MovieMine: Personalized Movie Content Search by Utilizing User Comments

Hyung W. Kim, Keejun Han, Mun Y. Yi, Joonmyun Cho, and Jinwoo Hong

**Abstract — ***User comments are one of the common online resources that reflect users' evaluative opinions about multimedia contents. The words used in user comments provide many clues about the users and about what they like or dislike. In this paper, we propose a novel query expansion method that utilizes user comments in order to consider user's different preferences in finding movies. We propose a personalized search system, called MovieMine, built upon this proposed method to provide personalized search results by expanding queries on the basis of earlier comments left by the user. Using an actual movie review dataset obtained from a large movie portal, we show that our system produces a significant performance improvement compared to the baseline condition. We expect our approach to be readily applicable to personalized searching of multimedia contents.* [1]

Index Terms — Personalized content search, user comments, automatic query expansion, recommender systems

I. INTRODUCTION

Connected TV, which is also known as smart TV, vastly expands the function of television sets by integrating the Internet and Web 2.0 features into contemporary television sets and set-top boxes. TV users can now access a wide range of contents not only from traditional broadcasting services but also from the Internet through a single device. While the availability of numerous contents on a TV means more choices, it also poses a great challenge to its users as they have to decide what to watch out of an almost infinite number of competing choices, highlighting the importance of content searches or recommendations that consider each user's individual preferences.

In the context of a recommender system, various studies have been conducted in an effort to recommend proper contents to connected TV users in accordance with their individual preferences [1][2]. In contrast, there has been relatively little effort to develop a personalized search method primarily focused on media contents. Personalized searches have been studied not for multimedia contents but for documents or pages on the Internet.

There are several popular movie search systems on the Web; however, they have notable problems, as illustrated below:

- Problem 1: The service does not return any search result if the search criterion is not supported by the movie metadata, which is pre-defined by the service provider. For example, Peter wants to find movies with twist endings, but he has no other information about the specific movies. Unfortunately, he is not able to find such a movie while using the current movie search system because he cannot provide the movie title, the actors, the director, or other movie-specific information predefined in the meta-data.

- Problem 2: The service returns the same search results to everyone regardless of individual preferences pertaining to movies. Peter, for instance, wants to find some touching movies so he searches for movies with the query *touching*. His search results consist of movie titles with the word *touching* partially included. The titles are mostly about romance; however, he dislikes romance movies. He wants to find non-romance touching movies not necessarily entitled touching. He does not know where to go.

Both problems may be solved using the simple techniques of manual classification, which involves hiring of people to extract additional features from movies and classify them manually, and self-user profiling, which asks users to complete a profile form when they register for the system and update it over time. These techniques are, however, costly and unreliable because they demand extra manpower, still with the potential of missing certain features even after significant commitment of time and effort. Moreover, many users just do not want to fill out a profile form.

Since the era of Web 2.0, user participation has been much more active on the Internet, encouraging users to freely exchange their ideas, post their opinions, share their favorite contents, and show their interests to the public. User comments are one of the common online resources that reflect users' evaluative opinions about multimedia contents. This new user-created source is a key to solve the problems described above

[1] This work was supported by the ETRI R&D Program of KCC (Korea Communications Commission), Korea [11921-03001, "Development of Beyond Smart TV Technology"]".

Hyung W. Kim is with the Next Generation Smart TV Research Department, Electronics and Telecommunications Research Institute (ETRI), Daejeon, South Korea (e-mail: h.w.kim@etri.re.kr).

Keejun Han is with the Knowledge Service Engineering Department, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea (e-mail: ham1212@kaist.ac.kr).

Mun Y. Yi is with the Knowledge Service Engineering Department, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea (e-mail: munyi@kaist.ac.kr).

Joonmyun Jo is with the Next Generation Smart TV Research Department, Electronics and Telecommunications Research Institute (ETRI), Daejeon, South Korea (e-mail: jmcho@etri.re.kr).

Jinwoo Hong is with the Next Generation Smart TV Research Department, Electronics and Telecommunications Research Institute (ETRI), Daejeon, South Korea (e-mail: jwhong@etri.re.kr).

Contributed Paper
Manuscript received 10/15/12
Current version published 12/28/12
Electronic version published 12/28/12.

because the keywords included in user comments can provide many clues about users and about what they like or dislike. The automated analysis of the user comments neither requires extra manpower nor laborious work. Despite its potential impact for personalization in the information retrieval field, however, user comments have not been applied to personalized search, to the best of our knowledge.

This paper introduces a new personalized movie search system, which is devised to produce personally relevant search results and to provide finer search functions. The proposed system derives user profiles by utilizing users' textual comments as well as ratings about the movies. The user profiles can then be used for query expansion in order to maximize individual preferences towards movie contents in the final search results. Further, the proposed system expands the capability of traditional search engines by enabling mode-specific (e.g., moving, gloomy), material-specific (e.g., zombie, robot), and other content-based searches. In traditional movie search systems, the service provider must manually enter this information at the cost of both money and time. On the other hand, our system automatically extracts this information from each movie by analyzing user comments left for the movie.

For the performance evaluations of the proposed method, we used two different measures: *precision* and *satisfaction*, in conjunction with an actual movie review dataset obtained from a large movie portal in Korea. We stipulate that good performance means that the searched movies are relevant as well as satisfactory to the user.

In summary, the main contributions of our paper are as follows:

- We propose a novel method of user profile composition for movies, for more refined and personalized search.
- We demonstrate, employing an actual movie review dataset, that our algorithm effectively finds relevant movies and satisfies users.
- To the best of our knowledge, we present the first automatic query expansion (AQE) system based on user comments.

The rest of the paper is organized as follows: Section 2 reviews prior research related to our study. Section 3 provides an architectural overview of the proposed system. Section 4 explains the methodology of implementing our personalized search system. Section 5 evaluates the performance of our system. Finally, Section 6 concludes our study.

II. Background

Our proposed system brings together two research areas: personalization and information retrieval (IR). There exist many studies in both research topics, but only a few studies focus on utilizing user comments for a personalization search system. In this section, we first review several personalization techniques and then prior research on user comments focusing on the IR area.

