

# Content-Based News Recommendation: Comparison of Time-Based System and Keyphrase-Based System

Servet Tasci and Ilyas Cicekli<sup>(✉)</sup>

Department of Computer Engineering, Hacettepe University, Ankara, Turkey  
{servettasci, ilyas}@cs.hacettepe.edu.tr

**Abstract.** As internet resources are increasing at an unprecedented speed, users are tired of searching important ones among them. So, what users need and where they can find them are getting more important. Users require a personalized support in sifting through large amounts of available information according to their interests and recommendation systems try to answer this need. In this context, it is crucial to offer user friendly tools that facilitate faster and more accurate access to articles in digital newspapers. In this paper, a content-based news recommendation system for news domain is presented and contents of news articles are represented by words appearing in news articles or their keyphrases. News articles are recommended according to user dynamic and static profiles. User dynamic profiles reflect user past interests and recent interests play much bigger roles in the selection of recommendations. Our recommendation system is a complete content-based recommendation system together with categorization, summarization and news collection modules.

**Keywords:** Recommendation systems · Content-based filtering · Keyphrase-based recommendation · Text summarization · Text classification

## 1 Introduction

The abundance of information with their dynamic contents is available on the web [2]. As there are many documents and resources, it is difficult to find what we need and where we can find them. Besides, websites and web pages are doubled every year. People are overwhelmed by the large amount of information on the web, therefore information overload problem is worsening at an unprecedented speed. As a consequence, user modelling and personalized information access are becoming crucial. Users require a personalized support in sifting through large amounts of available information, according to their interests.

Many search engines have emerged to alleviate this problem. These search engines use very large databases to index websites, and reduce information overload problem by allowing users a centralized search. But still users are tired of looking which documents are useful. Research results have shown that users often give up their search in the first try, examining no more than ten documents. This shows us that users want from a web site recommendations rather than search results.

When visiting a news website, users are looking for new information that they have not known before and that information will be an interest for them. The kind of information that will be an interest for a user depends on that user's past activities. User profiles are created in order to recommend related items to users. Since user profiles are inferred from past user activities, it is important to know how effective they would be to predict user behavior.

The problem of recommending items has been studied extensively, and two main paradigms have emerged. Content-based recommendation [15, 19, 20] try to recommend items similar to those that a given user has liked in the past, whereas collaborative recommendation paradigm identify users preferences are similar to those of the given user and recommend items that they have liked [4]. These two approaches have their own strengths and weaknesses. Hybrid recommender systems [7] combine these approaches. Li and Li [21] use hypergraph learning for news recommendation.

In this paper, we describe a content-based recommender system for news domain. A content-based recommender system consists of mainly three steps: *Content analyzer*, *profile learner* and *filtering component*. *Content-analyzer* extracts features from unstructured text to produce a structured item representation. *Profile learner* is the heart of a recommendation system and it uses user feedbacks which are usually inferred as explicit or implicit feedbacks. Explicit feedbacks are results of explicit evaluations of items by users and they indicate whether items are interesting or not to users. Implicit feedbacks try to extract user interests from user click behaviors. In our recommendation system, we use explicit feedbacks and five-level ratings for user feedbacks in evaluation. *Filtering component* implements some strategies to rank interesting items with respect to user profiles and generates recommendations.

News domains have some special characteristics [30] and recommendation systems have to deal with these characteristics. Since a news domain contains a large volume of information and documents are in unstructured format, recommendation systems require more computation power. News items typically have short self-lives and recommendations should be up-to-date. Since most news articles describe specific events, recommended articles should be related with interested events. The user interest can easily shift from one event to another event, and selection and ranking of news articles to be recommended should be done according to this fact.

In this paper, we present a content-based recommendation system for news domain named as *HaberAnalizi*. The presented recommender system uses dynamic time-based user profiles that are automatically updated. Recently read news articles play more important roles in the creation of user profiles. The recommender system described here is a complete web site that collects news articles from various online Turkish news websites and recommends articles to its users. Everyday approximately 1500 news articles are collected. Then, these news articles are classified into eight categories: *Magazine*, *Economics*, *Politics*, *Culture-Art*, *Techno-Science*, *Health*, *Sports and Education*. After that, news articles are recommended to users according to their dynamic and static profiles. An accurate profile of a user is critical for the success of news recommendation system. So how we construct user profiles plays an important role in the success of the system.

The rest of the paper is organized as follows. Section 2 presents the related work in news recommendation domain. Section 3 explains the details of our news recommendation system. Section 4 presents the performance evaluation of the system and Sect. 5 concludes the paper and gives some possible future work.

