MovieCommenter: Aspect-Based Collaborative Filtering by Utilizing User Comments

Minsam Ko[1], Hyung W. Kim[1], Mun Y. Yi[1], Junehwa Song[2], Ying Liu[1]

[1] Department of Knowledge Service Engineering
KAIST, Daejeon, 305-701, Republic of Korea
{ msko, hw_kim, munyi, yingliu }@kaist.ac.kr

[2] Department of Computer Science
KAIST, Daejeon, 305-701, Republic of Korea
junesong@nclab.kaist.ac.kr

***Abstract*—Collaborative filtering relies on numerical ratings for recommendations. While users consider various aspects of content as a basis of their evaluation, a numeric rating provides only an aggregated report of final assessment. The performance of a collaborative recommender system could be enhanced if the ratings are augmented by more specific information used for evaluation. In this paper, we present MovieCommenter, a recommender system that utilizes *movie aspects* – key features and users' opinions about the movie. We conducted a series of experiments to perform both qualitative and quantitative evauations of the system performance. The results show that our approach makes more precise recommendations than traditional approaches. Moreover, the interface of MovieCommenter was found to enhance the recommendation explanability, ability to explain how the recommendation was made. Because our approach is based on independent schema, this approach could be easily applied for recommending other domain contents.**

***Index Terms*—Collaborative filtering, Recommender system, Web services, Movie recommendation, Comment-based recommender**

I. Introduction

Entertainment contents are abundant and rapidly growing. Every day an enormous number of entertainment contents such as movies, books, and music, compete for potential customers' attention. Selecting satisfactory contents out of such huge collections, especially within a limited time, becomes harder and harder for customers as the range of choices continuously expands. People commonly seek external help by searching expert reviews or consumer opinions available from online portals or personal blogs. These resources provide useful information for movie selection but users still have to deal with digesting a large quantity of diverse information, with the risk of spoiling the actual enjoyment of the contents by finding unnecessary details in advance.

Collaborative recommender systems try to predict the utility of contents for a particular user based on the contents rated by other users, who are considered similar to the current user [1]. Most of these collaborative filtering systems utilize numerical ratings. The advantage of these systems using ratings is that data can be easily analyzed and summarized by mathematical methods.

However, numerical ratings do not retain the underlying information used for the evaluation. A rating is a numerical summary of user evaluation. While a user considers many elements and features of content as a basis of his or her evaluation, a numeric rating provides only an aggregated report of final assessment. The performance of a collaborative recommender system could be enhanced if the ratings are augmented by more specific information used for evaluation. In this study, we examine this possibility by extracting key features and user opinions from user comments and combining ratings with the extracted information.

More specifically, our approach utilizes user comments to extract *movie aspects* – defined as key features and user opinions about the movie. Because the length of a comment is usually limited to a small number of characters (80~150), most words in a comment tend to be strongly related to the key aspects of movie. Thus, by analyzing comments, important movie aspects can be readily captured. Further, in order to better utilize movie aspects, we also measure the strength of sentiment associated with the particular movie aspects. The sentiment assessment is made by utilizing numerical ratings. A rating, which is entered with a comment by the same user, reflects sentiment strength [7]. Without relying on complex natural language processing techniques, our approach captures the strength of sentiment. In addition, our approach can be easily adopted in various languages and domains, without being constrained by the unique characteristics and requirements of them.

MovieCommenter is a movie recommender system built to (1) provide accurate recommendations (quantitative results) by utilizing key movie aspects and (2) offer an interface that visualizes the key aspects (qualitative results) so that potential viewers could make more effective decisions. The experiments we conducted demonstrate that the system makes more precise recommendations than those made by traditional approaches. Moreover, the interface was found effective in enhancing the recommender system explanability, ability to explain how the recommendation was made.

The rest of this paper is organized as follows; Section 2 briefly reviews the state of the art in the recommender system field. Section 3 introduces the design requirements of our recommender system. Section 4 describes the implementation issues for our system. We then present the performance evaluation of the system including experimental results in Section 5. Finally, we finish our paper with conclusion in Section 6.

II. RELATED WORK

A considerable amount of effort has been made to improve recommendation techniques for diverse contents including news [4][10], books [12][17], and movies [5][8][14][15][20]. These studies utilize various resources in making recommendations. However, comments have received relatively less attention in the recommendation research area. Several studies are using comments for making recommendations yet most of them have simply treated comments as an extension or attribute of a targeted item without paying attention to properties and contents of comments [17].

A few studies tried to understand user comments and use their characteristics for recommendation. These approaches use words in comments that express how users felt about the movie to infer a rating about a targeted item [7] or ratings about predefined criteria [11]. Therefore, these approaches require a manual effort and time devoted to deduce and annotate sentiment. Our approach reduces this time consuming effort by utilizing users' ratings and comments together. Instead of analyzing strength of sentiment through complex natural language processing, we use the rating that has been inputted by the same user. Moreover, our approach does not require any predefined criteria because key movie aspects are dynamically created based the movie, enabling it to be easily adopted in various domains.

Broadly speaking, there are two types of recommendation approaches: collaborative filtering and content-based filtering [1]. Collaborative filtering recommends contents that are preferred by the users sharing similar tastes [1][10][12][15]. Most systems using this approach produce recommendations based on users' numerical ratings. They focus only on rating results and do not consider the reason why a user rated a particular movie this way. To improve the current practice, we utilize key terms in comments so that we can reflect the underlying reasons of user ratings. Comments contain useful contextual information in a user rating and this knowledge leads to more sophisticated personalization.

Content-based filtering makes recommendations based on the similarity between a target content and others that have been preferred by the user in past [1][5][17]. Content-based filtering research usually uses objective elements such as actor, director, genre and synopsis. Thus, it is difficult to predict how such a wide set of elements will exactly affect individual users. We use the aspects extracted from users' subjective comments, so that the features of movie can be better captured.

