available in the IMDB platform.

[3] http://www.imdb.com/

Fig. 1. Our prototype interface for opinion-based review clustering. Numbers in parenthesis indicate the number of reviews positively or negatively mentioning a feature. Clicking the link will list all respective reviews.

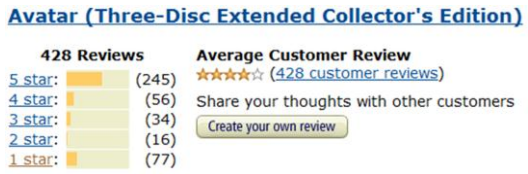

Fig. 2. Star rating-based clustering at *amazon.com*

II. Basic Review Clustering

In this section, we will showcase commonly used clustering approaches for reviews: a) star-rating-based clustering, and b) text-based clustering.

At amazon.com, reviews are clustered by their star rating (authors writing a review may rate each product on a scale from 1 to 5, with 5 being best). A summary of the review clustering is presented to the user at the shop page of each product (see Fig. 2). From here, users may manually explore the clusters (i.e. reviews are filtered by star rating). Furthermore, users may rate the usefulness of each review ("I think this review was useful"). Using this, Amazon can select the most useful positive (5 star) and most useful negative (1 star) review for direct comparison. Unfortunately, this approach is completely agnostic with respect to the content of reviews, different point of views cannot be distinguished or explored. For example, for the "Avatar Extended Special Edition", there are negative reviews dealing with the movie itself, as well as negative reviews discussing the poor packaging which leads to damaged discs. However, the system is not able to distinguish between those completely different reviews.

To remedy this problem, the most apparent solution is using a text-based clustering approach as commonly seen in information retrieval systems. While text-based clustering in order to identify meaningful groups of semantically similar reviews seems to be very promising, we argue that the use of sophisticated opinion mining techniques is essential for the clustering to be effective. We performed a pre-study, clustering 2,785 reviews of the movie "Avatar" using k-means clustering with k=5 (with five being a psychologically plausible number of texts for manual inspection). In order to facilitate this clustering, all occurring words in each review are stemmed and stop words are removed. Similarity between two reviews is computed by using inverse cosine distance.

It turns out that the results of this approach are hardly convincing: semantically, no clear cluster-wide opinions could be detected during manual inspection (e.g. the first cluster mainly contained reviews using many adjectives, while reviews in the third cluster frequently mention other movies). Furthermore, when statistically analyzing the clusters, more problems are apparent (see Table I): while clusters show different star ratings, the standard deviation of ratings is quite high (e.g. 3.02 for an average values of 6.59); thus although the reviews of one cluster are similar on word level, they contain wildly varying opinions. Furthermore, it turns out that it is difficult to cleanly separate clusters: the average distance of documents within one cluster to its centroid (aicd) is very large (around ~0.7 for all clusters on a scale of [0,1]). This means that the document space is very homogenous and no clear clusters are present on word level.

TABLE I RESULTS OF TEXT-BASED CLUSTERING OF REVIEWS

cluster	size	avg. star rating	std. dev. star rating	aicd[1)]
1	649	9.23	1.73	0.75
2	636	7.77	2.33	0.74
3	293	8.04	2.21	0.66
4	675	4.77	2.89	0.76
5	517	6.59	3.02	0.77

1) Average intra-cluster distance to centroid [0,1] – the smaller the better

This effect can mainly be contributed to the fact that text-based document clustering completely ignores actual semantics of texts, i.e. similarity is computed just on word level. If a system was able to cluster by a semantic similarity, the results would be of much higher quality. This is exactly where our approach steps in: the semantics of product reviews are the authors' opinions on the most defining product features. Thus, by mining these opinions such that further processing is possible, semantically meaningful navigation, clustering, and summarization of reviews is possible.

The next sections will outline how our system is designed to achieve this challenging goal.