## 2 Related Work

News recommendation has been studied for years and some methods have been proposed and evaluated. News-dude [6] is a news recommendation system uses TF-IDF values combining K-Nearest Neighbor (KNN) algorithm supporting a series of feedback options such as “interesting”, “not interesting”, etc. Most of the collaborative filtering systems use KNN for recommendation [8]. Likewise in our news recommendation system, we use five categories to assess recommendations.

Liang and Lai [22] propose a time-based approach to build user profiles from browsing. They calculated the user’s elapsed time while reading an article and also they took into account the article’s length and positions of the words in the document. In contrast to our recommendation system, they do not use news classification. They also calculate a time range for document reading and this is not proper because reader’s reading speed can vary. They can have the *first-rater problem* since no static profiling is used.

Tan and Teo [31] present a personalized news recommendation system named PIN. In this system, users choose a list of keywords to be used in the creation of their profiles and the system recommends news according to these profiles. Since user profiles are not updated, it is hard to make it dynamic, and user interest changes cannot be reflected in profiles.

Liu et al. [24] present that user current interests are critical for the success of a recommendation system. Their system is also a content-based recommendation system and they also do not use collaborative filtering. In their system, they construct user profiles automatically on any interaction that user makes with the system and this puts extra burden on their system. We update user profiles daily not on each user interaction. Their system infers user interests based on user click behavior on the website. No rating or negative votes are used for privacy reasons. In addition to the click distribution of individual users, they calculated the click distributions of the general public. When we compare to this method with our approach, we give more importance to ratings of last seven day clicks of individual user in the creation of user dynamic profiles and these profiles are updated daily. Public trend is not important for our system. Likewise their system and our system do not take into account of time spent on the page.

Gong [14] uses collaborative filtering based on user clustering and item clustering. First of all, they divide users into clusters using a clustering algorithm based on some similarity threshold and a user-item matrix is created. Then item clusters are created based on similarity measures. In order to solve sparsity problem, they use expensive Pearson similarity measure [25] for new items that user has not rated yet. News items are recommended using KNN algorithm and the created user-item matrix. In our approach we categorize news and for each category we use news items that user has rated last week or user static profiles if the user has not rated any news article.

Saranya and Sadhasivam [30] collect categorized news from news agencies and dump them into a database. They use only one news agency articles for categorization of articles and this means that an article can be categorized differently depending on which corpus is used. In our system we use all news articles for categorization. Likewise, they also use both static and dynamic profiles. Dynamic user profiles are constructed during every interactive search session. They use user clicks to update dynamic profiles every time user reads a news article. This can be costly to implement and users do not want to wait for new news items on online systems.

Li et al. [23] especially try to address issues as news selection, news representation, news processing, and user profiling. Their system consists of three major components which are news article clustering, user profile construction and news item recommendation. News items are clustered using hierarchical clustering. In order to build user profiles, they use three dimensions: news content, similar access patterns and preferred news entities. User profiles and news articles are represented using triplets in the form of  $\langle \textit{Topic vector}, \textit{patterns}, \textit{entities} \rangle$ . For recommendation, similarity between news item and user profile is computed. They assessed their system in 15 days. First 5 days 50 users read news and then successive 10 days they recommended news items to the volunteers. In our system we collected the data approximately for about 5 months.

### 3 Content-Based News Recommendation System

In this section we explain the details of our time-based news recommendation system. Our recommendation system is a complete system which is designed as a website which collects news articles from online Turkish newspapers and recommends to its users. Although our recommendation system domain is online Turkish news domain, it can be adaptable to other news domains because the only specific resource for Turkish language is a Turkish stemmer. Other parts of the system are not language dependent.

The major parts of our content-based news recommendation system are as follows:

- Getting News from News Sites
- News Classification
- Summarization of News
- Profile Construction
- Time-Based News Recommendation System
- Keyphrase-Based News Recommendation System

First of all, obtaining news articles from substantial news sites is explained, and then how categories of news items are found are explained. Our recommendation system can also produce summaries of news articles using a text summarization system [26]. Later we explain the creation of static and dynamic user profiles. Finally, we explain our news recommendation approach at the end of this section.

#### 3.1 Getting News from News Sites

Obtaining news contents from online news sites is not a trivial job. Because news sites publish news articles as parts of HTML pages and we have to obtain contents of news articles. An HTML page contains advertorial part, content part, links part

and other parts, etc. We need just news content of the page. In order get the content of a news article, we use XML Path Language (*XPath*) expressions. Each news site has its own characteristics and contents of news articles are extracted using news sites characteristics and XPath expressions. After the content part of a news article is extracted, it is feed into the rest of the recommendation system.