Adomavicius and Kwon proposed new recommendation techniques based on multi-criteria ratings [2]. Users' multi-criteria ratings give useful information to make recommendations. Yahoo Movie, for example, offers four criteria for rating: director, actor, music, and visual. By utilizing these criteria, main features of movie and users' preferences can be better understood. However, in this approach, the multi-criteria are predefined. A small number of criteria would be insufficient to indicate important features of the content, while many criteria would put heavy burden on the users. Our approach builds multi-criteria dynamically from user comments.

On the other hand, it has been recognized that explanations on why the recommendation is given improve the performance of a recommender system [9]. FilmTrust uses web-based social network to increase explanability [8]. Tagsplanation tries to explain recommendations based on tags [22]. In this paper, we increase the explanability of our system through the key aspects extracted from user comments. Even though more sophisticated text processing techniques are required, comments have more advantages in explaining recommendations because they contain users' detailed sentimental opinions more than tags or social network connections.

Outside of recommender systems, studies have explored the potential of comments in different ways. Several studies have used comments as an indicator of popularity of news [21] and blog posts [24]. Recent studies examined how useful comments could be identified [3][6]. Park et al. utilized comments to identify political orientation of news articles [19]. In this study, comments clearly represent users' political preferences. Such clarity of users' preference shown in their comments could improve building user profiles in recommender system. Lu et al. [13] suggest the method to extract main aspects from short comments in eBay through clustering. This work is similar to ours because we also extract important aspects from comments. However, our method considers users' preferences and generates a recommendation through personalization.

In the research of opinion mining, studies have been made on finding good or bad aspects of a product from users' reviews [18]. However, these studies do not focus on giving recommendations. In addition, they utilize the reviews without considering users' preferences. Even though some of these studies consider the writer's expertise about the product, the preference of the writer could be more important in the domain of cultural contents such as movie or book.

III. CHARACTERISTICS OF COMMENTS

It is necessary to understand the characteristics of comments to develop effective recommendations. User comments have become widely available on the Web as they can help other users' buying decisions. We quantitatively and qualitatively analyzed user comments to figure out how comments could be used for a more effective recommendation. Our analysis of comments was guided by the two possibilities explained below.

First, we analyzed comments to check whether the textual information in comments was informative and useful. Meaningful comments might be effectively utilized to improve users' understanding on items, compared with other explanatory texts such as title, genre, director, actor and synopsis.

Second, we examined how often each user left a comment on a specific movie. If a sizable number of comments were left on movies by a user, it could cover a large proportion of movies and be used to make better recommendations for the particular user.

A. Data set

We used data from Naver Movie (http://movie.naver.com). Naver Movie is one of the most popular movie portal sites in Korea. We collected data from the past 5 years (Jan. 2005 ~ May. 2010). The data includes the descriptions (title, genre, credit, and synopsis) and users' evaluations about movies. In Naver Movie, users can express their evaluations about a movie by entering a numerical rating (1 to 10) and a short comment (up to 40 Korean characters or 80 English characters) to explain the rationale of the rating. One user can give only one evaluation for each movie and their comments can be identified by Naver ID. Our collected data contains 2,269 movies and 2,189,989 evaluations by 883,583 users in total. Among the 2,269 movies, 1,728 movies are evaluated by at least one user.

B. Potential in Comments

Usefulness of Textual Information in Comment

The qualitative usefulness of textual information in comments is an important factor for recommendation. We checked whether comments contained more useful movie aspects (helping users' selection) than other text in movie descriptions such as title and synopsis by implemented a user study that involved five participants. The participants were graduate students in a research university in Korea. They were regular movie watchers, often choosing movies based on others' comments, but were not aware of the purpose of this study. First, we removed all the words except nouns, adjectives, and verbs in user comments and movie descriptions, then weighted the words by the term frequency (TF) in each set. Finally, we selected 150 most frequently appeared terms in comments and 150 most frequently appeared terms in movie descriptions. We gave the selected frequent terms to the participants in a random order and ask them to determine if each term is a useful movie aspect. In order to make the feedback as consistent as possible, we provide an evaluation guideline: a term is regarded as a useful movie aspect when it affects the user's decision on movie selection such as impressive features of movie or users' feelings.

TABLE I
PROPORTION OF USEFUL MOVIE ASPECTS IN EACH TEXT SET

	Participant				
	1	2	3	4	5
Term in Comment ①	120	53	128	105	71
Term in Description ②	32	13	108	39	8
① / (①+②)	0.79	0.80	0.54	0.73	0.90

Table I shows the study results. Most useful aspects came from the comments even though there were individual differences. On average, 75.25% useful terms chosen by a participant were located in comments, and all the participants chose more comment terms than description terms as useful movie aspects, without exception. The findings confirm that utilizing comments can be effective in capturing the essential aspects of movies.

Activeness of Commenting Behaviors

Nowadays, most movie websites provide user-friendly interfaces for users to leave comments on movies. Furthermore, the growth of social network services and mobile networks facilitates people's expression about their opinions and feelings with short texts. As a result, a very large number of comments are available online now, which could be utilized for broad advanced applications, such as recommender systems.

Our analysis found that both the users' participation in commenting movies and the amount of comments given by users were steadily increasing. A total number of 281,078 users left comments in 2009, while 89,433 users did so in 2005. In addition, the number of comments was also on increasing trend. The number of comments in 2009 has tripled from the number of comments in 2005.

In addition to the number of comments, users' commenting speed is also accelerating. It does not take long for a particular new movie to collect enough comments to launch recommender systems. Based on the comment amount, we define movies into several categories: ordinary movies that have 2000~3000 comments and popular movies that have more than 5000 comments. For each category, we estimated how long it will take to get 1,000 comments per movie. It takes about 10 days for an ordinary movie to get 1,000 comments (See Fig. 1). For popular movies, the time gets shorter: they only need three days to reach 1,000 comments. In most recommendation application areas, where speed is crucial because of the marking and business reasons, such a short accumulation period enables a prompt recommender system launch without any serious delay.