A. Personalization

In general, the term, "personalization", means providing right contents to right users in accordance with their preferences [3].

Other personalized information retrieval studies mainly have identified user profiles by 1) utilizing users' dynamic inputs, 2) using previous queries [4] and click-through data [5][6], and 3) analyzing users' social network profiles [7]. To the best of our knowledge, no prior research adopted user comments as a main source for user profiling.

There are also two additional ways of utilizing user profiles for personalization: 1) query expansion by re-weighting the original query or adding new terms to the query based on the users' interests [8] and 2) re-ranking and filtering of the search results using user profiles [9]. In this paper, we use user profiles for query expansion in order to identify additional contents that were not included in the initial search results.

Recommendation is one of the active research areas where those of personalization techniques are used in content providing services. It aims at recommending items that users had not yet considered, but are likely to be preferred. Recommender systems can be generally divided into three types: collaborative filtering [10], content-based filtering [11] and hybrid approach, which uses both of the two methods [12][13][14]. Collaborative filtering recommends contents by analyzing the common patterns of multiple users who share the same interests [15][16][17].

Unlike collaborative filtering, content-based filtering only looks at a history of an individual user to generate recommendation. In this system, a target content recommended for a user is determined by comparing it with other contents the user rated in the past. If the target content is similar to those contents the user rated high in the past, the content is also expected to be preferred by the user [18][19]. Although content-based filtering is simpler and easy to analyze for recommendation, collaborative filtering generally shows better performance than content-based filtering.

Other than recommendation, there are not many personalization techniques used in the content providing services. In searching contents, specifically, there are still rooms for improvement through personalization. In our study, we propose a novel method to search contents (i.e., movies) by applying personalization techniques.

B. Utilizing user comments in IR

Outside of personalization, there are several studies using user comments as a main source for their experiments. Some studies show that the number of user comments posted on news [20] and blog posts [21] is an indicator of popularity. Recent studies also introduced several methods to identify useful comments [22][23].

Utilizing user comments more directly for search, Yee et al. [24] have examined the potential impact of user comments on search accuracy in social Web sites [24]. In this study, the user comments were used for generating an index of YouTube contents with the search accuracy increase by 15%. Going beyond the prior work, our method uses user comments not only to create an index for contents but also to analyze users' preferences for personalized search results.

III. MovieMine

A. Architecture of the system

Fig. 1 shows the overall architecture of the proposed movie search system. This system provides personalized search results by expanding queries based on an analysis of earlier comments left by the users. Our system has the following four main elements: a User Analysis Module, a Query Expansion Module, a Movie Indexer, and a Movie Ranking Score Computing Module.

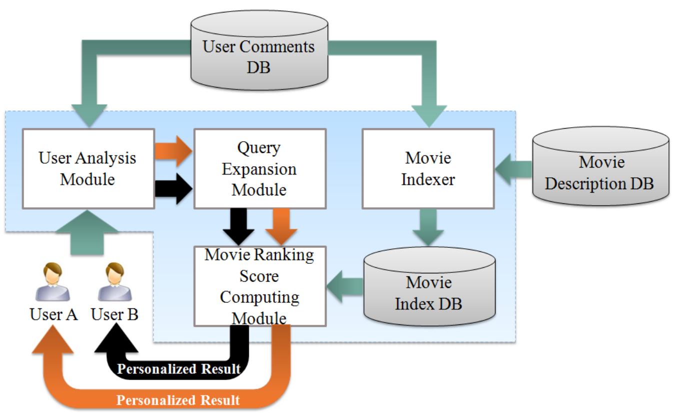

Fig. 1. System Architecture of MovieMine

The *User Analysis Module* extracts terms that an individual user prefers from their past user comments. A user comment consists of the text of a short length and a numeric rating score pertaining to the corresponding content (see Fig. 3). This module analyzes these comments in order to find the most relevant keywords related to the given query. For instance, if the given query is *touching* by user A, this module analyzes his or her past comments and extracts the keywords that are most relevant to the query *touching*.

The *Query Expansion Module* expands the given query with the keywords that are obtained by the *User Analysis Module.* If the most relevant terms in the initial query *touching* are *story, tear,* and *acting* in a weighted order with the most common term first, this module chooses one or more terms for query expansion, starting with the most common term.

Movie Indexer fetches movie descriptions and user comments related to each movie. A movie description includes the basic information of the movie such as its title, genre, actors, director, and synopsis. When creating a movie index for movie descriptions, *Movie Indexer* computes the weighting scores of the terms that appear in the movie descriptions by considering the representativeness of each term for the movie. On the other hand, for user comments, it calculates weighting scores of the terms in the user comments by adjusting its initial weighting score based on the numeric rating score assigned to the movie, adding more weight to those terms associated with higher rating scores for the given movie. This process is explained in detail in Section 4. The movie index is then stored in the *Movie Index Database (DB).*

The *Movie Ranking Score Computing Module* computes the final movie ranking score by calculating the similarity score between the query set and the corresponding movie index. It finally returns the movie search results to the user.

In summary, the *User Analysis Module* initially extracts terms that are relevant to the given query from user comments when a user inputs a query. The *Query Expansion Module* chooses at least one term from the list of terms obtained by the *User Analysis Module* in order to expand the initial query. The *Movie Ranking Computing Module* ranks the movies and then returns the personalized search results to the user.

B. Motivating Example

Let us assume that Peter and Kelly, who have different movie preferences, use our system. They want to search for a *touching* movie; however, they have no information about the movie title, the movie director, or the name of the actors when they begin their search to find their favorite movies. In our system, Peter and Kelly are provided different search results, although both enter the query *touching*.

In Peter's user comments, he frequently referred to music when he used the word *touching* in his comments (e.g., "the music was so touching at the last scene," "It was really touching! The music was also very good.") Thus, the term *music* is chosen as an expansion query term for Peter.

Meanwhile, Kelly frequently used the term *acting* with *touching* (e.g., "his acting was so touching," "I was so touched by his tears when he acted in this part") in her comments. Thus, the term *acting* is newly added to the current query set.