III. OPINION MINING

With the growing amount of available user-generated content, the field of opinion mining gained more popularity in the recent years. Early works like e.g. [5] focused on determining the polarity (i.e. negative or positive) of whole texts. These techniques could be sucessfully applied to e.g. classifing newspaper articles. However, in the area of e-commerce, the overall polarity of a text does not yield sufficient information. This can be attributed to the fact that user-generated reviews in these fields are usually accompanied with a star-rating, which directly represents the user's overall polarity. Now, the pressing question is why the user's reached his/her overall opinion, and which aspects and features of the product have been perceived positivly, and which aspects negativly. This information has a plethora of profitable usages: manufacturer or producers can use it to identify and alliviate their products' weaknesses, users can exploit the information for gaining a quick overview before potential purchases, or to quickly navigate through a large set of reviews.

These observations fostered feature-based opinion mining aproaches [6-8], which detect for each review the actual features which are discussed. Then, the polarity of each individual feature is detected, and finally summarized. The generic workflow of such systems is presented in the next section.

IV. THE FEATURE-BASED OPINION MINING WORKFLOW

In the following, will build upon the work of Hu and Liu, who proposed a generic feature-based opinion mining framework in [6]. However, this process will be heavily modified in order to increase its domain-specific performance. In a nutshell, the workflow of such a framework comprises three major steps:

- *Feature extraction*: in this step, all relevant domain features are automatically extracted from the available reviews. For the movie domain, resulting features are e.g., "plot", "special effects", or "character depth". In this work, in addition to just using generic domain features, we will also identify features specific to each individual movie.
  As a result, a database of generic movie features is generated, as well as a database of specific per-movie features.
- *Opinion extraction*: after identifying the relevant features, for each review the polarity of all mentioned features is detected. The polarity encodes the user's opinion with respect to the related features (e.g. negative, positive, neutral, etc.) .
  Using this information, individual sentences can be tagged with their polarity and covered features. Furthermore, the whole review can be tagged with a *review polarity profile*, i.e. the author's opinion of all features he/she mentioned. These polarity profiles can also be used for browsing and exploring the set of all reviews (e.g. "select all reviews positively mentioning the plot").
- *Summary generation*: in the final step, a summary of the essence of all reviews has to be created, i.e. the polarity profiles of all reviews have to be aggregated into a global *movie polarity profile*. During this final aggregation, additional aspects can be respected for weighting the influence of individual reviews like for example the trust into the review (e.g. expert vs.

user review, "how many people think this review was useful", etc.).

A. Feature Extraction

In most feature-based opinion mining systems, three major tasks have to be performed in order to extract meaningful features: a) part-of-speech (POS) tagging b) frequent feature detection c) feature pruning. During POS tagging, words in individual sentences are tagged by their respective part-of-speech type (i.e. nouns, verbs, adjectives, etc.). In the recent years, the natural language processing community spent a lot of effort into developing successful POS algorithms, thus several viable approaches for POS tagging are readily available [9-11] or [12,13].

The next task is to identify frequent features (i.e. which words describe features relevant in the text). In [6], the authors propose a sentence-based analysis, just retaining the nouns detected by the POS tagger. Then, association rule mining [14] is used to learn the most frequent features. The basic rationale of this procedure is, according to [6], that most users will mention at least the majority of the most important features in their review (co-occurrence based approach). Hence, association rule mining will return the relevant features if sentences (or whole reviews) are considered as being the transactions, and nouns as being items.

However, association rule mining comes with some significant drawbacks and challenges: on the one hand, the complexity of association rule mining algorithms is prohibitive for large-scale problems [15]; this holds true for the popular a-priori algorithm [16] used in [6], as well as more efficient algorithms like FPGrowth [17]. On the other hand, association rule mining generates (similar to other co-occurrence based feature extraction approaches like, e.g. [18]) many feature candidates which are no real features per-se, but are just common frequently co-occurring descriptive phrases like e.g. "no problem", or "slow" [19].