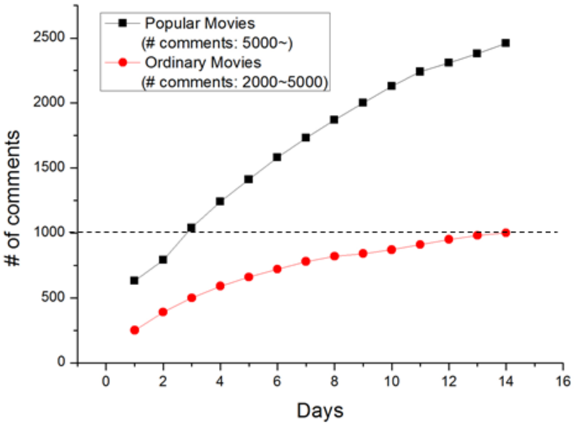

Fig. 1. Time for a Movie to Receive 1,000 Comments

Design Challenges

We have found that comments are growing every year and most of them are useful for content understanding. Such characteristic of comments could give many advantages to improve the performance of recommender systems. However, there are still challenges to effectively utilize comments for recommendation.

The first challenge comes from the well-known information overloading problem. Comments are useful in understanding the contents. However, if the amount is too large, it will become an annoying burden for people to digest all the information.

Besides, a large part of uninformative comments may mingle in the data collection. To alleviate the information overloading problem, we develop a method to extract key aspects of each movie from comments. Through the organized views of these aspects, users can easily understand other users' feedback without reading the whole comments thoroughly.

Second, users' different preferences have to be considered. Usually different opinions exist on a same movie. A user can agree or disagree with each other according to his/her preference. However, it is hard to know which opinion is agreeable for the user before he or she watches the movie. The author of the opinion is another important indicator to find agreeable comments. If we know other users with similar movie tastes, their comments should be more seriously treated. Because there are too many users and comments, it is practically impossible to manually identify other similar users. In our paper, we propose a method to automatically measure the similarity between users' preferences by analyzing and comparing users' rating histories.

Finally, an effective visualization is required to deliver recommendation and help users browse comments intuitively. The interface of our recommender system strongly supports the explanability so that the reason why the system recommends this particular movie to the user becomes clear. With this enhanced justification, the system should be able to provide highly trustable recommendations to users.

IV. SYSTEM DESIGN

In this section, we describe the MovieCommenter system, which is designed to resolve the aforementioned challenges. A recommendation by MovieCommenter is based on movie aspects. The movie aspects consist of (1) users' feelings about the movie, such as "exciting" or "disappointed", and (2) impressive features of the movie, such as "computer graphic", "actors", or "story". The movie aspects are extracted from comments and ratings are used to identify the sentiments associated with the aspects. Also, users' different preferences and relationships between them are considered in aspect extraction so as to achieve precise recommendations through better personalization.

MovieCommenter utilizes movie aspects to make accurate recommendations and presents them through an informative, visualized user interface. The detailed textual information in the aspects can lead to greater sophistication in the recommendations in comparison to the recommendations of other approaches based only on numerical ratings. For example, it is easier to understand "many viewers said "*CG effect* is really *amazing*, but *story* is pretty *boring*" than "it received a low rating". Further, visualizing the aspects can help users better understand recommendations.

A. System Architecture

MovieCommenter consists of five main parts: evaluation collection, aspect extraction, key aspect identification, classification, and visualization. First, in the evaluation collection part, user's evaluation of a movie is stored in the database. Next, the aspect extraction part is responsible for extracting movie aspects from user comments through text analysis. Then, key aspect identification part measures two weights to determine important movie aspects and summarize the sentiment about each key aspect. It counts the appearance of each aspect in different comments (Weight1: Importance) and analyzes whether users' overall sentiment about the aspect is positive or negative (Weight2: Sentiment). Further, in the classification part, the recommendation is determined based on the user's preference and aspects of movie. Finally, the visualization part presents the recommendation and aspects of a movie through a visualized interface.

B. Aspect Extraction

The Aspect Extraction part is responsible for extracting aspects in each comment and analyzing a users' sentiment about the aspects. First, informative terms that represent users' feelings or features of each movie were extracted as aspects. Next, a users' sentiment about the aspects was identified based on a numerical rating inputted by the user with the comment containing the aspects.

When terms are extracted from comments, the terms have no connection to the original context. The objective terms that contain no particular sentiment like "actor", "director", or "story", would not provide much information to other users. It is difficult to know the commenter's intention of using these terms. For example, the term "actor" does not reveal whether his/her acting was considered good or bad in the movie. Also, with only the terms extracted from a comment, it is difficult to know the strength of sentiment. Commenters could have various strength of feeling when the terms were originally used. To measure the strength of sentiment about an aspect, some researchers applied complex natural language processing techniques. However, the process is cumbersome and highly language-dependent.

We utilize user comments to identify movie aspects and user rating to determine the strength of sentiment about the aspect. Our heuristic method is simple yet effective. It is based on the idea that a user' rating and comment for the same content are strongly related. If the rating given with the comment containing a word "actor" is 9 out of 10, we deduce that "actor" is a positive aspect in the movie.

Aspect Term Extraction

To extract aspect terms from comments, we first remove uninformative words. All the words except nouns, adjectives, and verbs are removed through morphological analysis [25]. Some frequently-appeared terms in the domain of movie with less specific meaning like "movie" or "point" are also dropped. In addition, it is necessary to deal with synonyms with identical or very similar meanings. The terms having the similar meanings are grouped together, for example, "Computer Graphic" and "Graphic" are replaced with "CG".

Sentiment Analysis about Aspect

It is possible that a user expresses different sentiments in the same comment. Thus, directly linking a user's rating and all the aspect in his or her comment could be wrong. For example, in the comment “The scenes are beautiful, but the story is boring”, “scenes” and “story” are two different tones. The user, who left this comment, probably gave an overall point between 5~7 considering both positive and negative aspects. Assigning the same rating to “scenes” and “story” is not reasonable. As the sentiment of aspects, “scenes” has to gets a value higher than “story”.

We propose a method to deal with comments having multiple tones. The transition words such as “however”, “but”, or “through” are the key words that signify the existence of multiple tones. The transition words with the meaning of contrast frequently appear in comments with multiple tones, and separate aspects having different tones in a comment. (We have found that among the randomly sampled 100 comments with multiple tones, 91 comments have at least one transition word, which can separate a comment into multiple pieces according to tones.