In the next process, the ranking score is computed between the given query sets and the movie indexes. The movies "Once" (2006) and "I Am Sam" (2001), which earned the highest scores, are returned to Peter and Kelly, respectively. Fig. 2 shows this personalized search process.

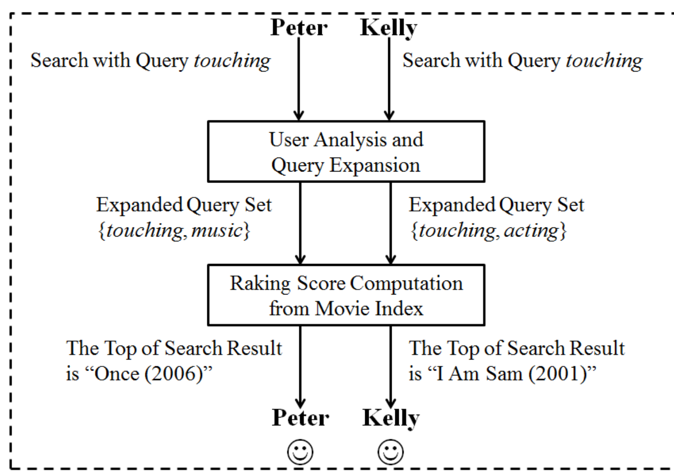

Fig. 2. An Example of Query Expansion Process

As shown in this example, our system finds movies even when a user does not have any basic information about the movies, such as their titles, genres, actors, or directors. In addition, it adds new terms to the initial query and expands the query in consideration of individual user preferences manifested in prior movie comments so that it can generate more personalized search results.

IV. Methodology

A. Automatic Query Expansion Using User Comments

Our system provides search results that reflect individual user preferences by means of an *Automatic Query Expansion (AQE)* method. AQE adds new queries to the initial query set by analyzing past query logs or profiles of individual users and then processes searches with this extended query set. In the proposed method, we use comments left by the user in the past because user comments contain more detailed, implicit information compared to query logs or user profiles.

Fig. 3 shows a typical content of user commenting system that is generally used on the Web. In the proposed system, we specifically adopt user comments with a short length requirement because terms are more meaningful in relatively short user comments. In other words, there can be many insignificant words in extended user comments. However, when the length is limited, users use more direct expressions to conform to the length limit [25].

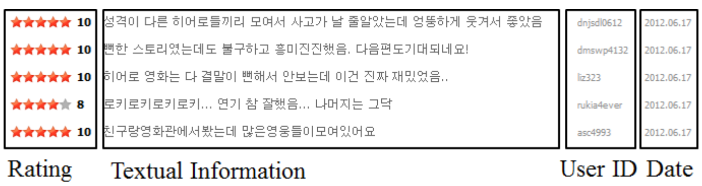

Fig. 3. An Example of User Comments

Fig. 4 shows the process that finds relevant terms to initial query term Q to be added for query expansion. First, our system loads the comments left by user A in the past from the User Comments DB (Table (a) in Fig. 4). Each comment is then split into several terms after stemming (Table (b) in Fig. 4). In the next step, the system only chooses user comments in which the query Q appears (Table (c) in Fig. 4). The query expansion weights of all terms in the chosen user comments are then calculated excluding the terms that are the same as the initial query term Q. To calculate the query expansion weight for each term in the user comments, we adjusted the Term Frequency – Inversed Document Frequency (TF-IDF) to the Term Frequency weighted by Rating Score – Inverted Comment Frequency (TFR-ICF) as follows:

$$w_{i,Q,A} = TFR_{i,Q,A} \times ICF_{i,Q,A} \tag{1}$$

$$TFR_{i,Q,A} = \frac{r_{i,Q,A}}{\sum_{k=1}^{n} r_{k,Q,A}} \tag{2}$$

$$ICF_{i,Q,A} = log \frac{|C_{Q,A}|}{|\{c_{Q,A} \in C_{Q,A} : term_i \in c_{Q,A}\}|} \tag{3}$$

$w_{i,Q,A}$, the query expansion weight of $term_i$ for user A about the query term Q, is calculated by multiplying $TFR_{i,Q,A}$ and $ICF_{i,Q,A}$, as shown in (1).

In (2), $r_{i,Q,A}$ is the sum of the rating scores of $term_i$ in user A's past comments in which the query Q appears. $r_{i,Q,A}$ is used instead of the term frequency (TF) in our system because a word in the user comments with a high score is more meaningful when seeking to capture user preferences. Thus, this metric prefers the words associated with high ratings. $\sum_{k=1}^{n} r_{k,Q,A}$ is the sum of all term frequencies weighted by rating score in the past comments of user A containing query A.

$|C_{Q,A}|$ in (3) is the cardinality of C, which is the total number of user comments of user A including query Q. $|\{c_{Q,A} \in C_{Q,A} : term_i \in c_{Q,A}\}|$ denotes the number of comments in which the $term_i$ appears in $C_{Q,A}$. $ICF_{i,Q,A}$ is a measure of whether the $term_i$ is common or rare across all user comments. For example, when there are three past user comments of user A, {Terms = {*Q, a, b, c, d*}, Rating Score = 10}, {Terms = {*Q, a, e, f*}, Rating Score=9} and {Terms = {*Q, c, d, f*}, Rating Score=7}, $w_{a,Q,A}$ and $w_{a,Q,A}$ can be calculated as follows (4) to (9):

$$\begin{aligned} TFR_{a,Q,A} &= \frac{r_{a,Q,A}}{\sum_{k=1}^{2} r_{k,Q,A}} \\ &= \frac{10+9}{(4\times 10)+(3\times 9)+(3\times 7)} = 0.216 \end{aligned} \tag{4}$$

$$\begin{aligned} ICF_{a,Q,A} &= \log \frac{|C_{Q,A}|}{|\{c_{Q,A} : term_a \in c_{Q,A}\}|} \\ &= \log \frac{3}{2} = 0.176 \end{aligned} \tag{5}$$

$$\begin{aligned} w_{a,Q,A} &= TFR_{a,Q,A} \times ICF_{a,Q,A} \\ &= 0.216 \times 0.176 = 0.038 \end{aligned} \tag{6}$$

$$\begin{aligned} TFR_{f,Q,A} &= \frac{r_{f,Q,A}}{\sum_{k=1}^{2} r_{k,Q,A}} \\ &= \frac{9+7}{(4\times 10)+(3\times 9)+(3\times 7)} = 0.181 \end{aligned} \tag{7}$$

$$\begin{aligned} ICF_{f,Q,A} &= log \frac{|C_{Q,A}|}{|\{c_{Q,A} : term_f \in c_{Q,A}\}|} \\ &= \log \frac{3}{2} = 0.176 \end{aligned} \tag{8}$$

$$\begin{aligned} w_{f,Q,A} &= TFR_{f,Q,A} \times ICF_{f,Q,A} \\ &= 0.181 \times 0.176 = 0.032 \end{aligned} \tag{9}$$

