Implementing a Content-Based Recommender System For News Readers

by

Mahta Moattari

Bachelor of Information Technology and Computer Science, Amirkabir University of Tech., 2010

A REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Computer Science

In the Graduate Academic Unit of Computer Science

Supervisor(s):	Weichang Du, PhD Computer Science
Examining Board:	Michael Fleming, PhD, Computer Science, Chair
	Scott Buffet, PhD, Computer Science

This report is accepted by the

Dean of Graduate Studies

THE UNIVERSITY OF NEW BRUNSWICK

May, 2013

Abstract

Recommender systems are widely used to suggest items to users based on users' interests. Content-based recommender systems are popular, specifically in the area of news services. This report describes the implementation of an effective online news recommender system by combining two different algorithms. Our first algorithm employs users' activity histories as inputs. Then it processes this data using a Bayesian framework to predict users' genuine interests[10], and as a result suggests new articles based on those interests. The other algorithm attempts to find keyword matches among the user's keywords and new articles' keywords to suggest new articles to that user. The Java language was used to implement these algorithms. To test the system, ten different users were chosen randomly among those users who posted comments for more than 50 articles from 2012/05/01 to 2012/07/30. These experiments show that our system successfully suggested new articles to users based on their fields of interest.

Dedication

I Dedicate this report to my dear parents. For their endless love, support and encouragement.

Acknowledgements

This report would not have been possible without the help, support and patience of my supervisor, Dr. Du. I would like to thank him for the opportunity and his helpful advice during my study.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction and Overview

Using the Internet gives us the advantage of easy access to information. On the other hand, internet usage leaves us with a massive amount of information, which we need to search through. Having too many options can be confusing and time consuming. People rely on the experiences of others for choosing movies, books, and other products, because they feel overwhelmed by the huge number of options available. This chapter will show that it is preferable to narrow these options down, to something more related to an individual user preference. In other words, content must be selected that meets user needs. Narrowing down options is where recommender systems play an important role.

1.1 Recommender Systems

Recommender Systems (RS) became a significant research area in the mid-1990s, after the first papers on collaborative filtering [1, 2] were writen. Burke [3] has defined recommendation systems as "An information filtering technology, that produces individualized recommendations as output or have the effect of guiding the user in a personalized way to interesting or useful objects in a large space of possible options". In the last decade, a lot of work has been done on developing

new approaches to RS. The area remains highly interesting because of the numerous applications like recommending books, CDs, movies, news, etc., that users use to overcome information overload. RS will collect current and past customer information and connect customers to products using this information.

Various advanced systems find the right data for users using their interests. Amazon helps customers to find the best products online. Pandora provides users with a lot of information and services related to different aspects of music. Pandora also captures user behaviour and helps users to find songs that they might be interested in based on their behaviour. Movie Lens offers to guide users to identify the movies in which they might be interested. YouTube is recognized as the world's largest collection of online videos. It is also known as a system that employs user browsing history to recommend videos. E-commerce websites take advantage of one or more recommender systems to direct the best solution to their customers. Websites such as Netflix, IMDB, Hulu, etc., create a relationship with a customer to recommend movies that suit that customer's interests. Keeping customers is very important to these websites; this relationship is useful for both customers and websites [4].

1.1.1 Content-based and Collaborative Filtering

Mostly, recommendation systems can be categorized as content-based, collaborative, or hybrid [5]:

Content-Based Filtering (CBF) is one of the traditional types of recommender systems. The root of the content-based filtering is in information retrieval [6] and information filtering [7] research. In this method, the algorithm will suggest new items to users based on user interest in the past. Content-based filtering can be used in different recommendation systems such as news article recommendation systems or TV program recommendation systems. The method varies partly

in each of these systems. However, some fundamental concepts stay the same, like the two sets of information that it works with: 1) a set of features that describe the items to be recommended and 2) a user profile built from past choices that the user made. Finally, content-based filtering will use information gained from the two sets to recommend a new item system compares any new item with those that exist in the user's profile [8]. However, CB techniques have some limitations, like the data scarcity problem. The only resource for modeling user interest is extracting features from their browsing or purchasing history [5]. Therefore, CB systems are not able to identify different items that the user may enjoy, because they attempt to find those items that are very similar to the items in the history of that user.

Collaborative Filtering (CF) is perhaps the most common and most widely implemented of the RS technologies [3]. Collaborative recommendations act based on user-user similarity. In this approach, the system will recommend new items to users based on other similar users' interests. For example, if there is a set of users called U, a new user called u, and a set of items called I, collaborative filtering will determine the users from U that have some similarities with u. These users would be labeled u'. Then the system will recommend those items from set I to u that are most popular with u'. One of the weaknesses in this system is that it cannot recommend items to users before some users have ranked the items. This characteristic may not cause a serious problem in some fields, but it can be really an issue for some subjects, like recommending news. Although Collaborative filtering methods are able to produce high quality recommendations, the performance decreases with the number of customers and products. Moreover, a user with unusual interests may not receive recommendations unless there are other users that have the same interests.

Hybrid Recommender System: Because every recommender system approach has its own limitations and weaknesses, one might want to combine them in different ways to avoid those weaknesses. These combinations can be categorized as

follows [9]:

- Implementing collaborative and content-based methods independently and mixing their predictions
- Combining some of the content-based features with collaborative method
- Combining some of the collaborative features with content-based method
- Forming an overall synthetic model that combines both content-based and collaborative features

1.1.2 Research Overview

Having outlined the strengths and weaknesses of the RS methods and having reviewed the evaluations of implementations that have been done in different areas (recommending CDs, books, movies or news), it is possible to choose the most appropriate approach in order to design a recommender system for specific usage. In this project, we aim to design a recommender system for news sites like the Canadian Broadcasting Corporation (CBC). Reading news online means news readers can access various news providers' articles around the world, which is why online news reading is popular. However, having so many options creates the problem of how to find the desired articles as easily and as fast as users need [10]. The answer to this problem is using recommender systems, and the most popular one in this field is the content-based recommender system. This method plays a vital role in recommender systems, because using this approach gives us the chance to recommend things that have not been rated before [10].

A collaborative filtering(CF) system has two main downsides. First, it is not able to recommend those stories that have not been yet read by other users, which means it cannot recommend new stories within one hour of their publication. This delay is a big issue for a news recommendation, because there are a lot of new stories

every day, which must be seen as they are published. Second, CF can result in an inaccurate suggestion. For instance, one of the most popular news categories is entertainment, and because of such a high popularity rate, it is likely that a user will have some similarity with users who like this kind of news. If a user does not like entertainment news stories, because of the popularity of entertainment news, these news stories will be recommended to them. This mis-recommendation happens because collaborative filtering does not involve users' interests. As a result, content-based filtering or a combination of content-based and collaborative filtering is a more appropriate method for news recommender systems.