Hence, we propose a *multi-stage language model approach*: the basic assumption of this approach is that each product domain (e.g. the domain of movies, containing features like "plot", or "acting") has its specific language, i.e. certain noun phrases are mentioned more often in product reviews than in common English language; and these words will most likely be relevant product features. Using this assumption, a language model for the current domain can be constructed [20] which can then be used to extract prominent features or terms (like e.g. in [19]). However, in addition to the domain-specific approach, we claim that the usage of an additional "layer" of language models is beneficial. Instead of just generating a single domain-specific model, for each product an additional language model is derived. The underlying rationale is that each item has distinctive features which are especially important, but are not important for other items of the same domain (e.g. the science-fiction movie "Avatar", with important features like "special effects" or "CGI", should be treated differently than for example "Pretty Woman", with features like "romance" or "character").

We start the process of extracting product features by building a list of candidates. Prior research indicates that technical terms and product features are generally nouns [21]. In consequence, the list of candidates actually comprises all nouns, simple and composed, extracted by means of POS and sentence chunking from all available product reviews (movie reviews in our case). Since one of the known shortcomings of user reviews is the strong mix between technical and general language, many of these nouns have no significant meaning with respect to the product field (for example neither 'comment' nor 'sea' from the following movie review snippet 'I am sure my comment will be lost in a sea of blue but anyways' are significant for movie language). Eliminating these nouns can successfully be performed with just simple heuristics. The basic assumption is that nouns belonging to general speech have the same or similar probability of appearing in text belonging to different fields. On the other hand, nouns which are more specific to a certain field have a higher probability of appearing in documents belonging to that product field than in documents belonging to other fields. Based on this observation we can define the characteristic power of a noun for a product field, as the probability that a document containing that noun belongs to the field in question compared to the probability that the document belongs to other fields: consider for example the noun 'plot'. Its characteristic power for the field of movies can be calculated as the probability that a document *d* containing the noun 'plot' is a movie review (the conditional probability $Pr(d \in M | 'plot' \in d)$, where *M* is a collection of movie reviews) minus the probability that *d* belongs to some other field ($Pr(d \in G \mid 'plot' \in d)$, with *G* representing a corpus of documents from various fields).

In other words, the problem of establishing whether a noun (*cf*) from the candidate feature list is significant for the product field in question, is reduced to a comparison between the following conditional probabilities: the probability that a document in which *cf* appears, belongs to product domain specific language on one side, and the probability that a document in which *cf* appears, belongs to general language on the other side. Therefore, c*f* can be pruned from the candidate feature list, if:

$$Pr(d \in M \mid cf \in d) - Pr(d \in G \mid cf \in d) \leq \theta \qquad (1)$$

with θ being a positive collection dependent parameter which can be tuned during setup of the system.

Although none of the two conditional probabilities from (1) can directly be calculated, they can be rewritten using the Bayes' Theorem as follows:

$$Pr(d \in M | cf \in d) = \frac{\Pr(d \in M) \cdot \Pr(cf \in d \mid d \in M)}{\Pr(cf \in d)} \qquad (2)$$

$$Pr(d \in G | cf \in d) = \frac{\Pr(d \in G) \cdot \Pr(cf \in d \mid d \in G)}{\Pr(cf \in d)} \qquad (3)$$

The advantage performing this transformation is that, as we will further discuss, each of the resulting probabilities can be calculated easily: $\Pr(d \in M)$ represents the number of product field specific documents normalized by the total amount of documents:

$$\Pr(d \in M) = \frac{|M|}{|M|+|G|} \qquad (4)$$

The same holds for $\Pr(d \in G)$.

$\Pr(cf \in d | d \in M)$ obviously represents the number of product field specific documents containing *cf*, normalized by the total number of product field specific documents:

$$\Pr(cf \in d | d \in M) = \frac{|\{d \in M | cf \in d\}|}{|M|} \quad (5)$$

This holds analogously for $\Pr(cf \in d | d \in G)$.

Finally, since we are ultimately only interested in comparing the values computed by formula (2) and (3), there is no need to calculate $\Pr(cf \in d)$, and it thus can be dropped from the denominator during comparison.