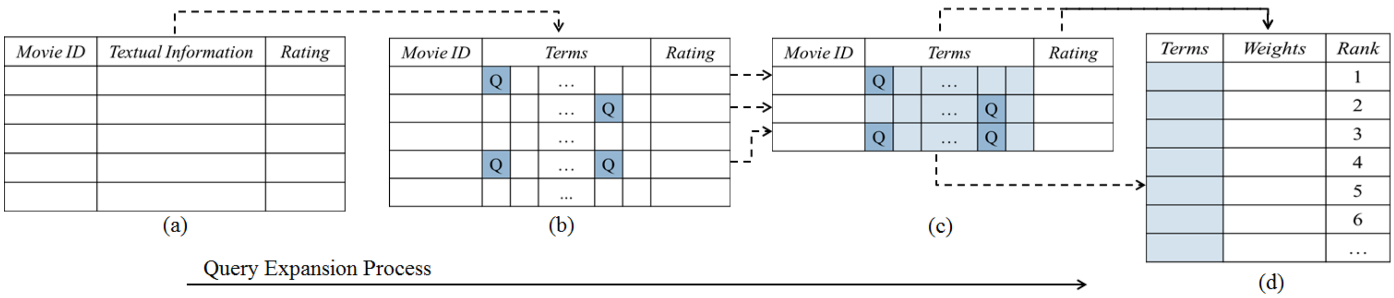

Fig. 4. Query Expansion Process for User A

As shown in (4) to (9), $\mathrm{term_a}$ and $\mathrm{term_f}$, which evenly appear a second time in three user comments, have different weights ($w_{a,Q,A} > w_{f,Q,A}$, $w_{a,Q,A} = 0.038$ and $w_{f,Q,A} = 0.032$). When the rating value associated with the user comments in which the terms appear is higher, we consider that the terms are more important to the user. Hence, the terms receive a higher weight.

After the weights of all of the terms in $c_{Q,A}$ are computed, those terms are re-ranked in descending order by their weights to create the query candidate table (3), as shown in Fig. 4. We then obtain the expanded query set Q' by adding at least one term to query set Q, which was initially created by User A. Thus, going back to the example above, the most important term a is added to the query set, creating a new query set Q', which is $\{Q, a\}$.

Table 1 demonstrates that the results of the query expansion that begin with the query *actor* are different for different users A, B and C. Although the initial queries of these three users were identical, their query expansion results are different because their movie preferences are different. If our system expands the initial query set with one term, the expanded query set of User A is {*actor, acting*}, which means that the query set searches for movies in which the performances of the actors are impressive. Meanwhile, the expanded query set of User B is {*actor, charming*}; this implies that the query set searches for movies in which the roles of its actors are charming. Lastly, the expanded query set of User C is {*actor, talented*}, which means it considers talented actors as the primary indicators in searching for movies.

In this section, we have explained the automatic method of expanding the initial query by considering users' differing preferences. In the next section, we explain our method of creating a movie index from movie descriptions.

TABLE I
TOP FIVE CANDIDATE EXPANSION TERMS OF QUERY "ACTOR" FOR THREE USERS

Rank	Expansion Term of User A	Expansion Term of User B	Expansion Term of User C
1	Acting	Charming	Talented
2	Old	Spectacle	Acting
3	Charisma	Story	Story
4	Voice	Handsome	Touching
5	Fun	Acting	Sorrow

We randomly chose three users from the dataset, as explained in the next section. The terms in the table were translated into English, as our original dataset was in Korean.

B. Generating a Movie Index with User Comments

To create a movie index, we used the movie description information (e.g., title, genre, actor, director, and synopsis) and the user comments (numerical rating and textual information) left for a movie.

Fig. 5 shows the overall process of generating the movie index. Here, we initially stemmed and spilt movie descriptions and user comments into their terms (see Tables (a), (b) and (c) in Fig. 5). $w_{i,A}$, which indicates the weight of $\mathrm{term_i}$, which appeared in the movie description and user comments of movie A, can be calculated by multiplying *Term Frequency weighted by Rating – Inverted Movie Frequency (TFR-IMF)* as follows:

$$w_{i,A} = TFR_{i,A} \times IMF_{i,A} \tag{10}$$

$$TFR_{i,A} = \frac{r_{i\in UC\ of\ A} + n_{i\in D\ of\ A}}{\sum_{k=1}^{N} r_{k\in UC\ of\ A} + \sum_{k=1}^{N} n_{k\in D\ of\ A}} \tag{11}$$

$$IMF_{i,A} = log\frac{|M|}{|\{m_A \in M_A : term_i \in m_A\}|} \tag{12}$$

In (11), $\mathrm{r_{i\in UC\ of\ A}}$ is the sum of the rating scores of $\mathrm{term_i}$, which appeared in the user comments of movie A. $\sum_{k=1}^{N} r_{k\in UC\ of\ A}$ is the sum of the term frequencies weighted according to rating score in the user comments about movie A. $\mathrm{n_{i\in D\ of\ A}}$ is the term frequency of $\mathrm{term_i}$, which appeared in movie A's description, and $\sum_{k=1}^{N} n_{k\in D\ of\ A}$ denotes the total number of terms appeared in movie A's description. Using (11), we can calculate the weight of $term_i$, which appeared in both the description and user comments. In particular, the significance of $term_i$ increases as the rating score in the user comment increases.