Liu et. al. [10] present research on developing a personalized news recommendation system in Google News. In their recommendation system, they create user profiles for those users who are logged in and have explicitly enabled web history. The profile shows users' news interests based on their past click behaviour. In order to create a user profile, they used a content-based method. The accuracy of this profile is a critical key to the success of this method. Since user interests have been derived from user activity in the past, Liu et. al. [10] tried to track users' change in interest over time and to find out if it is effective to use this information to predict future user behaviour. They measure user stability using a large-scale log analysis of Google News users. To guess the news interests of a specific user based on her/his activity in the past a Bayesian model was developed. Liu et. al. [10] also tried to predict current news trends. They recommend an article to a user using the user's genuine interests and current news trends. As a result, current important events (hot news items are not part of a user's interests, but the user might like to know about them) and stories that user might be interested in will be suggested to the user. Then Liu et. al. [10] combined their method with the collaborative method that existed in Google news' previous recommender system and offered a hybrid recommender system for Google News.

In their paper, [10] mentioned that for privacy protection reasons, detailed information about user clicks, such as the amount of time spent on the article, is not recorded by Google News. So their information about user interests might not be entirely accurate, because it contains noise. This noisy data can make the system results less accurate, specifically when dealing with small amounts of information. We tried to develop the same method as [10] while we have access to a smaller amount of data, so the data accuracy is more important in our case.

As a result, to create user profiles and guess user interests based on the same Bayesian framework, we considered user comments on the articles instead of user clicks. This is because, as mentioned earlier, user clicks on an article do not definitively prove that a user read the article and cannot be used as a proof of user interest. When the amount of data is small, inaccurate data can affect results and cannot be ignored. User comments on the articles assures us about user interest in the specific subject. One important aspect of the CBF method that is used in [10] is categorization. Because the articles in a news site are already categorized based on their subject, all we need to do is use the same categories. We used the same idea and categorized the articles based on the existing categories on the CBC website and then predicted user interest using the Bayesian framework. Also, we used another concept called keyword to rank websites that have been seen by the user. We used this ranking to estimate user interest and recommend articles to users as well. We combined the results of these different methods to recommend new articles to users. Our motivation for combining these methods was our user interest crossover. As an example of user interest crossover, a user who is mostly interested in political topics might be interested in a story about a politician playing football. Our contribution in this report was to create a content-based recommender system for a news website like CBC News that works as accurately as possible with a small amount of data. Furthermore, a goal was to be able to define user interests in a more flexible way.

1.1.3 Report Structure

In the next chapter we will talk about different recommender systems and advantages and disadvantages of each of them. We also mention their performance in different areas. In Chapter 3, we explain our scheme based on our case study, which is the CBC news site. The implementation part is described in Chapter 4, and Chapter 5 includes results and analysis of experiments that we've done on some random users. Finally, in Chapter 6 we will talk about future work and conclusion.

Chapter 2

Background

In this chapter, we explain content-based filtering in more depth. We also explain recommender systems based on content-based filtering.

2.1 Content-Based Filtering

As mentioned earlier, the root of content-based filtering is in information retrieval [6] and information filtering [7] research. In content-based filtering, new items are suggested to the user considering past user interests. In some fundamental concepts the method has similarity with CF, like the two sets of information: 1) a set of features that describe the items to be recommended and 2) a user profile built based on the user's choices in the past. Moreover, to recommend a new item, the system compares any new item with those that already exist in the user profile. As a result, we define each item with a set of attributes. Keeping items in a database is relatively easy for structured data, because we can describe them as attributes. Table 1 shows an example of structured data. In this example few attributes have been used, and each item is defined by the same set of attributes. In this case, different machine learning algorithms can be effortlessly employed to create a user profile or a user model. For unstructured data such as news articles we first need to

do some preprocessing to be able to define some attribute names and related values for them [8].

ID	Name	Cuisine	Service	Cost
10001	Mike's Pizza	Italian	Counter	Low
10002	Chris's Cafe	French	Table	Medium
10003	Jacques' Bistro	French	Table	High

Table 2.1: A restaurant database

This preprocessing is related to text processing or language processing, which is quite challenging work. One of the challenges is the existence of polysemous words (when the same word has several meanings) and synonyms (when different words have the same meaning). For example, in an article, apple may refer to an electronic device by the Apple Company or it might refer to the fruit. Similarly, power and electricity may also refer to the same thing [8].

One way of dealing with unstructured data is to represent it as structured data. This can happen in a Boolean manner because if we think of each word as an attribute, the presence and absence of words is the value for that attribute. Also, using structuring we can determine the importance of the words. Many weighting schemes can be used to calculate the importance of a word, but one of the most common and well-known methods for specifying word weights is known as term frequency/inverse document frequency (TF*IDF). TF refers the term's frequency, described as the number of times term t appears in the document d (TF(t,d)). Documents that have more occurrences of a given term obtain a higher score. In this way we can define the importance of words in a given article. Later, if the same words were repeated in a new article, it is likely that they are related. In this

method, we can also use the roots of the words instead of thinking of each word as an attribute [8].

2.1.1 User Profile

User Profiles can be used in most recommendation systems. As previously mentioned, user profiles demonstrate user interests. A user profile can contain different types of data. In this section we will discuss two of them.

- One type of data is user preferences, and the likelihood of that preference. Actually, for efficiency purposes, we might like to have the n most interesting items in the user profile
- Another type of data is the user interaction history, which contains the items that user has recently seen.

Having a history of user interaction can be useful. For example, in a news website, we need to know the articles that the user has read recently to avoid recommending them to him/her again.

A user profile can be created manually (explicitly), asking a user to define his/her interests by using check boxes. Such information includes age, gender, location, interests, etc. Another option that most websites choose is to ask for feedback on a product. Users are supposed to provide feedback either in the form of surveys or through general forms. However, the problem with this method is that most users refuse to write feedback and, even if they do, they might not give a complete list of their interests. E-commerce websites sometimes request that users express their opinions by selecting a value in a range of explicit ratings. This is usually less troublesome to users than filling out forms. On the other hand, user interest might change after a while and it is hard to manually update their preferences. So we are more interested in defining user profiles implicitly [8]. Monitoring user activity over

the web can provide implicit feedback. Usually, users are not aware that they are providing feedback to the system. One such example is YouTube, where a direct relation between the amount of time spent on a single video per session and user interest can be captured. This type of feedback may not be as accurate as the feedback in the form of explicit ratings by users, but users are not disturbed.

2.1.2 Learning a User Model

Creating a user model based on user preferences considered to be classification learning. There are different classification algorithms that we can use for this matter. ID3 is a decision tree algorithm. In ID3, partitioning training data in a recursive manner creates a tree. For example, texts can be classified under different subgroups until all the instances in a group belong to the same class. The partitions are formed based on certain features. For example, if we are classifying text, the classification feature can be the presence of a special word. Different studies show that although the ID3 algorithm can be an effective algorithm in case of structured data, it is not our best choice for unstructured data. For example, in the restaurant example, ID3 can easily do the job as a learning algorithm, while for news articles, ID3 may perform poorly. Because small trees are more preferable when we are using the decision tree method, the algorithm has a tendency to consider a small number of features. However, in text classification, most of the time there are lots of features, which is in contrast with the algorithm characteristic [8].