After the identification, our method assigns rating to each aspect in a comment according to the tone. A rating for assignment $r^{*}_{u,m,t}$ of term t used by user u for movie m is calculated as:

$$r^{*}{}_{u,m,t} = \alpha \cdot r_{u,m} + \beta$$

where $r_{u,m}$ represents the original rating giving by user u, the parameters α and β are differently determined as tones of aspects in the comment.

If the comment does not contain any transition word, the original rating is assigned to all the terms in this comment. Therefore, α and β are set to 1 and 0 respectively. On the other hand, in case of comment containing at least one transition word, a comment is divided into multiple segments based on the transition words. The tones of each segment are then identified. We predefined a positive term set and a negative term set. If most terms within a segment are in the specific set, all the terms within the segment are identified as having the tone of the set. Finally, α and β are determined according to the overall tone of the set. The first parameter α is set to 0.5 regardless of the tone. And the second parameter β is set to 0 if the tone of the term is negative and 5 otherwise.

C. Key Aspect Identification

In the key aspect identification part, the key movie aspects are distinguished and sentiments about the aspects are summarized. For doing these processes, the quantitative and qualitative weights, *Importance* and *Sentiment*, are used.

The first weight, *Importance*, indicates how popular each aspect is by calculating the number of users who mentioned the same aspect in their comments. This weight is used to determine key aspects for a movie. We only select those aspects with higher Importance weights to represent the movie and make a recommendation. This *Importance* weight only provides the quantitative meaning of comments without showing any positively or negatively aspects.

The second weight, *Sentiment*, represents how the aspect is rated by others. It is the average of ratings given by the users who mentioned the aspect. If the value is high, it means that the aspect is a positive element in a movie.

The above weights are personalized by User Similarity, which indicates the distance between two users’ preferences. The evaluations by users who have similar tastes on movies are considered more important. User Similarity is calculated by comparing users’ historic rating records. As they give more similar ratings for the same movies, the higher their User Similarity is. To calculate User Similarity, we use Cosine Similarity, a measure widely used in collaborative filtering [1]. The range of the value is from 0 to 1. As the value is closer to 1, two users have more similar preferences.

$$UserSimilarity_{u,n} = \frac{\sum_{i\in R_{u,n}} r^{*}{}_{u,i} r^{*}{}_{n,i}}{\sqrt{\sum_{i\in R_u} r^{*}{}_{u,i}{}^{2}}\sqrt{\sum_{i\in R_n} r^{*}{}_{n,i}{}^{2}}}$$

The Importance is calculated by summing the similarities of users who used the term in their comments. The following equation indicates the Importance of a term t in comments on a movie m for a user u. In the equation, $N_{m,t}$ represents a set of the users who used term t in a comment on a movie m.

$$Importance_{u,m,t} = \sum_{n\in N_{m,t}} UserSimilarity_{u,n}$$

Another weight, Sentiment is the average of rating weighted by the user similarity. The Sentiment of term t used by user u for movie m is calculated as:

$$Sentiment_{u,m,t} = \frac{\sum_{n\in N_{m,t}} r^{*}{}_{n,m,t} \cdot UserSimilarity_{u,n}}{\sum_{n\in N_{m,t}} UserSimilarity_{u,n}}$$

Finally, a movie is represented as a vector consisting of the key aspects based on their weights. The movie vector contains the top 20 aspects with the highest Importance weight, and each aspect has Sentiment weight as its value. These movie vectors are differently constructed for each user in order to personalize recommendation.

D. Classification

The classifier models are constructed for determining the class c of a movie as *Recommendable* or *Unrecommendable*. These models are built for each user based on their own rating and commenting history. We tested two different algorithms for classification: Support Vector Machine (SVM) and Naïve Bayesian (NB).

We trained a set of movie vectors, which were rated by a user and labeled as *Recommendable* or *Unrecommendable*. If a user’s rating on a movie is more than or equal to a threshold, the related vectors will be labeled as *Recommendable*, and others are labeled as *Unrecommendable*. To determine the threshold, we invited 30 people to give the minimum rating for satisfactory movies on a 10-point scale. Most of them answered seven points (mean: 7.15, standard-deviation: 0.69). Based on this user survey, we set the threshold to determine Recommendable and

Unrecommendable as 7.

Support Vector Machine Classifier

Support Vector Machine is a supervised method. For training, the movie vectors with labels (*Recommendable* and *Unrecommendable*) are inputted to the classifier for each user. The Sentiment value of each aspect is used as the weight of feature in SVM Classifier. Then, the classifier tries to find an optimal hyper-plane separating the classes of the movie vectors.

We used the implementation of the SVM multiclass [26]. We selected the linear kernel, which is recommended when the number of features is large, and searched for a good regularization parameter through cross validation.

Naïve Bayesian Classifier

This method adopts the language modeling approach [23]. First, it constructs a probabilistic model based on the training set. The model then analyzes how words are likely to appear in each class. The probability that aspect a would appear in a class c_i is calculated as:

$$P(a|c_i) = \frac{tf(a, c_i)}{|c_i|}$$

where $tf(a, c_i)$ indicates the frequency of aspect a in the training set for class c_i while $|c_i|$ is the total number of terms in the training set for class c_i.

Based on the probability model, the classifier determines the class of each movie by choosing the class that makes the highest probability value. The probability that a movie vector m_k belongs to c_i is the product of the probability $P(a|c_i)$ of its words. Sometimes, an aspect in the target movie to determine recommendation does not appear in the training set for any class. In this case, even though all the other aspects in the movie are more likely to appear in the class, the total probability is always zero. To deal with the problem, we assigned a non-zero probability following the Laplace smoothing technique [16].

$$P(m_k|c_i) = \prod_{a \in m_k} P(a|\, c_i)$$

In the suggested method, the classifier can discriminate the aspects with the same term, but in different tone. For example, an aspect "actor" may have different tones according to movie. If the classifier counts "actor" without distinction of its sentiment, the movie is wrongly understood. Our method converts a term "actor" to "actor (Pos.)" if the sentiment weight is higher than or equal to 7, and vice versa for "actor (Neg.)". Therefore, they have their own probabilities.