$|M|$ in (12) is the cardinality of M, which indicates the total number of movies in our dataset. $|\{m_A \in M_A : term_i \in m_A\}|$ is the number of movies in which the $term_a$ appears either in the description or in the user comments of movie A. $IMF_{i,A}$ is a measure of whether the $\mathrm{term_i}$ is common or rare across all descriptions and user comments about the movie.

Finally, we can calculate $w_{i,A}$ by multiplying $TFR_{i,A}$ and $IMF_{i,A}$. The value of $w_{i,A}$ represents the importance of $term_i$ for movie A.

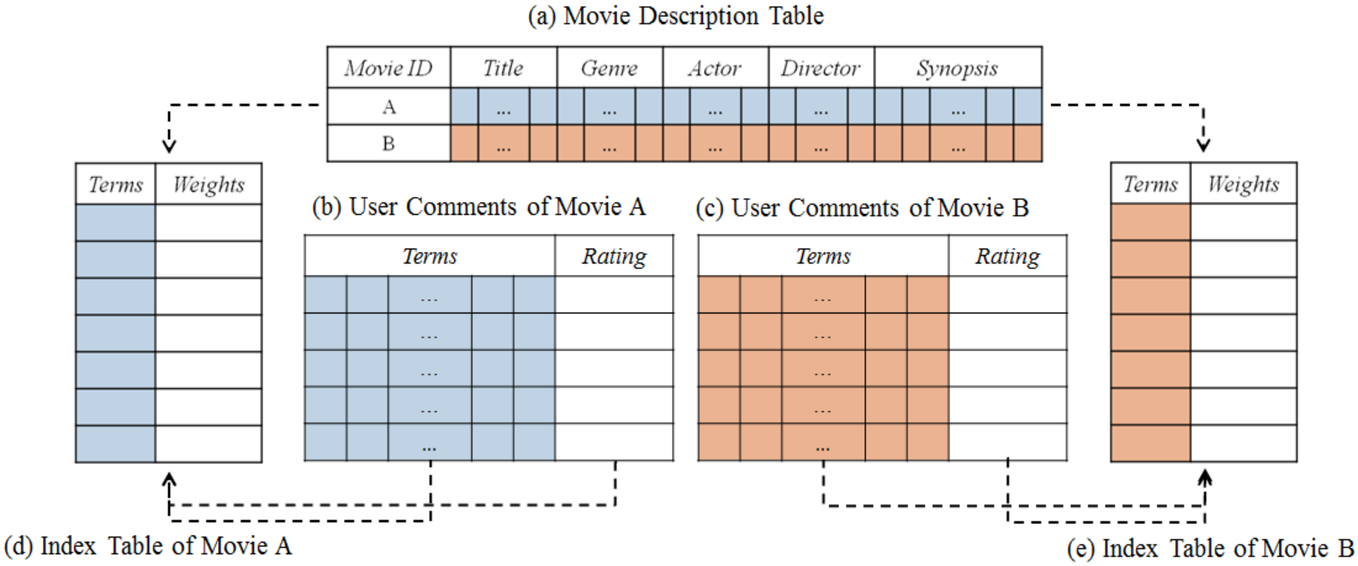

Fig. 5. Generation of the Movie Index via the Movie Descriptions and the User Comments

C. Computing Movie Ranking Score

The *Ranking Score (RS*) between the expanded query set Q′ and the movie A in the movie index can be calculated by means of *Cosine Similarity* (*COS*), which is commonly used in the IR field for computing similarities between two term vectors.

COS considers the expanded query set Q′ and the movie A in the index as vectors and then calculates the cosine angle between the two points in order to observe the likeliness of the two vectors.

The expanded query set Q′ can be represented as the vector $Q' = \{(qt_1, qw_1), (qt_2, qw_2), \ldots, (qt_n, qw_n)\}$, where qt_1 is the original query term and $qt_2, qt_3, \ldots, qt_n$ are the expanded queries terms. The values of the query terms' weights, $qw_2, qw_3, \ldots, qw_n$, differ based on the query expansion weights so that the normalized query expansion weights of qt_2 becomes 1.

Movie A's index can be represented as the vector $M_A = \{(term_1, w_1), (term_2, w_2), \ldots, (term_n, w_n)\}$, where w_n is the weight of $term_n$ as calculated by (10).

The correspondence of the expanded query set Q′ to the index of movie A is expressed as

$$RS\left(Q', M_A\right) = \frac{Q' \bullet M_A}{|Q'||M_A|} = \frac{\sum_1^n qw_i w_j}{\sqrt{\sum_1^k q_i}\sqrt{\sum_1^k w_i}} \tag{13}$$

where $\mathrm{w_j}$ is the weight of $\mathrm{term_j}$, which can satisfy $\mathrm{term_j} = \mathrm{qt_i}$. In addition, $\cdot$ denotes the scalar product and $|\,|$ denotes the magnitude of the vector. For example, when Q′ is {(action, 1), (story, 1), (zombie, 0.7)} and M_A is {(twisted-end, 2), (action, 4), (robot, 3), (story, 2)}, the ranking score between Q′ and M_A can be calculated as follows:

$$RS\left(Q', M_A\right) = \frac{(1+4)+(1+2)}{\sqrt{1^2+1^2+0.7^2}\sqrt{2^2+4^2+3^2+2^2}} \tag{14}$$

$$= 0.883$$

Following the same procedure, we can compute the RS of Q′ for the movies remaining in the movie index and return the search results from the highest RS value.

V. Evaluation

We evaluate the performance of the proposed system, *MovieMine,* with two measures: *Precision* and *Satisfaction*.

- *Precision*: This measure evaluates the degree to which the movies from the search results are relevant to the given query.
- *Satisfaction*: This measure evaluates the degree to which a user prefers the movies from the search results.

A. Data Acquisition

For the evaluation of our proposed system, we used an actual movie review dataset available from one of the largest portals in Korea. The data was deemed to be appropriate to assess the performance of the proposed system because it has the following characteristics of user commenting systems.