Another algorithm that can be used in this regard is the nearest neighbor algorithm. In this algorithm, similarities between the training items need to be found. Then the k nearest neighbors are classified in k group, and a label is defined for each group. Later, when we have an unlabeled item we first need to find out to which group the item belongs. Then based on the label of the group the label of the item can be defined as well. Depending on the data type (whether we are dealing with structured

or unstructured data) we can choose our similarity function. The Euclidean distance metric is often used for structured data, while in the case of unstructured data, the cosine similarity measure is often used [8]. Both of these similarity functions have been applied to several text classification applications [11, 12] and the nearest neighbor performs competitively with more complex algorithms, despite the absolute simplicity of the algorithm [8].

2.1.3 Probabilistic Methods

One approach in text classification is using probabilistic methods. There has been a lot of work in this regard. The Naive Bayes approach is a great example of a probabilistic method. The naive Bayes classifier is not only competitive with other learning algorithms, such as decision tree and neural network algorithms, but in some cases it has better performance. This algorithm performs exceptionally in text classification and has been used in many recent works.

In this section, we are going to discuss the naive Bayes classifier in detail, and our focus will be on applying the algorithm to the problem of learning how to classify text documents such as electronic news articles. This algorithm is one of the best algorithms for this type of learning problem. Bayesian methods are difficult to apply because they usually need initial knowledge of many probabilities[7]. Bayesian learning methods are based on Bayes' theorem, which is:

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)} \tag{2.1}$$

where $P(h)$ is the initial probability that hypothesis h holds, independent from training data. $P(h)$ is called prior probability of h. $P(D)$ is the prior probability of training data D given no knowledge about what the hypothesis holds. $P(D|h)$ is the probability of D given h. finally $P(D|h)$ is the probability that h holds given

the training data D, which also is called the posterior probability of h[7].

2.1.4 Advantages and Disadvantages of CBR

Advantages of content-based recommender systems includes:

- In order to construct the user profile, implicit feedback from users is enough.
- With the growth of database content over time, content-based recommender systems gradually perform better.
- CBR can support recommendation of new or unpopular items to users based on their interests.

Disadvantages of content-based recommender systems includes:

- CBR systems are limited to characters that are explicitly associated with items.
- Items might be indistinguishable when they are represented by the same set of features.
- When the CBR system only recommends items that are rated highly against a user's profile it is called over-specialization. In such cases, the user is limited to seeing items like those already rated. This problem can be solved by injecting some randomness in the predictions.
- New users may not be able to get precise predictions according to their tastes. Users need to have a sufficient number of ratings to get better predictions, which is sometimes not possible.
- Another example occurs when dealing with items like jokes or poems. We easily can determine if a joke is a lawyer joke or chicken joke, based on the word frequencies or word presence. However, distinguishing a funny chicken joke from other chicken jokes is not an easy task.

To solve this problem we can use other recommender systems like collaborative recommenders or we can even combine two methods in a different way, which is a called hybrid recommender system [8].

2.2 Related Work

The root of content-based filtering is in information retrieval [6] and information filtering [7] research. As we said earlier, content-based recommender systems suggest items similar to those that a user has previously liked [13, 14, 15]. In other words, different candidate items are compared with items that were rated by the user in the past and the best-matching items are recommended. In [16] they present an implicit news recommender system with the capability of integrating multiple modes, like radio-television channels, websites, and other kind of media like written text and spoken words. Di Massa et. al. [16] used natural language processing techniques like TF*IDF to extract users' interests from their RSS blogs. Then the relationship between those interests and both online newspaper articles and broadcast news stories is found using semantic analysis. Di Massa et. al.'s [16] recommender system is a combination of content-based filtering and collaborative filtering. In content-based filtering, the TF*IDF technique was used to define their keywords and then find articles' keywords that match the user interests, using a similarity function. They also find the similarity among users employing collaborative filtering.

In [10] they defined users' interests based on their clicks on different news categories. They create user profiles and use a Bayesian framework in order to predict user interests. Their recommender system is considered a content-based recommender system and is combined with the previous Google news recommender system, which was designed based on collaborative filtering. [15] In order to estimate the probability that a document is liked, they use a Bayesian classifier as well.

Moreover, to create a content-based recommendation, other techniques have been used, such as different machine learning techniques, including clustering, decision trees, and artificial neural networks [6]. These techniques vary from information retrieval-based methods in that they calculate utility predictions based not on a heuristic formula, such as a cosine similarity measure, but rather on a model learned from the underlying data using statistical learning and machine learning techniques [9].

In [17] they designed a recommender system, which is named Foxtrot, to recommend on-line research papers to academic researchers. Foxtrot is a hybrid recommender system. This system used methods of their previous work, Quickstep [18], plus added support for collaborative filtering, profile visualization and an expanded ontology. Papers are classified using a research paper topic ontology and a set of training examples for each topic. Recording users' web browsing and relevance feedback creates user profiles. Collaborative filtering is used to recommend papers to users. In [19] they discuss that the performance of content-based news recommender systems can be improved by using current probabilistic retrieval algorithms rather than by using relatively old and simple matching algorithms. In this paper, they referred to techniques such as simple keyword matching and TF*IDF as old algorithms, which were used in content-based recommender systems.

Chapter 3

System Design

In this chapter, we will discuss news recommendation systems and the scheme of our recommendation system in detail.

3.1 Introduction

The World Wide Web has hugely changed our everyday life. One of these changes is the migration of lots of news readers from traditional news reading, which involves physical newspapers, to online news reading using the Internet. Online news reading has become popular because news readers can access many news articles quickly and easily from all around the world. However, it is not that easy to find the desired articles when there are many options available[10]. To solve this problem, many websites use recommender systems. These systems suggest articles to news readers and the most popular approach in this field is the content-based recommender system. In general, content-based filtering is a great method that can be used in different recommendation systems. The key point of content-based filtering is to find new items that might be important to users by matching them with previous user interests. One of the advantages of this method is that CBF is able to suggest items that have never been rated before. This characteristic plays an important role

in news recommender systems, because it takes time for new articles to be ranked by users. Also the CBF method is able to determine individual differences between users. This technique has been used in various fields like email [20], news [21], and web search engines [22]. This method in the news domain is actually creating a personal newspaper by aggregating different articles that are interesting to the user. For the success of the CBF recommender system, it is critical to create an accurate user profile, which shows current user interests. Making an accurate user profile can be done manually or automatically [10]. Knowing how users' news interests change over time is essential, because user profiles are gathered from users' past activity. Understanding users' past interests, and their effect on predicting user behaviour in the future, is also vital.

In [10] they showed a large-scale log analysis of Google News users to measure the constancy of users' news interests. Based on what they found, users' interests changed over time based on the aggregate trend of news events. Furthermore, they predicted users' interests by developing a Bayesian model, and also employed a group of users' activities to show news trends. In their system in order to recommend new stories to an individual user, they considered users' genuine interests and current news trends. By using this method, a user will not miss the important news of the day, even if those news stories do not match user interests. We used their Bayesian model with slight changes to implement part of our content-based recommender system for the CBC news website.