Approximately from twenty online news sites we obtain approximately 1500 news articles every day. Lengths of news articles vary from a couple of sentences to longer documents. Contents of news articles are extracted and they are prepared for the rest of the system in order to be recommended to users.

### 3.2 News Classification

Most of news sites do not classify news articles. Even if they classify, classifications of news articles are so different at each news site. For example while a news item’s category is politics in a web site, the other web site could categorize it as economics. So we have to classify them for standardization and obtaining better recommendations to users. After news sites are analyzed, we have determined eight categories for classification: Magazine, Economy, Politics, Culture-Art, Techno-Science, Health, Sports and Education.

Most of classification algorithms are based on the single term analyses of the text, e.g. vector space model [29]. On this context every document can be thought as a vector. Each word in the document can be thought as an attribute of the document.

Some main online news sites categorize daily news, and we used 40,000 articles with known categories from these sites as our corpus for categorization. Using these categorized news articles, we determined subtle words computing their frequencies. A news article and each category can be shown as a vector of pairs of words and their normalized frequencies. The normalized frequency of a word is computed by dividing its frequency in document with the total number of words in that document. After that, we have tested different similarity measures for classification with our corpus of 40,000 news articles with known categories. The details of our news article corpus are given in Fig. 1. The classification methods that are tested with our corpus are Support Vector Machines

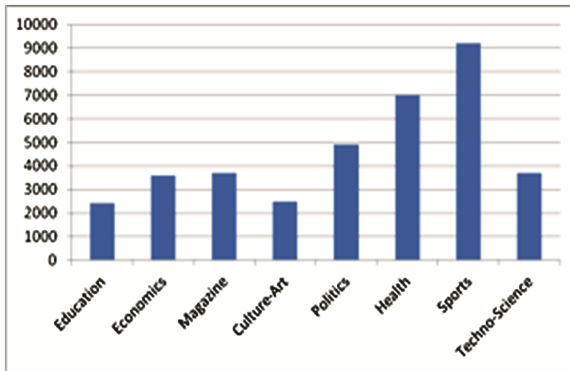


Fig. 1. News article corpus for classification.

(SVM), Euclidean Similarity, Cosine Similarity and Jaccard Similarity. Precision results of these methods on our corpus are given in Fig. 2 and it can be seen that Jaccard Method gives better results than other methods for our corpus. So we have decided to use Jaccard method for the classification of news articles in our system. We tried to eliminate the over-fitting problem using as much as many words as feature values.

	Jaccard	Cosine	Euclidean	SVM
Education	0,65	0,52	0,40	0,60
Economy	0,75	0,47	0,35	0,70
Magazine	0,65	0,50	0,42	0,60
Culture-Art	0,70	0,45	0,40	0,70
Politics	0,67	0,40	0,30	0,72
Health	0,82	0,45	0,60	0,75
Sports	0,80	0,70	0,65	0,70
Techno-Science	0,75	0,55	0,35	0,70
TOTAL	0,75	0,53	0,48	0,70

**Fig. 2.** Accuracy results of classification algorithms.

For classification we use stop word elimination and stemming. We remove approximately 150 stop words from our news articles. We also stem words using a Turkish stemmer named Zemberek [1].

### 3.3 Summarization of News Articles

News articles generally consist of few topics. These topics form main contents of news articles and some of these topics are explained deeply. A good summary of an article should cover major topics in that article. Summaries can have different forms [16] and they are commonly obtained by sentence extraction or abstraction methods [17]. Although abstraction methods can provide sophisticated summaries and they are hard to implement. On the other hand, sentence extraction approaches are easy to adapt and they can still produce useful summaries.

Since users want to read more news in short time, we use a summarizer in our recommender system to accomplish this need. One of the goals of our system is to provide users opportunity to reduce the information overload likewise multi-document summarization system [28]. We adapted a sentence extraction based summarization system [26] in order to generate summaries of news articles. We generate summaries whose lengths 30 % of original news articles using this summarization system.