E. Visualization

Finally, the recommendation and aspects about each movie are delivered to users in the visualization part. We developed the user interface of MovieCommenter in order to enhance the explanability of recommendation.

The explanability, an ability to give the reasons why a movie is recommended, positively affects the performance of a recommender system. As the explanability of the system increases, people show more trust in the recommendation. In addition, in case that the system provides incorrect recommendation, users are given a new chance to reconsider based on the explanation.

Generally, to generate the explanation, the distribution of numerical ratings or similarity with users who rated the movie is shown. These approaches enable users to have a quick and intuitive understanding of the recommendation because it is represented as numbers. However, they do not provide direct information about the key features of a movie. Our system additionally provides textual information to help people better understand the principles that govern the recommendation of the movies. In our interface, the aspects are sorted and displayed according to their weights (*Importance* and *Sentiment*). Showing the key aspects explains not only the reason why the movie is recommended but also the main features of the movie and other viewers' feelings.

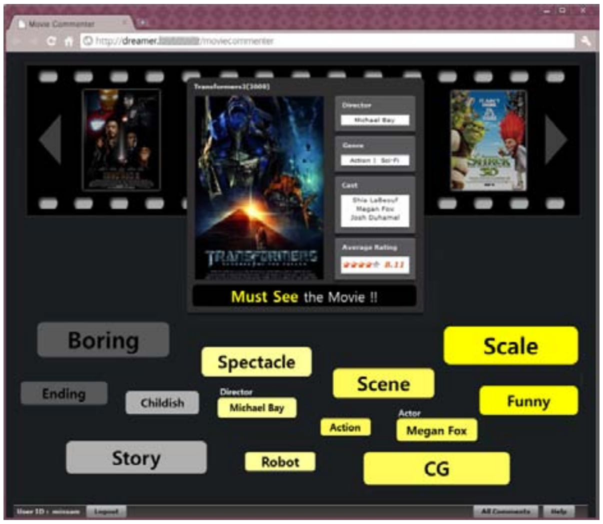

Fig. 2. Main Interface of MovieCommenter

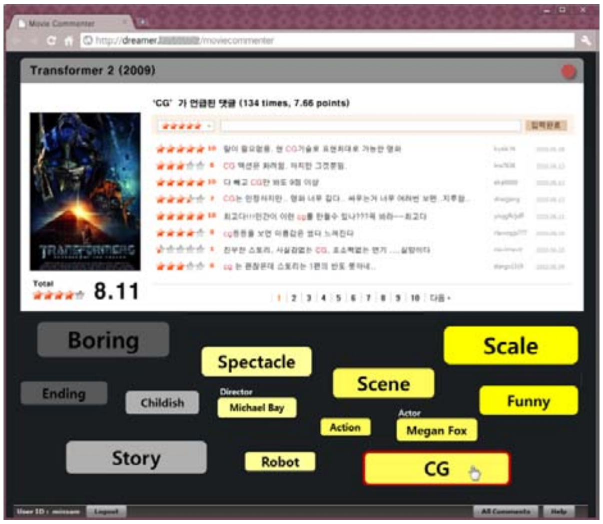

Fig. 3. Comment Browsing in MovieCommenter

Fig. 2 shows the main interface of MovieCommenter. By clicking a movie, the recommendation about the movie along multiple perspectives is presented to users. At the top of the interface, the basic description is shown. The recommendation message is located in the middle of the screen. The color of the mark is changed to bright yellow if the movie is strongly recommended.

Key aspects are displayed at the interface bottom. To effectively deliver these aspects, we present two weights of each aspect by two dimensions: *size* and *color*.

The size of a term is determined by *Importance*. The aspect takes up greater space in the screen if the aspect is mentioned by many users. Their user similarities with the target user further affects the size of the aspect.

The background color of a term reflects *Sentiment*. The color is closer to bright yellow if the aspect is positive. Otherwise, it is dark grey. Also, all the positive aspects are located in the right side.

The number of aspects affects users' understanding. If too many terms are shown concurrently, it will be burden for users to digest all of them. We tried to find the ideal number of aspects through experiments. Based on the experiments with 30 participants, we found that the most appropriated number is 10~15. When we set the number as 5, the participants are dissatisfied because of the lack of information. However, if there are too many terms, e.g., 20 terms, users complain that they are distracted and the screen is crowded.

If a user clicks one of the aspects, original comments from which the aspect is formed are presented in the form of list (See Fig. 3). The combination of the structured form and list, facilities the browsing of the comments and gives an opportunity for a thorough understanding.

V. EVALUATION

In this section, we discuss the performance of our system in terms of both system and users. First, we studied whether the extracted terms accurately represent the aspects of a movie. Second, we performed an accuracy test on recommendation results. For both tests, 2,756 users who evaluated at least 5 movies in 2010 were sampled from Naver Movie. We collected 132,962 ratings, entered by all the sampled users within the last 5 years. Finally, the effectiveness of the interface was examined through user studies.

A. Aspect Extraction

First, we evaluated whether the terms extracted from user comments could represent the aspects of a movie effectively. Because the recommendation in our system is determined based on a set of aspects, extracting representative aspects is the first-step and the most important task.

To determine the relevance of the extracted aspects, we recruited five people, who were college students at the same university in Korea.

First, according to our method, we prepared total 200 terms about 5 movies (40 terms for one movie) as key aspects about each movie. To do that, we used the sampled data from Naver Movie (user comments and rating data about the 5 movies) and additionally collected participants' ratings about 10 arbitrary movies in order to calculate *User Similarity*, which is required to calculate two weights, *Importance* and *Sentiment*. These aspects are given to the participants in a random order. For comparison, we prepared the top 40 terms for each movie based on TF-IDF weights and gave the terms to the participants in the same way. We then asked participants to check each term and to determine which movie aspects the term represents. A term is associated with an aspect if it contains sentimental information (e.g., "boring" and "exciting") or main features of a movie (e.g., "actor", "scene" and "CG").