3.2 Basic Steps

Our recommender system is meant to work with the CBC news site. In the CBC news site, in order to post a comment under specific articles, users need to sign up and log in. Also, articles are categorized under different categories and

subcategories. Two main news categories in this website are News and Sports. There are eight different subcategories for articles under the News category: world, Canada, politics, offbeat, arts, technology, health and business. Also there are eleven different subcategories for articles under the Sports category: soccer, football, basketball, hockey, baseball, ski, figure skating, golf, tennis, gymnastic and swim. The following picture demonstrates the architecture of our recommender system based on the CBC news site.

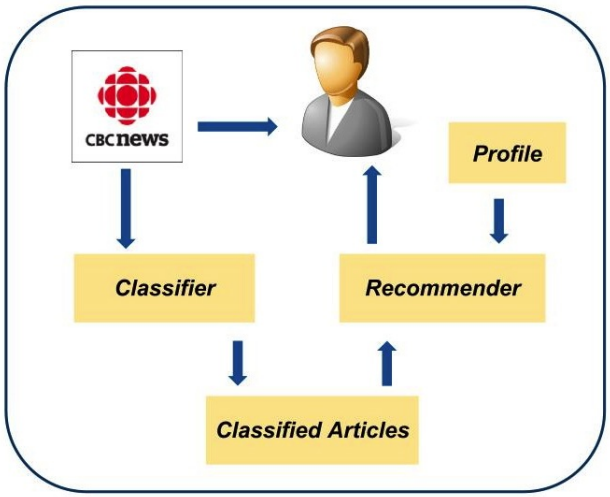

Figure 3.1: Architecture of our recommender system based on CBC news.

The classifier categorizes articles based on the CBC news classification. These classified articles are used in the recommender system to define users' interests. Users profiles are created implicitly, so there is no direct interaction between users and user profiles[27]. Later we will explain about creating users profiles implicitly in detail.

When we are designing a recommender system, we can use different techniques that we mentioned earlier. However, not all those techniques perform equally well in different situations. For example, the previous Google news recommender system was based on collaborative filtering [10]. CF can only recommend news that has been read by users with similar interests. This method has two major limitations when it comes to recommending news stories. One of them is the first-rate problem, which means we cannot recommend new stories that have not been read yet. This

problem is a serious one for a news website, because these websites are trying to present the most recent information to their users. Also, most of the time, users are interested in reading a news story on the same day of its publication and not after that. However, when we are using CF, the system will need at least some hours to be able to recommend the news story to a user. Also, it is hard to consider user individual interests when we are using CF. For example, some categories are generally popular, like entertainment. So, in the CF method, there is a great chance that the system will always find enough of a user's neighbors who are interested in entertainment news stories. Thus, it will keep suggesting that kind of news to users who are not interested in those topics.

Therefore, when it comes to recommender systems for news websites, it is better to use a content-based method or a hybrid method, which is a combination of content-based and collaborative filtering. Using this method, we can have access to users' genuine interests. Also, we would be able to suggest news that has not yet been rated. As we mentioned earlier, we have categorized news articles based on their topics by using text retrieval techniques. On the other hand, we employed users' news reading histories to predict the categories that is interesting for each particular user. The articles in those categories will be ranked higher than the rest of the articles and so they will be recommended to the user.

We categorized news articles into general topics just like the CBC news categorization, for two reasons. First, based on the data that have been stored in our database, it does not seem to be a good idea to categorize art and entertainment articles to subcategories like music, painting, etc., because there are small numbers of articles in these categories. Also, we decided to focus more on categorizing articles related to sport, since the CBC news website focused on categorizing sports news as well. So as a result, we have two basic categories, news and sports. The news category contains subcategories such as politics, art, health, and world news

that can also contain any of the mentioned topics. The sport category is divided into subcategories like hockey, football, golf, etc. This categorization was applied to users' interests categorization as well.

We implemented the Bayesian model from[10] with small changes. One difference is in the way that we describe user interest toward a special topic. In [10] they consider a user's clicks as a fact that shows user interest. In contrast we decided to count user comment(s) under different articles. Therefore comments are treated as a positive vote by a specific user for a specific topic. Our reason for using user comments instead of user clicks is that considering user clicks as evidence of user interest without knowing the time that user spent on the article gives us noisy data. Such noise might be easy to ignore if we have so much data that the ratio of inaccurate data to the total data that we have collected for a specific user might be insignificant. When a user posts a comment under an article it is rational to take it as a proof of user interest regardless of the content of the comment; otherwise, why would the user spend some time and write something even negative about a topic that she/he is not interested in at all.

This technique has its own limitations; for example, a user might not post comments for all the topics that she/he is interested in. Overall, we will have smaller amounts of data by taking into account user comments instead of user clicks but also less noise. Another motivation for us in choosing user comments over user clicks was accessibility. In many situations user clicks might be considered as private user information, while user comments are public and can be seen by anyone.

Our recommender system has two limitations. First, user interests might change over time, so we should be able to update the system incrementally. On the other hand, as we said, we are considering user comments as a positive vote; as a result, sometimes we might have little information, and our system should be able to perform reasonably even with a small amount of data.

3.3 User Interests Log Analysis

In every system in wich user interests (in the future) are predicted based on user interests in the past, user interest stability over time is an important factor. Different analysis has been done in this regard. In Wedig and Madani [23], they present large-scale analysis of Yahoo! search engine query logs to determine user interest stability and some other factors related to personalization. They found out after hundreds of queries that user interest distribution converges to a stationary distribution. Because user attitude toward searching is different from news reading, in [10] they conducted a large-scale log analysis of click behaviour on Google news. Their targets were those users that have an account and also explicitly enabled history tracking over a 12-month period. They sampled those users who had at least 10 clicks during 2007/07/01 to 2008/06/30. For each individual user u, they calculated a click distribution for each month t, based on different categories, and presented the result as a vector:

$$D(u,t) = (\frac{N_1}{N_{total}}, \frac{N_2}{N_{total}}, ..., \frac{N_n}{N_{total}}) \\ where N_{total} = \sum_i N_i \tag{3.1}$$

Assume we have a set of categories, $C = c_1, c_2, .., c_n$, and that the number of clicks that a user made under category c_i in month t is defined as N_i. Also N_{total} is the total number of clicks made by the user during month t. So $D(u,t)$ is presenting us with the proportion of time that the user spent on each category during the time period. For each user they compared user click distribution for the most recent month with the click distribution of all previous months. They concluded that the difference between click distribution of the user's most recent month and past months increases as they moved back in history. As a result, the old user history is less valuable for predicting the user's future interests. Figure 3.2 demonstrates the graph that they

drew to present their results.