### 3.4 Profile Construction

A personalized recommendation system needs user characteristics to recommend items. For that reason, the construction of user profiles is a crucial part of a recommendation system. In order to construct individual user profiles, the required information can be

collected *explicitly* through direct user intervention, or *implicitly* through agents that monitor user activities. Profiles that can be modified or augmented are considered as *dynamic*, in contrast to *static* profiles that maintain the same information over time. Most recommender systems use both static and dynamic profiles. Static profiles are used to solve cold-start problem in recommender systems. If keywords are given for categories in a static profile, recommendations are based on these keywords and this solves some parts of cold-start problem. If no keywords are given in a static profile, we recommend most recommended news to that user in order to solve cold-start problem. As users read more news articles, their dynamic profiles improve. In our system, we use both dynamic and static profiles for recommendation. A comprehensive survey of various user profile construction techniques is provided in [13].

A static or dynamic profile consists of a set of probabilities for categories and a list of interest words for each category. Summation of all probability values in a profile is equal to 1 and the probability value of a category indicates the percentage of user's interest in that category. The probability value of a category determines the amount of news articles that will be recommended to that user. The interest words of a category are a set of pairs of words together with their normalized weights. The weight of an interest word is the normalized frequency of that word and reflects the amount of user interest to articles containing that word. The interest words of a category are used to select news articles in that category for recommendation to the user. A sample profile will be as follows:

```
Profile = {
  Magazine      : 0.20, MagazineWords,
  Economy       : 0.15, EconomyWords,
  Politics      : 0.25, PoliticsWords,
  Culture-Art   : 0, {},
  Techno-Science : 0.12, ScienceWords,
  Health        : 0.08, HealthWords,
  Sports        : 0.16, SportsWords,
  Education     : 0.04, EducationWords
}
```

In our recommender system, users fill profile forms and give interest words for categories. This information is used in the creation of a static profile. From given information, a static profile is created. The probability value of each category in a static profile is the same and this means that same amount of articles is recommended for each category if a static profile is used for recommendation. The interest words of a category are the words given for that category when the user fills a profile form. If the user does not give any interest words for a category, no articles in that category are recommended to that user when that static profile is used for recommendation.

Dynamic profiles are constructed according to news articles read by users. For each user, two dynamic profiles are created if that user reads news articles regularly. The first dynamic profile is the last week dynamic profile and it is constructed if the user has read news article during last seven days. The second dynamic profile is the last month dynamic profile and it is created according to news articles read by that user during last 30 days.

If a user has rated news items in last seven days the last week dynamic profile is created using the news items that are rated as *interesting* by the user in the last seven days, and news article recommendations are done according to this last week dynamic profile. If a last week dynamic profile is not constructed for a user, the last month dynamic profile of that user is used for recommendation if that profile is available. If no news articles are read by a user in the last month, no dynamic profiles will be available for that user and the static profile for that user is used for recommendation.

Dynamic profiles of a user are updated once every day. Recently read news articles are more likely to indicate user's current interests. For that reason, news articles read in the last days have more weights in profile constructions. Thus, the effect values of words appearing in recent day articles are multiplied with bigger weights than words in older day articles. Weights used in the construction of dynamic profiles decrease gradually towards past. These weight values are determined using news reading trends of users.

Category probability values are updated according to the articles read in that category. Before category interest words are updated with an *interested* article, stop words are eliminated from article words and article words are stemmed. Interest words in the profile are updated with these stemmed words. Frequency values of article words are used when updating weights of interest words in dynamic profiles.

### 3.5 Content-Based News Recommendation System

A typical recommendation system consists of an object set and a user set [33]. The aim of a recommendation system is to meet the requirements of users using this object set. In our recommendation systems, the object set is the set of daily news articles. Furthermore, a recommendation system should be able to suggest objects to users, which users would not discover for themselves [34] because of huge amount of objects. Our recommendation system tries to suggest news articles to users from a huge set of daily news articles.

Our recommendation system collects daily news articles from online newspaper sites. Everyday approximately 1500 news articles are collected from these newspaper sites. Our recommendation system recommends  $N$  daily articles to a user depending on that user's profiles where  $N$  is set to 50 in our evaluation sessions. Categories of daily collected news articles are computed by the categorization module of our system.

In order to recommend news articles to a user, first a profile of the user is determined for recommendation according to the user's history. If the user read and rated news articles in last seven days, the last week dynamic profile is used for recommendation. If the user has not read any article in the last week, the last month dynamic profile is used. If no dynamic profile available for the user, the static profile is used.

After a profile of a user is selected, news recommendations are done with respect to that profile. For each category  $C_i$ , the number ( $N_{C_i}$ ) of news articles that will be recommended for that category is determined by multiplying the probability value of that category with the total number ( $N$ ) of articles to be recommended. Then articles in category  $C_i$  of daily news articles are compared with the category interest words in the profile in order to measure similarities of articles with the category interest words. The most  $N_{C_i}$  similar articles are recommended to the user for the category.