Next, we calculated the precision for each participant while varying the total number of terms from 5 to 40, according to the rank of *Importance* and TF-IDF. We define precision about aspect extraction as:

$$precision\ @\ top\ k = \frac{\#\ of\ relevant\ aspects}{k}$$

Fig. 4 shows the average precision of 5 participants. *Importance* is more effective in representing key aspects of movie rather than TF-IDF. On the high ranking terms, the extraction by *Importance* records considerably high precision (rank 5: 0.98, rank 10: 0.96), while the extraction by TF-IDF also shows good performance (rank 5: 0.93, rank 10: 0.88). As the ranking of terms gets lower, however, the gap between the performances of two methods becomes wider. On the total 40 terms, the precision of the extraction by *Importance* is near 15% higher than the extraction by TF-IDF.

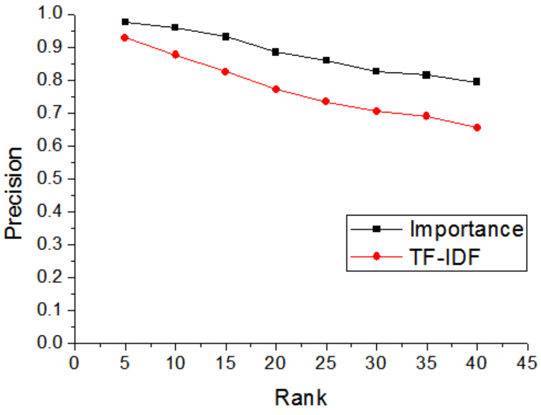

Fig. 4. Precision of Aspect Extraction

B. Accuracy of Recommendation

To evaluate the accuracy of recommendation suggested by our methods, we constructed the classifier models for the 2,756 sampled users. For each user, we sorted the evaluations (ratings and comments) by time and used the top 50% to construct the classifier model, namely training data. Other remaining 50% ratings were reserved for model validation. We compared our methods with other methods generally used in the recommendation area.

Algorithms to Be Compared

We compare our method with two traditional methods. The first one is the *collaborative filtering (CF)* approach. This approach estimates a user's rating about a movie based on the average of numerical ratings weighted by *User Similarity*. We used Cosine similarity to calculate *User Similarity*.

We also compared our approach with *content-based filtering (CBF)* approach. We tested CBF with two different input data: movie descriptions and user comments.

We weighted terms by TF-IDF and made a vector for each movie. Then, we labeled each vector as *Recommendable* if a user's rating for the movie is more than or equal to 7 and *Unrecommendable* otherwise. Similar to our method, the classifier model is constructed for each user. We adopted Support Vector Machine (SVM) and Naïve Bayesian (NB) classification methods to set up the classifier models.

Accuracy Measures

First, we calculated the classification accuracy of each method. The accuracy for a user is defined as the number of correctly classified movies divided by the total number of movies in the test set. We also measured precision, recall and F-measure which is the harmonic mean of precision and recall about the *Recommendable* class. Table II describes the accuracy measure.

TABLE II
ACCURACY MEASURES

		Prediction	
		Recommended Movie	Not Recommended Movie
Class	Recommendable	A	B
	Unrecommendable	C	D

Accuracy = (A+D) / (A+B+C+D)

Precision = A / (A+C)

Recall = A/ (A+B)

F-measure = (2 * precision * recall) / (Precision + Recall)

Results

Table III shows the accuracy of each method. The results show clearly that our approach makes the most accurate recommendation compared to others. All our proposed methods achieve more than 75% accuracy, while the accuracies of other methods stay lower than or equal to 70%. The CBF methods utilizing comments make better recommendation than the CBF method using description does. However, our method utilizing comments and ratings makes better recommendation than the CBF method using only comments. The CF method achieves the best precision among all the compared methods, but its accuracy is still lower than those of our methods.

Besides the classification accuracy, our two methods show the highest F-measure values, and the CBF methods using comments make recommendation better than other CBF methods that only use the movie description text. Especially, our methods show the ability to cover all movies that users should watch. They have high recall values, which are higher than 0.85. The CF method shows precision slightly better than our methods, however, its recall is much lower.

TABLE III
RECOMMENDATION ACCURACY

	Weights	Input Data	Accuracy	Recommendable movies		
				F-measure	Precision	Recall
CF	Cosine Similarity	Rating	0.7018	0.7795	0.8141	0.7477
CBF (SVM)	TF-IDF	Desc.	0.6730	0.7351	0.7870	0.6896
		Comments	0.7004	0.7647	0.7837	0.7466
CBF (NB)	TF-IDF	Desc.	0.6417	0.7283	0.7662	0.6939
		Comments	0.6793	0.7659	0.7767	0.7554
MC (SVM)	**Importance &Sentiment.**	**Rating & Comments**	**0.7629**	**0.8177**	**0.7854**	**0.8527**
MC (NB)	**Importance &Sentiment**	**Rating &Comments**	**0.7785**	**0.8308**	**0.7957**	**0.8692**

Next, we analyzed the results with recommendation ranking. Even though we classify movies into two classes, e.g., *Recommendable* and *Unrecommendable* classes, the recommendation results can be ranked. We ranked the results of each method separately. The recommendations by the CF method are ranked by how many users are involved and how largely they have similar tastes. For methods using SVM classifier (MC-SVM and CBF-SVM), we used the discriminant value to rank the recommendations. The SVM classifier generates the discriminant value for each class, and chooses the class with the highest value. Lastly, we used the posterior probability to rank the recommendations based on the NB classifier (MC-NB and CBF-NB). We measured the accuracy of recommendation, by varying the ranking from top 10% to top 100%.

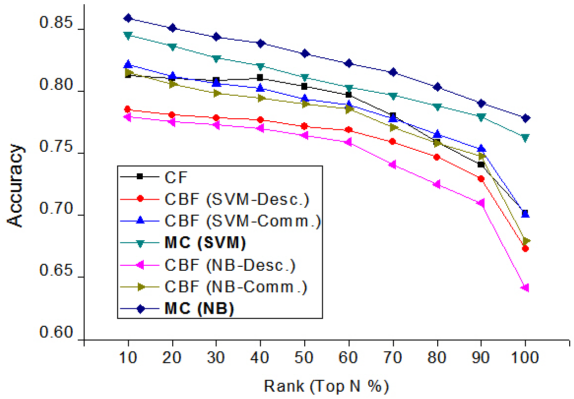

Fig. 5. Recommendation Accuracy by Ranking

Fig. 5 shows the comparison results. Along each ranking section, our suggested methods show the significant accuracy improvement when compared with other methods. In addition, our methods have a stable accuracy with small gaps between the higher ranks and the lower ranks. Our methods only drop 8% from the top 10% to 100% rank, while the other methods drop more than 11~13%.