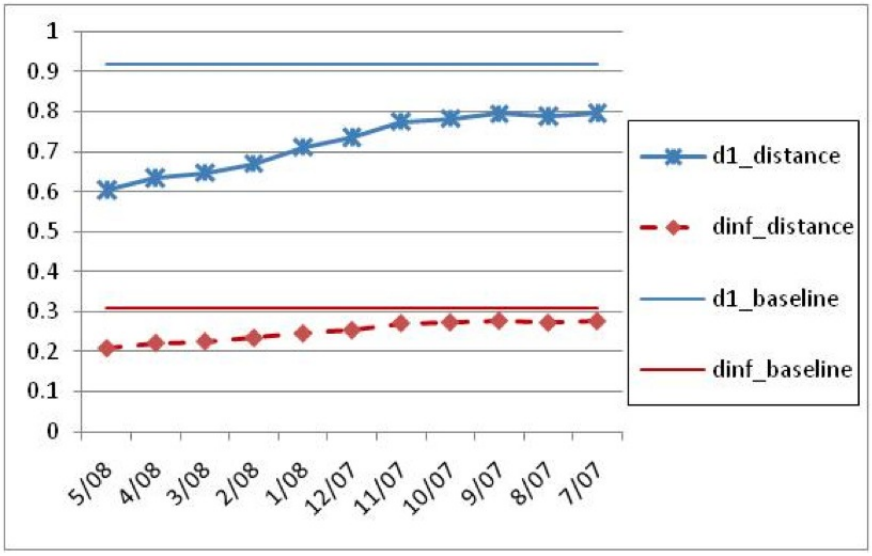

Figure 3.2: Click distribution of users in different months[10].

3.3.1 News Trend

Earlier we discussed two different factors[10] that must be taken into account in order to recommend news articles to users. We said that one of the elements that has an effect on user interests in news reading is news trends. In [10] they categorized users based on their nationality and analyzed their general behaviour towards big news events in their country. For each country, the distribution of all the clicks made by the users from that country in a past time period t, represents the common interests, and is denoted as $D(t)$. They assumed that the general interests of the users of a country would change with respect to the big news events of that country, and the log analysis supports this assumption as well. For example, during the US election campaign that started in late 2007, the percentage of clicks on national news stories doubled. This shows that those users who do not normally pay attention to national politics became interested in this category during this time. Figure 3.3 displays the results of their log analysis.

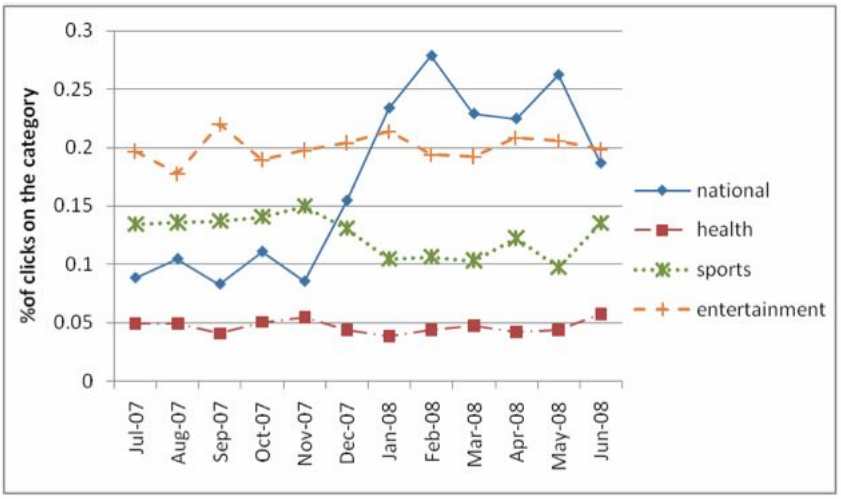

Figure 3.3: Changes in user interests according to news trends in USA[10].

Another example shows changes in user interests in different countries during the 2008 Olympic games. As is shown in Figure 3.4, in August 2008, the graph of general interests in sports news shows an increase in user clicks in countries like Spain, United Kingdom and United States. Also it is obvious that sports news are more popular among users in Spain compared to users located in UK and US.

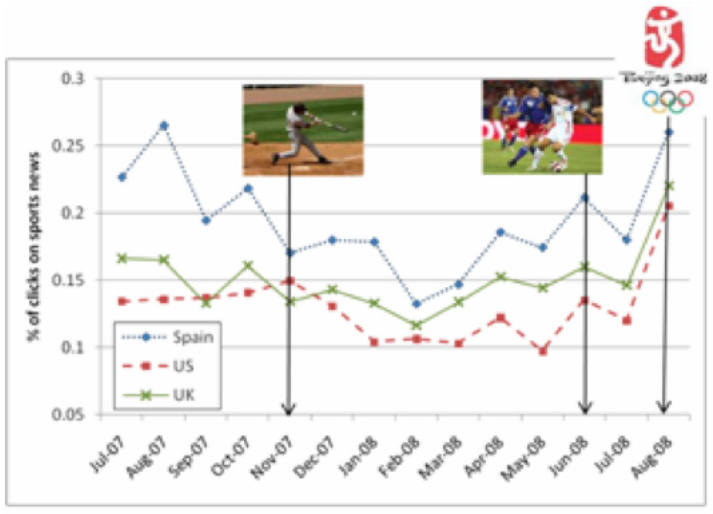

Figure 3.4: Changes in user interests in sports-related articles according to news trends in different countries[10]

This graph also illustrates some additional trends in user's interest in sports-related stories The dramatic drop in the graph of US after November in 2007 is

related to the end of the baseball season in US. Also the rise of the graph in June 2008 is because of Euro Cup games, which US users were less interested in, compared to the graphs of UK and Spain.

3.4 Bayesian Model

The log analysis demonstrates that user interests are predictable based on two different factors: user activities in the past (long-term interests) and current news trends (short-term interests). The long-term user interests, which are also called genuine user interests, are related to user characteristics, so it will stay with the user for a long time. On the other hand, short-term user interests are based on the big news trends and probably will not last longer than the life of the news event itself. In [10] they considered the ratio of total user clicks on special articles to the total user clicks in general for the past several hours, to determine the big news trends. Since we are using user comments as evidence of user interest in specific topics, we cannot determine news trends easily, because it is hard to find enough comments in the few hours after the articles were published.

As a result we changed the framework to suit our purpose, which is predicting the users' genuine interests according to their activities in the past. The system will predict user interest for each time t, so in the next step we combine the results to achieve an accurate prediction of the user's genuine news interests. However, as we mentioned earlier, we used keywords of articles to find new articles that match the user's interests as well. Thus the final recommendation is based on the results of both these techniques. We can control the effect of each of these techniques in our recommender system by selecting the desired rate.

3.4.1 Predicting Users' Genuine News Interests

We considered the comment distribution of a particular user for a specific time period, t, in the past, presented by $D(u,t)$. We would like to discover the user's genuine interests revealed in $D(u,t)$. In order to compute the genuine interest of a user in a specific category c_i, we need to find the probability of a user being interested (posting a comment) in the articles given the category in a period of time $t, p^t(comment|category = c_i)$. We can compute this probability using a Bayesian rule, as follows:

$$\begin{aligned} interest^t(category = c_i) &= p^t(comment|category = c_i) \\ &= (\frac{p^t(category = c_i|comment)p^t(comment)}{p^t(category = c_i)}) \end{aligned} \tag{3.2}$$

$p^t(category = c_i|comment)$ shows the probability of a user's comment being in category c_i. The number of the user's comments in category c_i over the total number of user comments in all categories in time period t gives us this probability. To calculate $p^t(category = c_i)$ we should find the ratio of articles in the specific category c_i to the total number of articles in the time period t. $P^t(comments)$ is the probability of the user commenting on any article, regardless of the article category, which means the total number of the user's comments on different categories over the number of all comments in time period t.