- It enables users to score a movie with a numerical value from 1 to 10.
- It allows users to write short comments of up to 140 Korean letters (140 English letters also).
- It allows users to leave only one comment for each movie, preventing redundancy.

The movie data were collected by our web crawler developed for this research and it included the movie descriptions (i.e., the title, genre, director, actor, release date and synopsis) as well as the users' comments for each of the movies listed on the portal site from January 2005 to May 2010.

Our dataset contains 2,269 movies and 2,189,989 user comments written by 883,583 users in total. For the evaluation, users who had left comments for more than 50 movies were chosen for the sample user set out of the total user set so that user profiles can be built with sufficient comments. As a result, the final dataset was reduced to 1,658 movie titles and 117,014 user comments (text comments with ratings) entered by 1,335 users. The User Comments DB, Movie Description DB, and Movie Index DB were generated using this dataset. Table 2 presents the descriptions of the dataset used for our experiments.

TABLE II
DATASET DESCRIPTIONS

Type	Amount
Total # of Movies	1,658
Users	1,335
User Comments	117,014
Avg. # of the Characters in a User Comment	45.53 (Korean)
Average Rating Scores	7.33
Standard Deviation of Rating Scores	2.28

B. Experiment Query

To obtain query terms for the performance evaluation, we recruited 30 people (undergraduate and graduate college students in a major research university in Korea) and asked them to write down 15 query terms but not limited to movie titles, genres, actors, or directors. As a result, we collected a total number of 167 unique terms, of which the top 20 frequent terms were chosen for the experiment. Table 3 shows the chosen query terms.

TABLE III
TWENTY QUERIES CHOSEN FOR THE PERFORMANCE EVALUATION

No.	Terms	No.	Terms
1	Fun	11	CG
2	Actor	12	Ending
3	Story	13	Sentimental
4	Touching	14	3D
5	Acting	15	Laugh
6	Action	16	Recommend
7	Sorrow	17	Comedy
8	Tear	18	Happy
9	Director	19	Music
10	Twist-ended	20	War

The terms above were translated from Korean to English. Each query term was considered as a primary indicator of evaluating movie relevancy. For example, if someone searches for movies with the query term "Story," it is highly likely to indicate that he or she wants to find those movies that have good story flows.

C. Experiment Measure

To evaluate the performance of our system, we used the Normalized Discounted Cumulative Gain (NDCG) measure [26]. Unlike Precision@K or Recall@K, DCG is a rich measure, as it assigns more weight to highly ranked objects. It is computed as follows:

$$DCG(p)\begin{cases} G(1) & ,if\ p=1 \\ DCG(p-1)+G(p)/\log(p) & ,otherwise. \end{cases} \quad (14)$$

In this equation, p is a particular rank position and DCG(p) denotes the DCG value accumulated at a particular rank position p. The gain values, G(p), for the *Precision* and *Satisfaction*, were different. For the *Precision* measure, we set G(p) = 1 if there were user U's comments that contained the query term Q for a movie; otherwise, G(p) = 0. For the *Satisfaction* measure, we set G(p) as the rating score that user U gave to a movie. We then transformed its value to a normalized score between 0 (the worst possible DCG given the gain values) and 1 (the best possible DCG given the gain values, Ideal DCG) using (15).

$$\mathrm{NDCG(p)} = \frac{DCG(p)}{IDCG(p)} \quad (15)$$

D. Experiment Results

We evaluated the performance of our system with the top 5 movies from the search results obtained using 20 queries. We added up to 6 additional terms to the expanded query set. We created three different movie indexes by using *Desc* (only fetching movie description), *Desc+Comm* (fetching movie description and textual information in user comments), and *Desc+Comm /w R* (fetching movie description and textual information and ratings in user comments). In calculating the weight of rating score, we used two methods: *Term Frequency – Inversed Comment Frequency (TF-ICF)* and *Term Frequency Rating – Inversed Comment Rating (TFR-ICR)*. Q indicates an initial query set and Q+*n* means *n* terms are added to the initial query set.

TABLE IV presents the NDCG at 5 when the gain value equals the precision score. As the value gets closer to 1.0, relevancy of the top 5 movies is increased. Using description only to create movie index (*Desc*), it shows the lowest performance in overall because the number of terms in *Desc* is limited. Compared to the *Desc*, the advanced method of additionally including textual information in user comments (*Desc+Comm*) shows a significant improvement of performance. With description, textual information, and ratings in user comments (*Desc+Comm /w R*), it shows the best performance because it assigns more weight to positive terms in the sources.

In comparing the performance between *TF-ICF* and *TFR-ICF*, *TFR-ICF* that reflects user rating scores to term weighting performs better than TF-ICF. With Q+5, *TFR-ICF* performs 77.27% better than the initial query Q. Adding more than 5 terms does not improve the performance, because the search result does not change when adding more than 5 terms to the expanded query set.

TABLE IV
NDCG (GAIN VALUE=PRECISION) AT 5 FOR PROPOSED METHODS

	Query Expansion using TF-ICF			Query Expansion using *TFR-ICF*		
	Desc	*Desc+ Comm*	*Desc+ Comm /w R*	*Desc*	*Desc+ Comm*	*Desc+ Comm /w R*
Q	0.041	0.194	0.198	0.041	0.194	0.198
Q+1	0.052	0.209	0.273	0.051	0.268	0.282
Q+2	0.062	0.223	0.300	0.065	0.303	0.318
Q+3	0.072	0.233	0.316	0.073	0.318	0.334
Q+4	0.078	0.245	0.323	0.077	0.329	0.347
Q+5	0.088	0.252	0.330	0.086	0.336	**0.351**

TABLE V presents the NDCG at 5 when the gain value equals the rating score. As the value gets closer to 1.0, top 5 movies in the search results more tend to be the movies that the user likes because a high gain value implies that the user would give a high score to that movie. Similar to the experimental results by *Precision* measure, the results of *Satisfaction* shows the best performance with movie index by using movie descriptions, textual information and rating in user comments (*Desc+Comm /w R)* and query expansion by using *TFR-ICF*.