The similarity of a news article with interest words is measured using the cosine similarity measure. Stop words are eliminated from article words and article words are stemmed before they are compared with interest words. Title words of an article have more weights in the computation of the similarity measure than regular article words.

### 3.6 News Recommendation System with Keyphrases

Users tend to read news articles that have similar contents. News article suggestions can be done using this idea. The usage of dynamic profiles and words appearing in previously read news articles in our content-based recommendation system tries to capture this fact indirectly.

Keyphrases of an article describes the semantic content of that article and what that article talks about. Since keyphrases of articles describe contents of those articles, content similarity of articles can be measured by comparing their keyphrases. Using this idea, we created a version of our recommendation system which employs keyphrases in recommendation.

A keyphrase can be a single word or a sequence of words which semantically describes the content of a text. In fact, a set of keyphrases for a text can be seen as a summary of that text. In addition to summarization, keyphrases can be used for indexing documents in order to be used in precise searches of those documents [3]. Unfortunately, most of digital documents are not associated with keyphrases and most of news articles do not have keyphrases. So, keyphrases of texts should be extracted using keyphrase extraction algorithms [10, 11]. Many keyphrase algorithms [18, 27] for Turkish texts are given in literature, and we use a Turkish keyphrase extraction algorithm [27] which is based on ideas in KEA keyphrase extraction algorithm [32] in order to extract keyphrases of Turkish news articles.

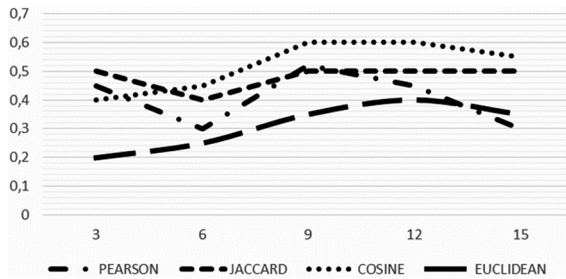
In order to create a good recommendation system based on keyphrases, some major steps should be done. A version of our recommendation system which is based on keyphrases employs following major steps.

- Extracting keyphrases of news articles.
- Finding similar news articles using keyphrases and similarity metrics.
- Recommending news articles to users using similarity metrics between daily news articles and news articles read before by users.

Since our recommendation system is designed for Turkish news domain, we use a Turkish keyphrase extraction algorithm [27] to extract keyphrases of Turkish news articles. This extraction algorithm returns a list of keyphrases which are sorted according to their importance for a given Turkish text. The top  $N$  keyphrases of a news article are used to represent that article.

We use keyphrases of news articles to measure similarities among them. In order to decide how many keyphrases should be used to represent news articles and which similarity metric gives the best result to measure similarity among news articles, we created a small test set. The test set contains two sets of articles and similarity algorithms find similar news articles in the second set for each article in the first set. Of course, articles in the second set are marked whether similar to any article in the first set. A set of different

similarity metrics like *Cosine*, *Jaccard*, *Euclidean* and *Pearson Coefficient* are tried to determine similar news articles using their keyphrases. Figure 3 shows accuracy results of similarity algorithms when different numbers of keyphrases are used for articles. Cosine similarity has better results than other similarity metrics according to Fig. 3. According to the results in Fig. 3, when ten keyphrases are used for news articles, the best scores are obtained experimentally. As a result of experimental evaluation, cosine similarity metric and top ten keyphrases for news articles are decided to be used in our system



**Fig. 3.** Accuracy results of similarity algorithms for news article sets using different number of keyphrases.

In order to recommend daily news articles to users, similarities of daily articles with articles recently read by users are determined by the selected similarity algorithm using article keyphrases. Top similar daily articles are recommended to users. Comparison of our main recommendation system with our keyphrase based recommendation system is given in the evaluation section.

## 4 Evaluation

Evaluation of a recommendation system is important for determining the efficiency of the system. There are many evaluation methods in order to evaluate a recommendation system such as precision and recall values, MAE (mean absolute error) value, RMSE (root mean squared error) value and ROC (receiver operating characteristic) analysis. For example, Cinematch system in the Netflix prize competition evaluates itself using RMSE [5]. For our recommendation system, we use precision, recall, f-measure and accuracy values for evaluation, and we also give the ROC analysis of our results.