3.4.2 Combining predictions of past time periods

The previous equation calculates the user's genuine interests in a particular time period. To accurately find the user's interests we need to combine those predictions. We shall use N^t (number of user's comments in time period t) to normalize the predictions that we made in time period t, because the greater the number of a user's comments that have been recorded, the better the predictions that can be

computed. As a result the following formula can be used to predict the user's genuine interests.

$$interest(category = c_i) = (\frac{\sum_t (N^t * interest^t(category = c_i))}{\sum_t N^t}) \tag{3.3}$$

3.5 Suggesting new articles to a user, using keywords

By choosing an article from a website like CBC news and looking for the source of the page, keywords can be found within the Meta tag (Figure 3.5). This Meta tag contains the words that were repeated in the articles. Search engines use this Meta tag very often to return the results that match the requested query.

Figure 3.5: Meta tag containing the words that were repeated in the articles

We extract these keywords automatically from page sources and store them in our database, to help us define categories for different articles. As we mentioned earlier, categorizing articles based on CBC categorization limits our recommender system when it comes to user interest crossover. Crossover happens when a particular user is not interested in category c_i in general but she/he is attracted to a specific article from category c_i because this particular article contains something

that matches the user's interests. For example user A is interested in movies, health, and world news, and there is no evidence that this user is interested in sport at all. Also article B is categorized under football, but the content shows that the football game was between some famous actors. So based on the content we can say that user A might be interested in this article but based on our previous categorization, there is no way that this article would be recommended to user A.

As we said, keywords contain those words in the content that were repeated the most. So if we compare the keywords of article B to the keywords of the articles that have been read by user A in the past, we might be able to find word(s) that match. Thus we will be able to recommend article B to user A although they do not belong to the same category, based on our categorization. This method helps to evaluate user interests from different points of view. In fact it is a different categorization of user interests, which differs from regular categorization like sports, politics, etc. Finding new articles based on different categorizations will result in a more accurate recommendation.

For the sake of simplicity, the keywords are not actually categorized, so every time that we want to find a new article, the keywords of the article should be compared to the keywords of the articles that have been visited by the user in the past to find out if there is a match. This comparison will result in a very accurate answer. On the other hand, it will take lot of time, especially if the number of user keywords is high. So one can use different machine learning algorithms to categorize them.

Chapter 4

System Implementation

We developed a content-based recommender system based on the proposed model that recommends new articles to users based on their history of posting comments. Figure 4.1 demonstrates the elements in our system.

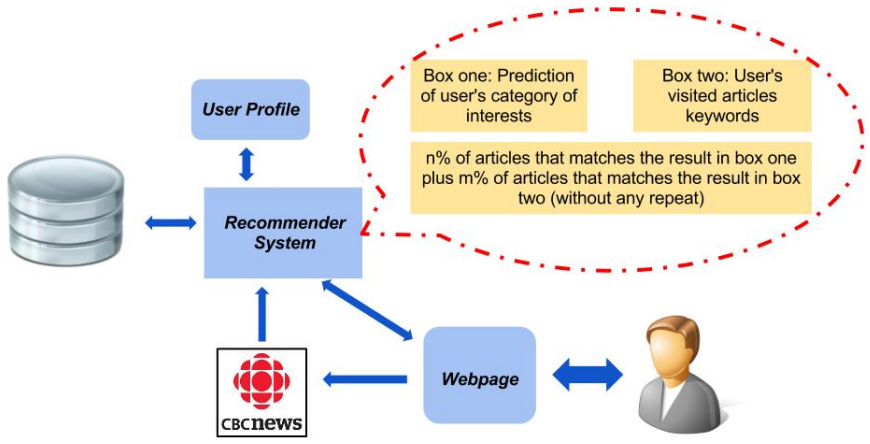

Figure 4.1: System architecture.

In our first step different data like user Id, user comments, article titles, article keywords, etc. were collected automatically from CBC News website from 2012.05.01 untill 2012.08.30, and stored in a database. In this project MySQL is used as a

relational database management system (RDBMS). MySQL is the most popular open source DBMS [24].

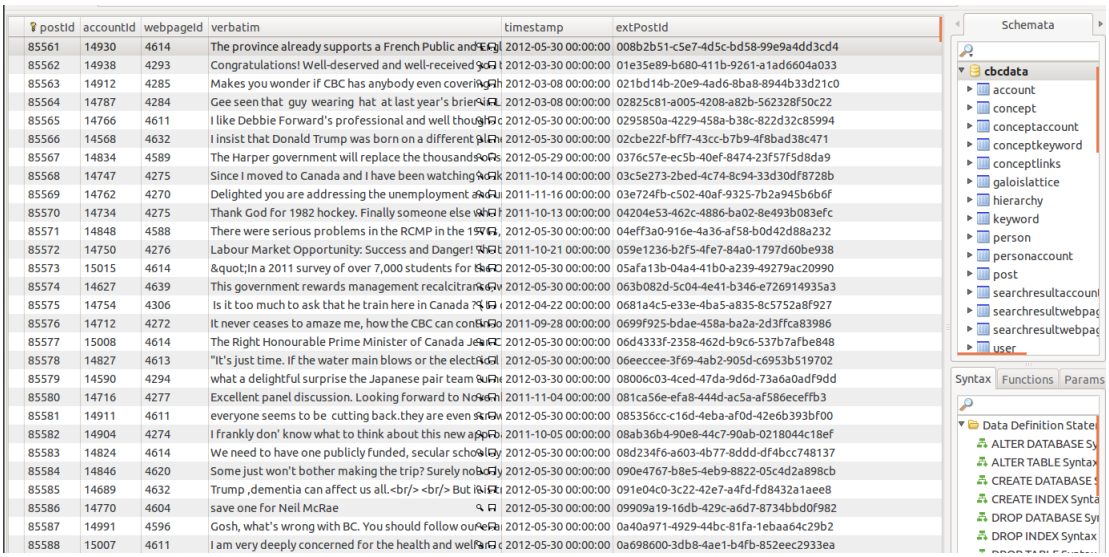

Figure 4.2: Result of a query in MySQL.

Then specific queries are requested from the database using Java. Eclipse has been used as a development environment (Figure 4.3). This result might be saved in the data structure Array List of different classes or might be used directly for calculations or comparisons.

The goal is to predict user interests using the proposed Bayesian model, and then based on those predictions and also article keywords comparison, recommend new articles to the user. So first of all we look for the visited articles (which are stored in our database under the webpage attribute) and their categories, and by visited articles we mean those articles that the user talked about by posting comment(s), and then we keep the number of visited articles for a specific category and date. Also we need the number of all user comments and the number of all articles for that specific category and date. Using this information we are able to predict user interest for different dates.