Finally, we investigated into whether our method can make effective recommendations for those movies that are hard to decide (that is, viewers' opinions are controversial). If most people's opinions about a movie are similar, e.g., positive or negative, we could easily make the recommendation. However, it is a challenging task to make the decision if a movie attracts the evenly matched feedback from both sides: positive and negative. In this case, the role of the recommender system is more important. We grouped the movies in the test set into four sub-sets according to the proportion of the positive ratings (ratings >= 7) to the negative ratings and measured the average accuracy for movies in each set. The accuracy for a movie is defined as the number of correctly classified cases divided by the total number of the evaluation about a movie. Fig. 6 shows the results of the test data. For a better presentation, we selected one best method from each approach: the Collaborative Filtering, the Content-based Filtering (Text source: comment, Classifier: SVM), and MovieCommenter (Classifier: Naïve Bayesian)

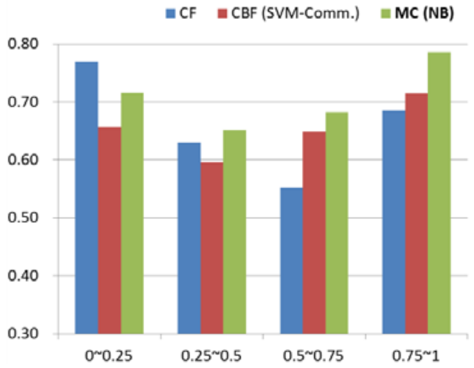

Fig. 6. Accuracy According to Proportion of Positive Ratings

Every method makes good recommendations on the movies that the proportion of the positive ratings is very low (0~25%) or very high (75~100%). However, for the movies having the similar number of positive and negative evaluations, the accuracy of every method is decreased. Especially, the accuracy of the CF method shows the biggest difference according to the proportion. The CF method achieves the highest accuracy on the movies that most people negatively evaluated, but shows the lowest accuracy on the movies that the proportion of the positive ratings is from 50~75%. The reason should be that the CF method only uses numerical rating data. On the other hand, the accuracy of the CBF method based on textual information is relatively consistent.

The accuracy of our method is consistent similar to the CBF method, but higher in entire range, which is benefited from the effective combination of numerical ratings and textual information (comments). In addition, on the movies that the proportion of the positive ratings is from 0.5 to 0.75, our method generates the best result. This range covers the most important and challenging recommendation tasks, because more than 40% of the sampled movies are located in this group and these movies with diverse ratings will introduce more burdens to people who are looking for a movie.

C. *User Studies on Interface Designing*

To evaluate the system interface, we performed a user study by recruiting 30 participants (college students), who were familiar with online movie web sites. The ages of participants ranged from 21 to 30 and they had diverse academic backgrounds. The objective of the study was two-fold: To assess (1) the explanability about the recommendation and (2) the helpfulness in browsing comments. Questionnaires and interviews were used to collect the user evaluation. In the questionnaire, users were asked to express their agreements with questions on a Likert scale between 1(not at all) to 5(extremely). In addition to questionnaires, we conducted interviews with them and noted the underlying reasons of their responses.

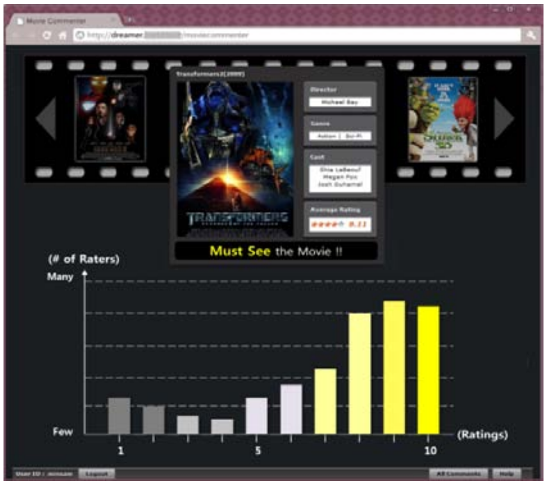

Fig. 7. Alternative Interface Showing the Distribution of Ratings by Similar Users in Collaborative Filtering

Before evaluating the explanability of our system, we analyzed whether our interface would precisely deliver the meaning of aspects about a movie because this is an essential feature of the proposed system.

We introduced the prototype of our system with three sample movies (each movie had more than 20,000 comments) to 30 randomly selected participants. First, we asked participants to play with our prototype system until they became familiar with it. We then asked the participants to evaluate the importance and sentiment of aspects through the interface.

Most of the participants (strongly) agreed that the meanings of aspects were represented clearly in the interface. Particularly, for the question about importance of aspects, every participant gave over 4 points (mean: 4.23, standard deviation: 0.47). In the interview, the participants agreed that bigger-sized words drew more of their attention. Thus it turned out to be an effective strategy to emphasize important terms using larger font sizes than other regular words. For the next question about sentiment of movie aspects, most participants also responded with positive answers (mean: 4.07, standard deviation: 0.83), but the score was slightly decreased because the participants had different concepts and preferences on colors. Some participants suggested that we use complement colors such as red and blue to represent different sentiments of aspects, while most participants were satisfied with the current colors: bright yellow and dark gray.

Next, we examined the explanability of our system. To compare the performance, we prepared another prototype giving an explanation as showing the distribution of ratings according to the number of similar users (See Fig. 7). We defined three criteria to measure the explanability as follows.