In order to verify the efficiency of our proposed recommendation system, we tested it for five months on 30 users with approximately 1500 news items per day. Daily 30–50 news articles are recommended to each user. If a user has a dynamic profile, always 50 articles are recommended. After reading recommended news, users are asked to evaluate recommended news by clicking one of five rankings: *not interested*, *less interested*, *no comment*, *interested*, *mostly interested*. Users are assumed that they are interested with an article if they select one of three choices (*no comment*, *interested*, *mostly interested*) since *no-comment* choice in our evaluations reflects some interest in articles.

	Not Interested	Less Interested	No Comment	Interested	Mostly Interested	Number of Recommended Articles	Accuracy Results
Education	210	267	380	193	65	1115	0.57
Economy	343	442	613	435	145	1978	0.60
Magazine	638	815	1103	1006	500	4062	0.64
Culture-Art	345	267	225	295	125	1257	0.51
Politics	575	703	585	897	215	2975	0.57
Health	223	137	145	115	65	685	0.47
Sports	975	1370	1670	1572	820	6407	0.63
Techno-Science	336	610	245	195	135	1521	0.38
TOTAL	3645	4611	4966	4708	2070	20000	0.59

**Fig. 4.** Evaluation results for rated articles.

Then we randomly selected 20 days and 20 users who rated all of 50 recommended articles daily. Thus we collected 20,000 rated articles by users. All recommendations are made according their dynamic profiles. For these 20,000 rated articles, ratings and distributions over categories are given in Fig. 4. The average accuracy result for all categories is 0.59 and Magazine category has the highest accuracy result. According to Fig. 4, some interesting results can be inferred for each category. Users mostly interested news items in Sports, Magazine and Politics categories.

For any recommendation system it is hard to determine the number of items to be recommended to users. If the number of recommended items is high, the recommended items may contain too many unrelated items and users will be unhappy with this result. On the other hand, if fewer items are recommended, users may not be satisfied from this

Number of Recommended Articles	Precision	Recall	F-Measure	Accuracy
10	0.60	0.21	0.31	0.46
20	0.70	0.48	0.57	0.58
30	0.73	0.76	0.75	0.70
40	0.63	0.86	0.72	0.62
50	0.58	1.00	0.73	0.58

**Fig. 5.** Evaluation of news recommendation system and the computation of optimum news count.

result either. For this reason, we decided to compute optimum news count for recommendation in our system.

In order to compute optimum news count, we made a detailed analysis on the results given in Fig. 5. According to the accuracy result (0.59) in Fig. 5, users approximately found 29 articles to be *interesting* out of 50 recommended articles on the average. Recommended 50 articles were sorted according their similarity values and they were presented users in the decreasing order. We checked the number of articles found to be *interesting* in top 10, 20, 30, 40 and 50 recommended articles, and *Precision, Recall and F-Measure* results are presented in Fig. 5. According the results in Fig. 5, best precision and f-measure results are obtained in top 30 recommended articles and this indicates that optimum news count for our system is around 30. This means that users approximately find 22 articles interesting out of 30 recommended articles. Since the precision results for top 10 and 20 are not high enough, our similarity measure can be improved in order to move *interesting* articles into higher ranks.

As seen in the ROC curve graph representing *recall* against *fallout* in Fig. 6, recall value increasing rapidly until news count value reaches 30 then it increases slowly until value reaches 50. This means that 30 is the optimum news count for our system. ROC curves reflect precision/recall optimization for systems [9, 12].

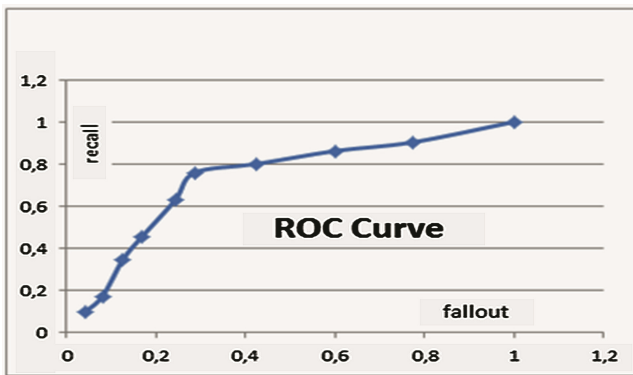


Fig. 6. ROC curve for optimum news count.