TABLE V
NDCG (GAIN VALUE=RATING SCORE) AT 5 FOR PROPOSED METHODS

	Query Expansion using TF-ICF			Query Expansion using *TFR-ICF*		
	Desc	***Desc+ Comm***	***Desc+ Comm /w R***	***Desc***	***Desc+ Comm***	***Desc+ Comm /w R***
Q	0.291	0.639	0.637	0.291	0.639	0.637
Q+1	0.322	0.643	0.639	0.331	0.647	0.659
Q+2	0.361	0.648	0.642	0.377	0.651	0.669
Q+3	0.398	0.652	0.645	0.416	0.655	0.674
Q+4	0.427	0.654	0.644	0.446	0.655	0.675
Q+5	0.456	0.657	0.646	0.475	0.658	**0.676**

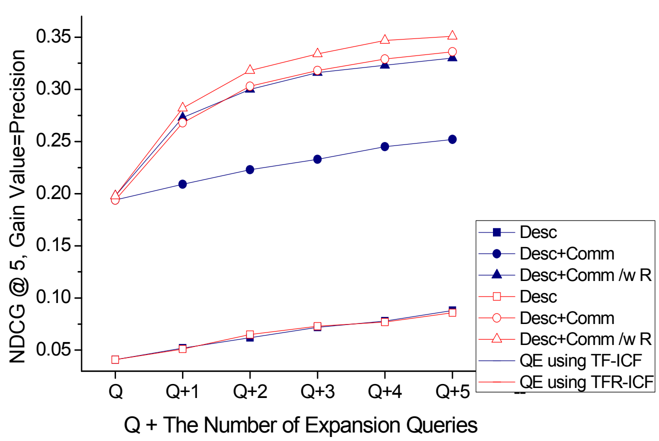

Fig. 6. NDCG at 5 with Gain Value=Precision for Compared methods

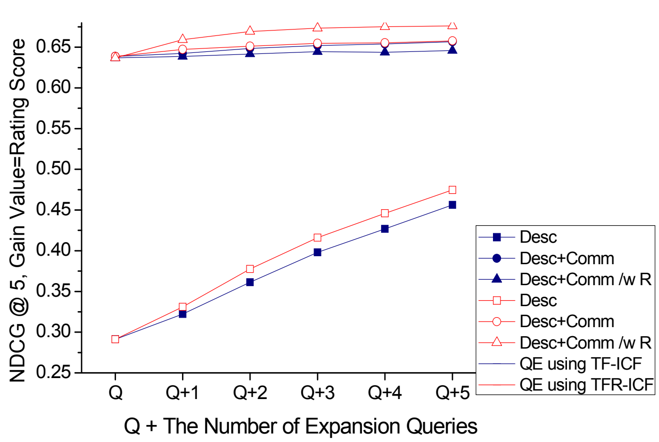

Fig. 7. NDCG at 5 with Gain Value=Rating Score for Compared methods

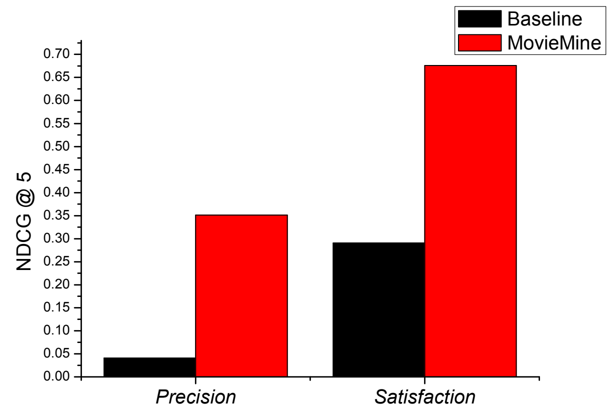

Fig. 8. Precision and Satisfaction NDCG at 5 compared between baseline and MovieMine

From Fig. 6 and Fig. 7, we can observe that overall performance increases with using 1) descriptions and user comments; 2) *TFR-ICF* as our term weighting scheme for query expansion; and 3) more number of query terms.

Fig. 8 summarizes a performance comparison between baseline and MovieMine (baseline: index - Desc, # of expansion query terms – zero, MovieMine: index – Desc+Comm/w R, # of expansion query terms – 5 terms using TFR-ICF weights). Our system shows a significant performance improvement compared to the baseline, which creates movie index only using description and searches movies with initial query Q, for the *Precision* and *Satisfaction* measures (*Precision* NDCG@5 - base: 0.041, MovieMine: 0.351; *Satisfaction* NDCG@5 - base: 0.291, MovieMine: 0.679).

VI. CONCLUSION

In this paper, we proposed a novel query expansion method that utilizes user comments in order to consider user's different preferences in finding movies. MovieMine, a new personalized search system built upon the proposed method creates a movie index by using the textual information and ratings in user comments left for each movie, in addition to movie description data. In our experiment, using an actual movie review dataset obtained from a large movie portal, we verified that the proposed system shows a significant performance improvement compared to the baseline condition. Further enhancing the proposed method by considering similar users in their movie tastes is a potentially fruitful direction for future research. However, it should be noted that the proposed approach is readily applicable to more versatile and personalized searching of multimedia contents.

REFERENCES

[1] S. Lee, D. Lee and S. Lee, "Personalized DTV program recommendation system under a cloud computing environment", IEEE Trans. Consumer Electron., vol. 56, no.2, pp. 1034-1042, May 2010.

[2] M. Bjelica, "Unobtrusive relevance feedback for personalized TV program guides", IEEE Trans. Consumer Electron., vol. 57, no. 2, pp. 658-663, May 2011.