With the conditional probabilities calculated, the candidate list is finally reduced to the product domain specific nouns. Although this may seem sufficient, there is still a large number of nouns in the list like, for example 'movie', 'film', 'character', 'actor'. While these words do belong to the movie domain, they are less meaningful in describing the particularities of a certain movie. For eliminating these domain specific, but sill general features, we employ the same process we have applied for eliminating general nouns as a second stage. Now, the difference is that *M* represents the document collection comprising reviews of the product in question only, while *G* represents the document collection containing the remainder of product reviews. Again this approach punishes nouns which have a similar significance in both describing the product as well as other products from the collection.

B. Opinion Extraction

During the opinion extraction phase, three major tasks have to be performed: a) identification of adjectives potentially carrying a sentiment for detected features, as well as detection of their polarity, b) assignment of the identified adjectives to a specific feature mentioned in the sentence, c) detection of the final polarity of each feature / adjective pair.

For easier handling, the POS-tagged sentence is converted to an object-oriented representation, i.e. each sentence object contains the actual word and the according POS type. Furthermore, during this conversion, the sentence is scanned for the presence of features using the feature database constructed in the previous step. This also allows for an easy identification of multi-word features like e.g. "special effects". Features are finally represented as self-contained objects, identified by a special type attribute (e.g. the verb "special" and the noun "effects" will be stored as the feature "special effects").

Now, each sentence is scanned for adjectives; and for each adjective the semantic polarity is detected. While [6] depended on expanding seed lists for polarity detection, we will use SentiWordNet [22,23]. SentiWordNet is an online service maintaining a large collection of common adjectives, and storing for each adjective the degree (from 0 to 1) of being neutral, positive, or negative. The usage of SentiWordNet is especially convenient and effective for commonly used adjectives like e.g. "good" or "bad". However, depending on the domain, users tend to be very creative with the usage of non-common adjectives (e.g. "trail-blazing", "quasi-comedic", "scycotic"). Therefore, reliable semantic polarity detection cannot depend on SentiWordNet alone. To properly address these special adjectives, the detection procedure is adapted so that the evaluation of adjectives with unknown semantic orientation is delegated to the users or administrators of the opinion mining system. This feedback then is used to build a database of the polarity of domain-specific adjectives, thus reducing the need for human interaction over time.

Next, each adjective is assigned to the closest feature in the sentence. This assignment process is based on the assumption that adjectives which refer to a feature are often found in the direct neighborhood of the corresponding feature. To do so, the assignment of the adjectives is achieved by calculating the distance of each adjective to all detected features in a sentence, and eventually assigning the adjective to the feature with lowest distance. Should there be two features with the same distance; the adjective is assigned to the feature mentioned first.

Finally, the polarity of each adjective / feature pair is established. For this purpose, the direct neighborhood of the examined adjective is scanned for negation words (like e.g. "not". Negation words can usually be detected by POS taggers). If such a negation word is found, the polarity of the adjective / feature pair is reversed.

Finally, after all sentences of a review are processed, the review's overall polarity profile is generated. For the sake of simplicity, each review can have one of four polarity values per feature: 'positive', 'negative', 'neutral', and 'not mentioned'. The final feature polarity is decided by a majority vote between all sentences of a review, i.e. if most sentences are positive with respect to a feature, the overall polarity is also positive (if a feature is not mentioned by some sentence, the overall polarity is 'not mentioned').

C. Summary Generation

In this phase, summarizing polarity profiles per movie are created. Here, multiple approaches are possible:

a) *Simple* aggregation of review polarity profiles: for each feature mentioned in the polarity profiles of any review of a given movie; positive, negative, and neutral mentions of that feature are added up independently. This results in three counts per feature (positive strength, i.e. number of reviews positively mentioning the feature; negative strength; and neutral strength).

b) *Weighted* aggregation of review polarity profiles: this approach is similar to the simple aggregation, but reviews are not all treated with the same weight. Additional heursitics can be applied to scale the opinion scores, like e.g. relative usefulnees of a review. For example, in IMDB, users can rate how useful a given review is. Consequently, the opinion scores of reviews only infrequently rated as being useful should be scaled down, e.g. they do not count as a full positive mention.