#### 4.1 Evaluation of Recommendation System with Keyphrases

In order to compare the results of our main recommendation system with the results of the recommendation system based on keyphrases, 50 articles that are recommended to users during the evaluation of main recommendation system and evaluated by users are re-sorted using the recommendation system with keyphrases. With this evaluation process, our main objective is to see whether keyphrase based recommendation can move interested articles upward direction in recommendation lists or not. The results of this evaluation process are given in Fig. 7. When the results in Figs. 5 and 7 are compared, it can be seen that more interested articles are moved into top 10 and 20 recommended article list. According precision results in Figs. 5 and 7, 6 of top 10 recommended articles

are found as interesting articles by users, 8 of top 10 recommended articles could have been found as interesting if keyphrase based system is used. According to the results, top 30 and 40 article lists are not improved by keyphrase based recommendation system. Although this evaluation process for keyphrase based recommendation system cannot be seen as a complete evaluation process, it reflects that the usage of keyphrases can improve the performance of the recommendation system.

Number of Recommended Articles	Precision	Recall	F-Measure	Accuracy
10	0.80	0.28	0.41	0.54
20	0.75	0.52	0.61	0.62
30	0.73	0.76	0.75	0.70
40	0.63	0.86	0.72	0.62
50	0.58	1.00	0.73	0.58

**Fig. 7.** Evaluation results of keyphrase based recommendation system.

## 5 Conclusion

Our news recommendation system is a time-based recommendation system and selects items for recommendation with respect user past interests. User past interests are represented by dynamic profiles and dynamic profiles reflect user interests for categories and user interest words for categories. Articles for recommendation are selected depending on their similarities with category interest words.

Our recommendation system is a complete recommendation system together with a news categorization module, a news summarization module and collection module for collecting news articles from online news sites. Users can read recommended daily news articles or they can browse all daily news in each category.

Users can rate news articles by explicitly giving a score between 1 and 5. User interest on article can be determined by an explicit evaluation by user same as in our system or user clicks together with amount of time spent to read that article. We selected the first approach because of its simplicity and clear rating mechanism. In bigger systems, user may not want to explicitly evaluate recommended articles, so the second approach may more appropriate for collection user interests.

The performance of a recommendation system can be affected by the performance of similarity measurement between articles and user profiles. Different similarity methods can be used in order to improve performance. Keyphrases are important for news recommendation systems and keyphrases can be used to measure similarity between documents. Our keyphrase based recommendation system improved performance of our main recommendation system.

Our news recommendation system focuses on selecting most interesting news articles that are similar news articles read by users before. However, readers can be burdened by more or less identical ones among returned news articles. Detection of news articles with identical contents can be useful to assist users in further reading and helping readers skip similar articles.

**Acknowledgements.** This paper is a revised and extended version of our KDIR-2014 paper whose title is “A Media Tracking and News Recommendation System”.

## References

1. Akin, A.A., Akin, M.D.: Zemberek, an open source NLP framework for Turkic languages (2007). <http://zemberek.googlecode.com>
2. Asanov, D.: Algorithms and Methods in Recommender Systems. Berlin Institute of Technology (2011). <http://www.snet.tuberlin>
3. Aygul, I., Cicekli, N., Ilyas Cicekli, I.: Searching documents with semantically related keyphrases. In: The Sixth International Conference on Advances in Semantic Processing (SEMAPRO 2012), Barcelona, Spain (2012)
4. Balabanovic, M., Shoham, Y.: Fab: content-based collaborative recommendation. *Commun. ACM* **40**(3), 66–72 (1997)
5. Bennett, J., Lanning, S.: The netflix prize. In: *KKDD Cup and Workshop* (2007)
6. Billsus, D., Pazzani, M.: A hybrid user model for news story classification. In: *The Seventh International Conference on User Modeling*, Banff, Canada, pp. 99–108 (1999)
7. Cantador, I., Bellogín, A., Castells, P.: Ontology-based personalised and context-aware recommendations of news items. In: *The 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology* (2008)
8. O’Conner, M., Herlocker, J.: Clustering items for collaborative filtering. In: *The ACM SIGIR Workshop on Recommender Systems*, Berkeley, CA (1999)
9. Davis, J., Goadrich, M.: The relationship between precision recall and ROC curves. In: *The 23rd International Conference on Machine Learning (ICML)* (2006)
10. Ercan, G., Cicekli, I.: Keyphrase extraction through query performance prediction. *J. Inf. Sci.* **38**(5), 476–488 (2012)
11. Ercan, G., Cicekli, I.: Using lexical chains for keyword extraction. *Inf. Process. Manag.* **43**(6), 1705–1714 (2007)
12. Fisher, M.J., Fieldsend, J.E., Everson R.M.: Precision and recall optimisation for information access tasks. In: *First Workshop on ROC Analysis in AI, European Conference on Artificial Intelligence (ECAI 2004)*, Valencia, Spain (2004)
13. Gauch, S., Speretta, M., Chandramouli, A., Micarelli, A.: User profiles for personalized information access. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) *Adaptive Web 2007*. LNCS, vol. 4321, pp. 54–89. Springer, Heidelberg (2007)
14. Gong, S.: A collaborative filtering recommendation algorithm based on user clustering and item clustering. *J. Softw.* **5**(7), 745–752 (2010)
15. Goossen, F., IJntema, W., Frasinca, F., Hogenboom, F., Kaymak, U.: News personalization using the CF-IDF semantic recommender. In: *The International Conference on Web Intelligence, Mining and Semantics (WIMS)* (2011)
16. Hahn, U., Mani, I.: The challenges of automatic summarization. *Computer* **33**, 29–36 (2000)