TABLE IV
CRITERIA TO MEASURE THE EXPLANABILITY

Criteria	Description
Justification	This explanation helps users understand why the movie is recommended.
Usefulness	This explanation provides useful information for users to understand about a movie itself regardless of being recommended or not.
Contextual Flexibility	This explanation helps users decide if the movie is suitable for their specific situation or mood.

Based on these criteria, we asked participants to evaluate how well the system elaborates the reasons for recommendation. We conducted interviews with a 5-scaled multiple choices to represent the degree of agreement with each criterion. Table V shows the results of this experiment.

TABLE V
AVERAGE DATA FOR 30 STUDY PARTICIPANTS ABOUT EXPLANABILITY (MEAN ± STANDARD DEVIATION)

	MovieCommenter	Compared Interface
Justification	4.00 ± 0.69	3.50 ± 1.11
Usefulness	4.27 ± 0.74	2.40 ± 1.00
Contextual Flexibility	4.27 ± 0.58	1.70 ± 0.65

With the questions about justification, the results show that our interface is slightly better than alternative interface. Many participants reported that the textual information in our system gives a more precise explanation than numerical information.

Our user study also had positive results from the questions about the usefulness of the system explanation. Most participants answered that the aspects in the screen certainly contributed to an understanding of a movie. In contrast, for the alternative interface, most participants complained that it did not offer any metadata to understand the main features of a movie.

Similar results were found for the last criterion, contextual flexibility. Most of the participants answered that our system provided excellent references for them to make a smart decision. Key aspects shown here allowed them to apply their own knowledge and inference skills during the process of completing decision. On the other hand, participants answered that the alternative interface seemed to contain only numbers for rating, providing insufficient information to make their decision.

VI. CONCLUSION

Our approach has been found successful in producing more accurate recommendations than those made with traditional approaches. Users were able to understand both recommendation and content better through the informative view of the aspects, which were extracted from user comments. We expect our approach to be readily applicable to recommending other entertainment contents such as books and music. Future research needs to examine this possibility.

ACKNOWLEDGMENT

This treatise was supported by the project of Global Ph.D. Fellowship which National Research Foundation of Korea conducts from 2011.

REFERENCES

[1] Adomavicius, G. and Tuzhilin, A., “Toward the Next Generation of Recommender Systems: A survey of the State-of-the Art and Possible Extensions,” *IEEE Transactions on Knowledge and Data Engineering* 17, 3(2005), 734-749.

[2] Adomavicius, G. and Kwon, Y., “New Recommendation Techniques for Multi-Criteria Rating Systems,” In *Proc. IEEE Intelligent Systems*, 2007.

[3] Chen, J., et al., “Short and Tweet: Experiments on Recommending Content from Information Streams,” In *Proc. CHI*, 2010.

[4] Claypool, M., et al., “Combining Content-Based and Collaborative Filters in an Online Newspaper,” In *Proc. SIGIR*, 1999.

[5] Debnath, S., et al., “Feature Weighting in Content Based Recommendation System Using Social Network Analysis,” In *Proc. WWW*, 2008

[6] Faridani, S., et al., “Opinion Space: A Scalable Tool for Browsing Online Comments,” In *Proc. CHI*, 2010.

[7] Ganu, G., et al., “Beyond the Stars: Improving Rating Predictions Using Review Text Content,” In *Proc. WebDB*, 2009

[8] Golbeck, J. and Hendler, J., “FilmTrust: Movie Recommendations using Trust in Web-based Social Networks,” In *Proc. IEEE Consumer Communications and Networking Conference*, 2006.

[9] Herlocker, J.L., et al., “Explaining Collaborative Filtering Recommendations,” In *Proc. CSCW*, 2000.

[10] Konstan, J.A., et al., “GroupLens: Applying Collaborative Filtering to Usenet News,” *Communications of the ACM* 40, 3(1997), 77-87.

[11] Leung, C., W., et al., Integrating Collaborative Filtering and Sentiment Analysis: A Rating Inference Approach. In *Proc. ECAI-workshop on Recommender Systems*, pages 62-66, 2006

[12] Linden, G., Smith, B. and York, J., “Amazon.com Recommendations: Item-to-Item Collaborative Filtering,” *IEEE Internet Computing*, Jan./Feb. 2003.

[13] Lu, Y., Zhai, C. and Sundaresan, S., “Rated Aspect Summarization of Short Comment,” In *Proc. WWW*, 2009.

[14] Melville, P., et al., “Content-Boosted Collaborative Filtering for Improved Recommendations,” In *Proc. AAAI*, 2002.

[15] Miller, B.N., et al., “MovieLens Unplugged: Experiences with an Occasionally Connected Recommender Systems,” In *Proc. IUI*, 2003

[16] Mitchell, T.M., *Machine Learning*, McGraw Hill. 1997

[17] Mooney, R.J. and Roy, L., “Content-Based Book Recommending Using Learning for Text Categorization,” In *Proc. SIGIR*, 1999.

[18] Pang, B., et al., *Opinion mining and sentiment analysis, Foundations and Trends in Information Retrieval 2*, 1-2(2008), 1-135.

[19] Park, S., et al. “The Politics of Comments: Predicting Political Orientation of News Stories with Commenters’ Sentiment Patterns,” In *Proc. ACM CSCW*, 2011

[20] Sen, S., Vig, J. and Riedl, J., “Tagommenders: Connecting Users to Items through Tags,” In *Proc. WWW*, 2009.

[21] Tsagkias, M., et al., “News Comments: Exploring, Modeling, and Online Prediction,” In *Proc. ECIR*, 2010.

[22] Vig, J., et al., “Tagsplanations: Explaining Recommendations Using Tags,” In *Proc. IUI*, 2009.

[23] Yamron, JP., et al., “Statistical models of topical content,” *The Kluwer International Series On Information Retrieval*, pages 115-134, 2002.

[24] Yano, T. and Smith, N.A., “What’s Worthy of Comment? Content and Comment Volume in Political Blogs,” In *Proc. AAAI*, 2010.

[25] Korean Morphological Analyzer MACH 1.0. Available online at http://cs.sungshin.ac.kr/~shim/demo/mach.html

[26] SVM multiclass classifier. Available online at http://svmlight.joachims.org/svm_multiclass.html