[3] C. Shahabi and Y.C. Chen. "Web information personalization : Challenges and approaches," In Proceedings of 3rd Workshop on Databases in Networked Information Systems (DNIS), 2003

[4] B. Tan, X. Shen, and C. Zhai, "Mining long-term search history to improve search accuracy," In Proc. of ACM KDD, pp 718–723, 2006.
[5] T. Joachims, L. Granka, B. Pang, H. Hembrooke, and G. Gay, "Accurately interpreting clickthrough data as implicit feedback," In Proceedings of ACM SIGIR, pp 154–161, 2005.
[6] Z. Dou, R. Song, and J.-R. Wen. "A large-scale evaluation and analysis of personalized search strategies," In Proceedings of WWW, pp 581–590, ACM, 2007.
[7] D. Carmel, N. Zwerdling, I. Guy, S. Ofek-Koifman, N. Har'el, I. Ronen, E. Uziel, S. Yogev and S. Chernov, "Personalized social search based on the user's social network," In Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM'09), 2009.
[8] P. A. Chirita, C. S. Firan, and W. Nejdl, "Personalized query expansion for the web," In Proceedings of ACM SIGIR, pp. 7–14, 2007.
[9] X. Shen, B. Tan, and C. Zhai, "Implicit user modeling for personalized search," In Proceedings of ACM CIKM, pp 824–831, 2005.
[10] H. Kwon and K. Hong, "Personalized smart TV program recommender based on collaborative filtering and a novel similarity method", IEEE Trans. Consumer Electron., vol. 57, no. 3, pp. 1416-1423, August 2011.
[11] H. Shin, M. Lee and E. Y. Kim, "Personalized digital TV content recommendation with integration of user behavior profiling and multimodal content rating", IEEE Trans. Consumer Electron., vol. 55, no. 3, pp. 1417-1423, August 2009.
[12] R. Sotelo et al., "TV program recommendation for groups based on muldimensional TV-Anytime classifications", IEEE Trans. Consumer Electron., vol. 55, no. 1, pp. 248-256, Febrauary 2009.
[13] A. Martinez et al., "What's on TV tonight? An efficient and effective personalized recommender system of TV programs," IEEE Trans. Consumer Electron., vol. 55, no. 1, pp. 286-294, February 2009.
[14] G. Adomavicius, and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the art and possible extensions," IEEE Transactions on Knowledge and Data Engineering vol. 17, No. 3, 2005, pp 734-749.
[15] J.A. Konstan,, et al., "GroupLens: Applying collaborative filtering to usenet news," Communications of the ACM vol. 40, no. 3, 1997, pp 77-87.
[16] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: item-to-item collaborative filtering," IEEE Internet Computing, vol. 7, no. 1, pp. 76-69, Jan./Feb. 2003.
[17] B.N. Miller, et al., "MovieLens unplugged: experiences with an occasionally connected recommender systems," In Proceedings of. IUI, 2003
[18] S. Debnath, et al., "Feature weighting in content based recommendation system using social network analysis," In Proceedings of WWW, 2008
[19] R.J. Mooney and L. Roy, "Content-based book recommending using learning for text categorization," In Proceedings of ACM SIGIR, 1999.
[20] M. Tsagkias, et al., "News comments: Exploring, modeling, and online prediction," In Proceedings of ECIR, 2010.
[21] T. Yano, and N.A. Smith,, "What's worthy of comment? content and comment volume in political blogs," In Proceedings of AAAI, 2010.
[22] J. Chen et al., "Short and tweet: Experiments on recommending content from information streams," In Proceedings of CHI, 2010.
[23] S. Faridani, et al., "Opinion space: A scalable tool for browsing online Comments," In Proceedings of CHI, 2010.
[24] Wai Gen Yee, Andrew Yates, Shizhu Liu, and Ophir Frieder, "Are web user comments userful for search?", In Proceedings of 7th Workshop on Large-Scale Distrubutued Systems for Information Retrieval (LSDS-IR), pp. 63-70, 2009.
[25] M. Ko et al., "MovieCommenter: Aspect-based collaborative filtering by utilizing user comments", In Proceedings of CollaborateCom, pp. 362-371, 2011.
[26] K. J¨arvelin and J. Keklinen. "Ir evaluation methods for retrieving highly relevant documents". In Proceedings of the 23th Intl. ACM SIGIR Conf. on Research and development in information retrieval, 2000.

BIOGRAPHIES

Hyung W. Kim received his BS in Computer Science and Electronic Engineering from Handong Global University (HGU) in 2010. He receivee his MS in knowledge service engineering from Korea Advanced Institute of Science an d Technology (KAIST) in 2012. He joined ETRI, Daejeon , Rep. of Korea in 2012. He is currently working as a researcher in smart search and recommendation system de velopment for smart TV project His research interests include information retr ieval for smart TV contents, data mining and text-mining.

Keejun Han received his BS in information technology from Australian National University (ANU) in 2010. He will receive his MS in knowledge service engineering from Korea Advanced Institute of Science and Technology (KAIST) in 2013. His research interests include information retrieval for semi-structure data, difficulty analysis for learning materials, and data mining.

Mun Y. Yi is an Associate Professor at KAIST (Korea Advanced Institute of Science and Technology). He received his PhD in information systems from University of Maryland, College Park. His current research focuses on knowledge engineering, semantic web, IT training, and human–computer interaction. He currently serves as an Associate Editor of International Journal of Human-Computer Studies and a Senior Editor of AIS Transactions on Human-Computer Interaction.

Joonmyun Cho received his BS, MS and PhD in mechanical Engineering from Korea Advanced Institute of Science and Technology (KAIST) in 1993, 1995 and 2006, respectively. He joined ETRI, Daejeon, Rep. of Korea in 2007 and was involved with the URC (Ubiquitous Robotic Companion) project until 2011. He is currently working as a project leader in smart search and recommendation system development for smart TV project. His research interests include ontology-based knowledge base system and intelligent agent system.

Jin-Woo Hong received the BS and MS degree in electronic engineering from Kwangwoon University, Seoul, Korea, in 1982 and 1984, respectively. He also received the Ph.D degree in computing engineering from the same university in 1993. Since 1984, he has been with ETRI, Daejeon, Korea, as a principal researcher, where he is currently a managing director of the next generation Smart TV Research Department. From 1998 to 1999, he conducted research at Fraunhofer Institute in Erlangen, Germany, as a visiting researcher. His research interests include multimedia framework technology, broadcasting media and service, personal broadcasting, realistic media, and platform technology of Smart TV.