17. Hovy, E., Lin, C.-Y.: Automated text summarization in SUMMARIST. In: Mani, I., Maybury, M.T. (eds.) *Advances in Automatic Text Summarization*, pp. 81–94. The MIT Press, Cambridge (1999)
18. Kalaycılar, F., Cicekli, I.: TurKeyX: Turkish keyphrase extractor. In: *The 23rd International Symposium on Computer and Information Sciences (ISCIS 2008)*, Istanbul, Turkey (2008)
19. Kompan, M., Bieliková, M.: Content-based news recommendation. In: Buccafurri, F., Semeraro, G. (eds.) *EC-Web 2010. LNBIP*, vol. 61, pp. 61–72. Springer, Heidelberg (2010)
20. Li, L., Chu, W., Langford, J., Schapire, R.E.: Contextual-bandit approach to personalized news article recommendation. In: *The 19th International Conference on World Wide Web*, pp. 661–670 (2010)
21. Li, L., Li, T.: News recommendation via hypergraph learning: encapsulation of user behavior and news content. In: *The Sixth ACM International Conference on Web Search and Data Mining*, pp. 305–314 (2013)
22. Liang, T.-P., Lai, H.-J.: Discovering user interests from web browser behavior: an application to internet news services. In: *The 35th Annual Hawai'i International Conference on Systems Sciences*. IEEE Computer Society Press (2002)
23. Li, L., Wang, D., Li, T., Knox, D., Padmanabhan, B.: SCENE: a scalable two-stage personalized news recommendation system. In: *The 34th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Beijing, China, pp. 124–134 (2011)
24. Liu, J., Dolan, P., Pedersen, E.R.: Personalized news recommendation based on click behavior. In: *The 14th International Conference on Intelligent User Interfaces*, Hong Kong, China (2010)
25. McLaughlin, M.R., Herlocker, J.L.: A collaborative filtering algorithm and evaluation metric that accurately model the user experience. In: *The 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Sheffield, United Kingdom (2004)
26. Ozsoy, M.G., Cicekli, I., Alpaslan, F.N.: Text summarization using latent semantic analysis. *J. Inf. Sci.* **37**(4), 405–417 (2011)
27. Pala, N., Cicekli, I.: Turkish keyphrase extraction using KEA. In: *The 22nd International Symposium on Computer and Information Sciences (ISCIS 2007)*, Ankara, Turkey (2007)
28. Radev, D.R., Fan, W., Zhang, Z.: Web in essence: a personalized web-based multi-document summarization and recommendation system. In: *The NAACL-01*, pp. 79–88 (2001)
29. Salton, G., McGill, M.J.: *Introduction to Modern Information Retrieval*. McGraw-Hill Inc., New York (1986)
30. Saranya, K.G., Sadhasivam, G.S.: A personalized online news recommendation system. *Int. J. Comput. Appl.* **57**, 6–14 (2012)
31. Tan, A.-H., Toe, C.: Learning user profiles for personalized information dissemination. In: *The 1998 IEEE International Joint Conference on Neural Networks*, Alaska, pp. 183–188 (1998)
32. Witten, I.H., Pynter, G.W., Frank, E., Gutwin C., Nevill-Manning, C.G.: KEA: practical automatic keyphrase extraction. In: *The Fourth ACM Conference on Digital Libraries*, pp. 254–256 (1999)
33. Zhou, T., Ren, J., Medo, M., Zhang, Y.: Bipartite network projection and personal recommendation. *Phys. Rev. E* **76**, 046115 (2007)
34. Zhou, T., Zoltan, K., Liu, J., Medo, M., Wakeling, J.R., Zhang, Y.: Solving the apparent diversity-accuracy dilemma of recommender systems. *Proc. Natl. Acad. Sci. U.S.A.* **107**(10), 4511–4515 (2010). The National Academy of Sciences