The result of this calculation is stored. Thus whenever we need to calculate the genuine user interest, we combine the stored result of different dates to find the

answer. One advantage of storing predictions based on date and combining them whenever we need is that, as soon as we update our database, we can compute the user interest for the new date and use it in our calculation. Also we can eliminate those results that are outdated. As we mentioned in chapter 3, user interest changes over time. Aside from using user history to predict a user's interests, whenever a new article's keywords are entered into our system, we search our database using the given userId and the keywords to find matches. Finding matches shows that the user is interested in those new articles, so they can be recommended to the user.

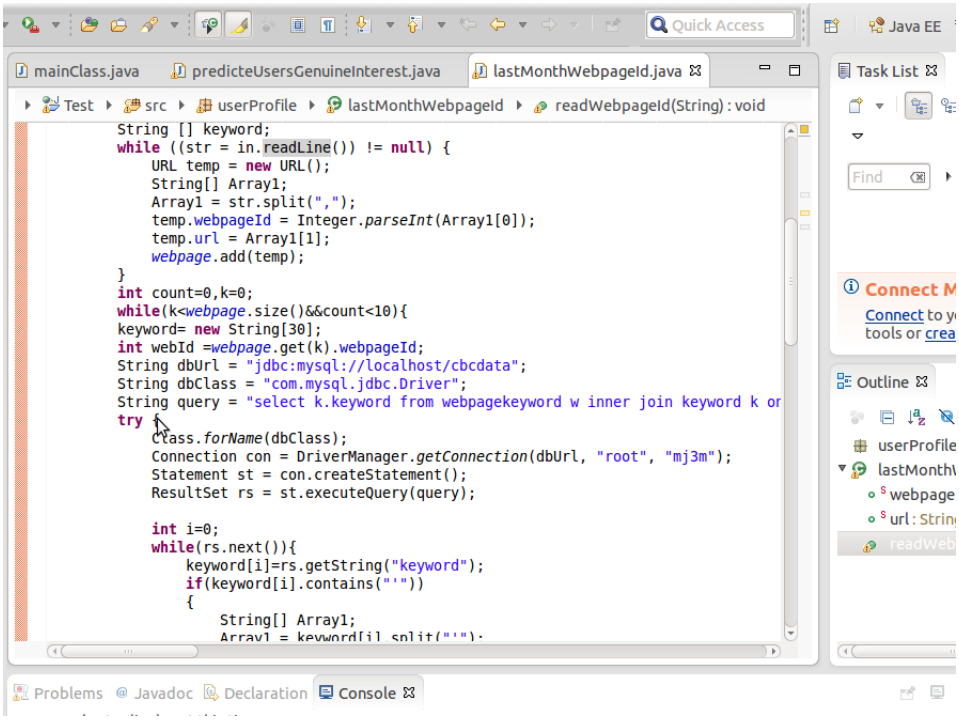

Figure 4.3: A query is requested from the database using Java

In order to test the system we designed a very simple website, which we called demo website. In this website we can choose a specific user Id from a drop down list and submit it to see the links of articles that are recommended to that particular user, as in Figure 4.4. By clicking on the links the real pages can be visited and the user can have access to recommended articles. The website is designed using Java Server Pages (JSP), which is a technology that helps software developers construct dynamically generated web pages based on HTML, XML, or other document types[25].

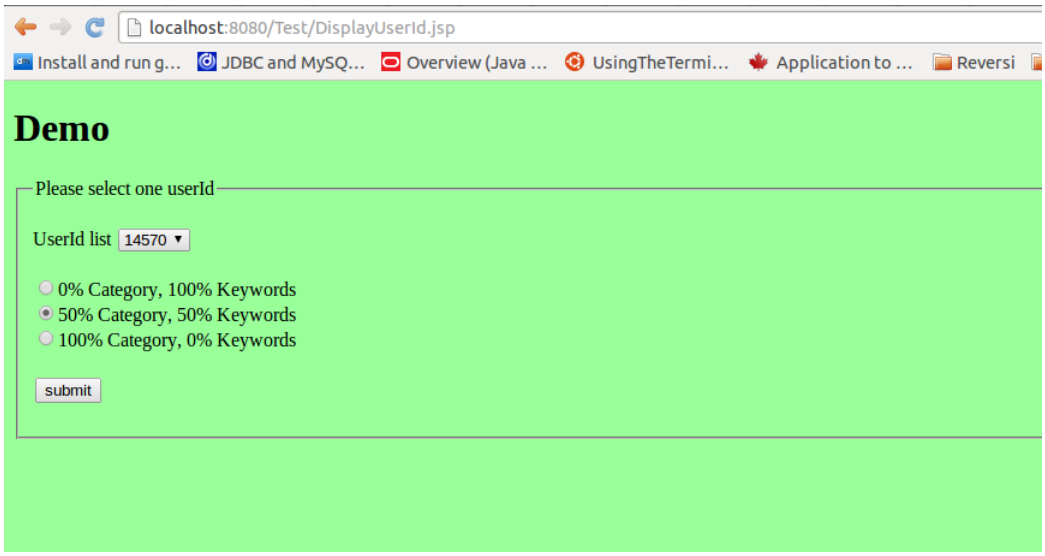

Figure 4.4: First page of our demo webpage

When a user Id is submitted, it will be sent to the predict-user-genuine-interest function in the program where different data related to that user can be easily accessible. There is a list of new articles' Ids and their titles in the database. Using this information, the category of each of the new articles will be determined and then checked to see if this particular user is interested in any of the specified new articles, based on the results of our prediction. Those articles that match the user's interests will be displayed on the screen as a link (Figure 4.5). We will also look for the matches between the new articles' keywords and the user's keywords, which are stored in our database. If any match was found, it will be displayed as a link along with the other results. Also we can define a threshold, so if the number of matches meets the threshold, the article will be recommended to the user; otherwise it will be ignored.

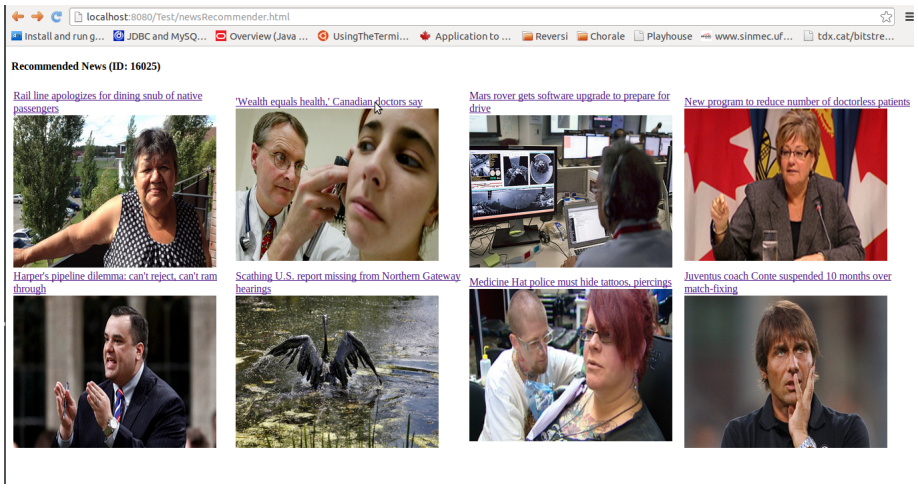

Figure 4.5: Articles that are recommended to a specific user.