After positive, negative, and neutral mentions of all reviews of a given movie are aggregated, the polarity profile

needs to be pruned. In order to generate a summary like in Fig. 1, only the top-5 positive or negative features beyond a certain threshold are included. By adjusting the threshold or number of selected features, the detail of the summary can be controlled. Please also keep in mind that the same feature can appear as being positive as well as being negative, if there are widely diverging opinions on the features polarity in the reviews.

D. Review Clustering and Opinion Excerpts

Finally, we use the per-review polarity profiles to cluster all reviews in order to select the most representative reviews of a movie. If users trying to familiarizing themselves with a certain movie, simply reading one review from each cluster should suffice to be able to easily grasp all common and relevant opinions with regard to that movie.

Opinion-based clustering of reviews is a trivial task as soon as the polarity profiles have been computed: simply, one cluster is formed for the top-k_1 positive, and the top-k_2 negative features. The respective reviews can easily selected using their polarity profiles (and thus, a single review could be in multiple clusters). Furthermore, only relevant excerpts of a review with respect to a certain feature could be displayed to the user (called *opinion excerpts*. They are created by selecting just the paragraph or surrounding sentences of the occurrence of a feature). Opinion excerpts allows for a very quick and efficient skimming through the relevant opinion. Also, a direct comparison can easily be facilitated for features showing very diverse opinions: by displaying the opinion excerpts of the most useful (as given by user feedback) positive and most useful negative review with respect to a feature, a side-by-side view as shown in Fig. 3 can be created.

V. Evaluation

A. Feature Opinion Extraction

For this evaluation, opinion excerpts for the most relevant features of several movies are extracted, and manually checked for semantic correctness. The aim of this evaluation is to measure if user opinions are correctly identified and classified.

The procedure is as follows closely the approach already introduced in section IV.D: for three movies ("Avatar", "Gladiator", and "Amélie"), the top-5 positive and top-5 negative features are extracted. For each of these features, all respective reviews are selected (i.e. those reviews discussing the feature with the correct polarity). Then, for each review, the text excerpts relevant for the opinion are extracted (either the paragraph mentioning the opinion, or the three surrounding sentences; whichever produces less text). The top-5 (ordered by user-provided usefulness of the review the excerpt was contained in) opinion excerpts for each feature polarity are manually inspected. It is checked if the text does indeed express the opinion which was detected by the system. It turned out that 72% of all opinion excerpts have been classified correctly, while for 28% of all excerpts, the wrong opinion polarity was detected.

Generally, most misclassified opinions can be contributed to one of the following problem areas:

- *Referential problems*: sometimes, adjectives do not refer to the closest detected feature, but are semantically connected by referential connectors like e.g. "it", "this", etc. Furthermore, the feature could be actually referring to a completely different entity (e.g. when a review compares two movies). Example (from a review for "Avatar"): "In contrast to Avatar, Star Wars had an amazing plot."
- *Complex negation*: here, an author uses an opinion statement which is later verbally negated. While our algorithm includes negation detection, more complex verbal negations are often missed, and are generally hard to detect. Example: "Claiming that the plot is great would be outright lying." This problem contributes to 29% of the evaluated misclassifications.
- *Sarcasm or irony*: sarcastic or ironic comments are popular in user generated reviews. Thus, adjectives having the opposite polarity of the author's actual opinion are used, and detected by our system. 14% of all evaluated misclassifications can be attributed to sarcasm and ironic comments. Example: "So you could indeed say that the plot is absolutely amazing :-)" (sentence appearing after a long list of plot holes)
- *Insufficient accuracy of Part-of-Speech tagging*: during our work, we assumed that POS tagging generally shows high accuracy. But since POS taggers usually disregard the context or domain of a given text, many misclassifications on part-of-speech level can be observed, e.g. "alien" is identified as an adjective by our POS tagger (and associated with a negative polarity). However, most authors of sci-fi reviews use it as a noun. POS errors contribute to 7% of all of our misclassifications.

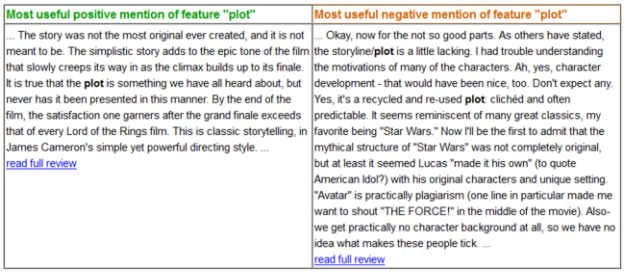

Fig. 3. Prototype interface for most useful positive and negative mention of a feature (here: "plot" for "Avatar"). The paragraphs surrounding the opinion are extracted.

Overall, the accuracy of our polarity detection algorithm is satisfying. However, significant improvements in polarity detection accuracy could be achieved with specialized techniques for reference handling (which also in includes techniques for specialized entity recognition, e.g. detecting movie names our director names), and detection of complex negations. Those two problem areas contribute to over 65% of the system's misclassifications.

TABLE III TOP-5 POSITIVE AND NEGATIVE FEATURES FOR "AVATAR"

Positive Mentions	
Feature	*Count*
Special effects	216
CGI	184
Screen	151
Plot	139
Visuals	93
Negative Mentions	
Feature	*Count*
Plot	182
Dialogue	67
Script	64
Storyline	64
CGI	24

TABLE II TOP-5 POSITIVE AND NEGATIVE FEATURES FOR "GLADIATOR"

Positive Mentions	
Feature	*Count*
Acting	293
Epic	184
Cast	114
Special effects	103
Plot	93
Negative Mentions	
Feature	*Count*
Plot	67
Acting	44
Characters	42
Dialogue	41
Scene	23

B. *Summarization*

In thus subsection, we will briefly showcase two generated summary profiles. These profiles are created by the simply aggregation method as described in IV.C for the movie Avatar (see TABLE II) and Gladiator (see TABLE III). The less obvious features selected by the algorithm can be semantically justified as follows: for the feature "screen" of the movie "Avatar", many users argue that the movie is very impressive at a big screen in a cinema (and not at home on DVD). This opinion is expressed very frequently. For "Gladiator", users often mention that the movie has an epic quality, which is usually considered being a positive characteristic. The usefulness of the selected features is evaluated in the next section.

C. *User Satisfaction*

For measuring the user satisfaction with respect to our approach, we have conducted an online user survey. The aim of the study was to compare our approach to the state-of-the-art alternative of reading whole user reviews. We performed this study in the domain of movie reviews. Our aims are to examine the following: how helpful are the two methods in providing an overall impression of a movie, and how much effort is needed for reading the provided information. Furthermore, we try to determine the acceptance of users for both approaches.

Focusing on the movie "Avatar", we have prepared two documents of similar length: one document contains three full length user reviews with fundamentally different ratings (2, 6 and respectively 10 stars out of 10). The three reviews have been handpicked based on the usefulness rating of all IMDB users (i.e. the selected reviews are among the most useful reviews according to the vote of the IMDB community). We have limited our survey to three carefully chosen reviews (summing to about $1^{1/2}$ A4 pages) after experiencing negative feedback regarding the needed effort for reading a higher number of reviews in a pre-study. The other document comprises the "Avatar" profile generated as described in section IV.C., accompanied by the review excerpts for the top 5 positive and negative features as described in IV.D. As in the case of the document containing whole reviews, the excerpts have been chosen based on how useful the corresponding reviews have been perceived by the community. This document also sums up to $1^{1/2}$ A4 pages. Users were randomly split into two groups, and performed either the survey containing full-text reviews, or the survey containing our opinion excerpts.

Please note that selecting the most useful reviews required a considerable amount of prior user feedback to derive a good usefulness rating. During manual inspection, it turned out that most reviews rated as being not useful, did not discuss any movie feature explicitly, but instead mainly contained very general statements like e.g. "Best / worst movie of all times.". Thus, there is also an indication that by working with opinion excerpts, reviews with a low user usefulness rating are more likely to be avoided. This property could be exploited if there are no usefulness ratings readily available.

A total of 54 users have participated to the online survey. About 80% of them have also seen the movie before. Analyzing the rating distribution of these users (which have seen the movie), with an average rating of 7.5 stars and a standard deviation of 1.9 stars, their rating behavior is consistent with the ratings from IMDB, confirming the representativeness of the user sample. 31% of the users performing the first survey containing three complete reviews stated that the cognitive effort necessary to read and analyze the document was very laborious. Only 16% of the users participating in the second survey containing opinion excerpts shared that assessment despite the fact that this document contained fragments extracted from ten reviews and was of the same length. At the same time, only 3% of the users of the first survey didn't mind at all reading the complete document. For users who read the opinion excerpts, this percentage is four times higher (12%). Both results show clearly that our approach is considered more comfortable. Regarding the usefulness of the provided information (Fig. 4), both approaches result in a similar distribution of the perceived usefulness (users were

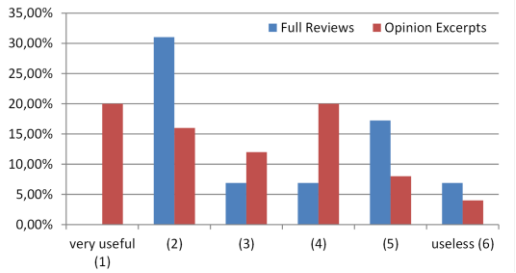

Fig. 4. Perceived usefulness of full reviews and opinion excerpts

confronted with questions asking if they now have a good overview of the strength and weaknesses of the movie after reading the document). Still, the opinion excerpts showed to be more helpful to users with an average rating of 2.9 (1 being most useful), while the approach using complete reviews showed an average of 3.5. Further inquiries on the users evaluating the opinion excerpts showed that over 70% of the users found organizing the information based on important features as being positive, and 64% of the users liked the fact that only excerpts of reviews have been presented.

All in all, we can conclude that our opinion excerpt based approach is suitable for providing users with good overview information on products. At the same time, the user experience is perceived as being significantly better than just reading reviews.

VI. Conclusions & Outlook

In this paper, we showcased how opinion mining techniques can be used to allow e-commerce platforms to provide effective decision support to their users. Our approach focuses on automatically analyzing the content of the plethora of user-generated product reviews which are available on most nowadays online platforms. In contrast to traditional purely rating-based systems, we analyze reviews semantically on a per-feature level. Especially, we are able to automatically derive the most discriminating features of a given product, as well as the respective user opinions (e.g. positive, negative, or neutral) from the review corpus. This information can be used to generate summaries and opinion overviews which go far beyond the capabilities of rating-based systems: on one hand, feature summaries allow for grasping a quick overview of the strong and weak aspects of a product. On the other hand, we are able to directly extract the representative text parts of reviews containing relevant opinions. This allows for selecting the most relevant reviews (i.e. containing the most important opinions), as well as generating textual overviews which further deepen the information gained by the previously mentioned feature summaries. In general, digesting the information contained in large numbers of product reviews has become much easier for users, and allows users for a quick forming of opinion with respect to the offered products. Thus, this approach perfectly augments black-box techniques like e.g. recommender systems or even simple best-seller lists which are usually unable to provide a good semantic justification why a user should be interested in an offered product. Finally, well informed users have more trust into their shopping decisions, and thus their overall shopping experience is better.

In future works, we will further aim at improving the accuracy of our feature extraction and polarity detection algorithms. The extensive evaluations performed in this paper provide a good baseline for developing even more efficient and elaborative techniques by showing the advantages and strengths of our current approach, as well as identifying potential areas of improvement.

[1] D. Bollen, B.P. Knijnenburg, M.C. Willemsen, and M. Graus, "Understanding choice overload in recommender systems," *ACM Conf. on Recommender Systems*, Barcelona, Spain: 2010.

[2] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: item-to-item collaborative filtering," *IEEE Internet Computing*, vol. 7, Jan. 2003, pp. 76-80.

[3] R.M. Bell, Y. Koren, and C. Volinsky, "All together now: A perspective on the NETFLIX PRIZE," *CHANCE*, vol. 23, Apr. 2010, pp. 24-24.

[4] Y. Koren and R. Bell, "Advances in Collaborative Filtering," *Recommender Systems Handbook*, Boston, MA: Springer US, 2011, pp. 145-186.

[5] S. Pang, B., Lee, L. Vaithyanathan, "Thumbs up? Sentiment Classication using Machine LearningTechniques," *EMNLP*, 2002, pp. 79-86.

[6] M. Hu and B. Liu, "Mining and summarizing customer reviews," *ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD)*, Seattle, USA: 2004.

[7] M. Hu and B. Liu, "Mining Opinion Features in Customer Reviews," *19th Conf. on Artifical Intelligence*, San Jose, USA: 2004.

[8] B. Pang and L. Lee, "Opinion Mining and Sentiment Analysis," *Foundations and Trends in Information Retrieval*, vol. 2, Jan. 2008, pp. 1-135.

[9] R. Mitkov, *The Oxford handbook of computational linguistics*, Oxford University Press, 2005.

[10] T. Brants, "TnT: a statistical part-of-speech tagger," *6th Conf. on Applied Natural Language Processing (ANLP)*, Washington, USA: 2000, pp. 224-231.

[11] A. Ratnaparkhi, "A Maximum Entropy Model for Part-Of-Speech Tagging," *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1996.

[12] "Infologistics' NLProcessor," *http://www.infogistics.com/textanalysis.html*.

[13] "Apache OpenNLP," *http://incubator.apache.org/opennlp/*.

[14] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules," *20th Int. Conf. on Very Large Data Bases (VLDB)*, Santiago de Chile, Chile: 1994.

[15] W. Kosters, W. Pijls, and V. Popova, "Complexity Analysis of Depth First and FP-Growth Implementations of APRIORI," *Machine Learning and Data Mining in Pattern Recognition*, vol. 2734, Jun. 2003, pp. 77-119-119.

[16] B. Liu, W. Hsu, and Y. Ma, "Integrating Classification and Association Rule Mining," *20th Int. Conf. on Knowledge Discovery and Data Mining (KDD)*, New York, USA: 1998.

[17] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation," *ACM SIGMOD Int. Conf. on Management of Data*, New York, USA: 2000.

[18] S. Morinaga, K. Yamanishi, K. Tateishi, and T. Fukushima, "Mining product reputations on the Web," *8th ACM Int. Conf. on Knowledge Discovery and Data Mining (KDD)*, Edmonton, Canada: 2002.

[19] C. Scaffidi, K. Bierhoff, E. Chang, M. Felker, H. Ng, and C. Jin, "Red Opal: product-feature scoring from reviews," *8th ACM Conf. on Electronic Commerce (EC)*, San Diego, California: 2007.

[20] K. Kageura and B. Umino, "Methods of automatic term recognition: A review," *Terminology*, vol. 3, Jan. 1996, pp. 259-289.

[21] H. Nakagawa and T. Mori, "A simple but powerful automatic term extraction method," *Int. Workshop on Computational Terminology*, Morristown, NJ, USA: 2002.

[22] A. Esuli and F. Sebastiani, "SENTIWORDNET: A Publicly Available Lexical Resource for Opinion Mining," *5th Conf on International Language Resources and Evaluation (LREC)*, Genoa, Italy: 2006.

[23] S. Baccianella, A. Esuli, and F. Sebastiani, "SENTIWORD NET 3.0 : An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining," *7th Conf on International Language Resources and Evaluation (LREC)*, Marrakech, Morocco: European Language Resources Association (ELRA), 2008.