Aside from user Id, the user can choose the percentage of involvement of each method in order to get suggestions (Figure 4.4). For example the user can choose to use methods separately, or the user might choose 50 percent of each method.

Chapter 5

Experiments

To evaluate the accuracy and performance of our recommender system we ran some experiments. Random samples were selected and analyzed, using the algorithm separately and in combination.

5.1 Experimental Data Sets

We selected 10 users randomly using the condition of having visited at least 50 articles from 01/05/2012 to 30/07/2012. We chose the threshold of 50 based on our experiments, which show that our system will perform poorly if the number of visited articles drops below 50. Also considering the selected period of time, which is 90 days, if a user reads fewer than 50 articles in 90 days the probably she/he is not really interested in the newspaper after all.

5.2 Experimental Method

The method that we used to evaluate our system is precision and recall. This method is used widely to evaluate pattern recognition and information retrieval methods. For example consider an information retrieval case: documents are the

instances, and relevant documents should be returned in response to a given search term. The fraction of retrieved instances that are related to the search term demonstrates precision, and the fraction of related instances that are retrieved is called recall. So a precision score of 1 shows that all the results were relevant to the search term, but it cannot tell us if the entire set of existing relevant documents were retrieved. On the other hand, a recall score of 1 indicates that the entire existing relevant documents were retrieved but it is not able to show us how many of the retrieved instances were irrelevant. There is an inverse relationship between precision and recall. If one tries to increase recall, precision will decrease and vice versa: increasing precision decreases recall. So these two concepts are not isolated from each other and the results show the system performance in relation to each other[26].

5.3 Experimental Results and Analysis

As we said, we chose 10 different users to test our system, and we used the precision and recall method to evaluate the results. In our first step we found user interests for each of these 10 users, using the information stored in our database (a human checked the database and drew the following table). We define human interests based on the article's category in the CBC website. Subcategories like Canada, health, etc. are shown in the table. These subcategories are under the News category. We searched the database for each user and counted the number of visited articles in each category. Our test data are those articles that were released from 02/08/2012 to 14/08/2012. We also counted the number of articles in different categories from 02/08/2012 to 14/08/2012 (Table 5.2-5-11).

User Id	14541	14570	14751	15047	15196	15786	15850	16025	17047	46261
News/Canada	7	22	36	18	36	28	12	19	45	14
News/Health	2	3	0	0	3	4	8	0	0	1
News/World	5	12	26	6	19	5	8	21	24	11
News/Politic	4	7	18	8	52	52	4	1	2	5
News/Technology	0	4	2	4	2	1	6	5	1	14
News/Arts	3	0	1	0	3	1	2	0	1	0
News/Business	2	1	24	3	1	5	4	9	18	5
News/Offbeat	1	0	2	1	2	3	5	2	0	0
Hockey	2	0	2	1	0	0	0	0	0	0
Football	2	0	3	9	0	0	1	0	0	0
Figure Skating	1	0	0	0	0	0	0	0	0	0
Soccer	27	2	0	0	0	0	0	0	0	0
Baseball	0	0	0	0	0	0	0	0	0	0
Basketball	0	0	2	0	0	0	0	0	0	0
Total # of visited articles	56	51	116	50	118	99	50	57	91	50

Figure 5.1: This table shows number of articles that each user posted comments about in different categories.

In real situations our system will recommend articles to users on a daily basis. But here we combined the results of 12 different days for each of the algorithms (category-based and keywords-based) to describe our system performance. The numbers of articles that can be recommended to each user vary based on the categories and keywords of the articles that were released on that day. But we will not recommend more than 10 articles per day. So whenever the number of recommended articles is more than 10 we will recommend the top 10 most interesting articles to the user based on the probability of user interest in each category.

As we mentioned in chapter 4, users might ask for recommendations based on category or based on keywords, or she/he might choose a combination of both algorithms (50 percent of the results of both algorithms). In that case after computing the results for both algorithms and keeping one set of those results that are identical, we sort them based on user interest. Then we will use half of the most interesting articles of each set of results and recommend them to the user. For example if the result of category-based is 12 articles and the result of keywords-based algorithm is 8 articles, first we check for identical articles. If we find any, we will ignore one of them (those that belong to keywords-based algorithm results). Then we should sort them based on user interest probability and then we will choose 6 articles from the category-based results and 4 articles from the keywords-based results and demonstrate them in the form of links to the user.

5.3.1 Running the algorithms separately

We ran each of the algorithms for our 10 selected users (test data are the articles from 02/08/2012 to 14/08/2012) and then found the category of each article in the results sets. In Table 5.2 we demonstrate a number of recommended articles in each category and then calculate precision and recall for each algorithm based on these data. The precision score of the category-based algorithm for user 14541 is 1.0, which means all the articles that are recommended are relevant to the categories of user interests. The recall score in this algorithm is close to 1.0, which means almost the entire existing articles that were relevant to the user's interests were recommended, with the exception of 6 articles in Football and 3 articles about Offbeat that are missing. This means the recall is:

$$112/121 = 0.93 \tag{5.1}$$

User Id	14541	Recommended articles (category based)	Articles from 02/08 - 14/08		
News/Canada	7	30	30		
News/Health	2	10	10		
News/World	5	11	11		
News/Politic	4	8	8		
News/Technology	0	0	9		
News/Arts	3	2	2		
News/Business	2	6	6	Precision	1.00
News/Offbeat	1	3	6	Recall	0.93
Hockey	2	13	13		
Football	2	15	21		
Figure Skating	1	0	0		
Soccer	27	14	14		
Baseball	0	0	12		
Basketball	0	0	11		
Total # of visited articles	56	112	153		
User Id	14541	Recommended articles (keywords based)	Articles from 02/08 - 14/08		
News/Canada	7	0	30		
News/Health	2	0	10		
News/World	5	0	11		
News/Politic	4	0	8		
News/Technology	0	0	9		
News/Arts	3	0	2		
News/Business	2	0	6	Precision	0.89
News/Offbeat	1	0	6	Recall	0.07
Hockey	2	1	13		
Football	2	0	21		
Figure Skating	1	0	0		
Soccer	27	7	14		
Baseball	0	0	12		
Basketball	0	1	11		
Total # of visited articles	56	9	153		

Figure 5.2: Information related to user 14541, first table displays the result of category- based algorithm and second table displays the results of keywords-based algorithm, Also note that the last column in this tables and rest of the tables in this chapter contains the total number of the articles that have been visited by all ten users.

Because the recall of the category-based algorithm is close to 1.0, most of the articles that are recommended by the keywords-based algorithm are duplicates of category-based results. We checked for unique results from the keywords-based algorithms and we found that one of the articles about soccer was not in category-based results.

This user has not read any article about basketball but in the second table one article about basketball is recommended. That is because the user was interested in articles related to the Olympics and this basketball article also is related to Olympics. The result of precision, which is calculated for the keywords-based algorithm, is less than 1.0, because we computed the precision based on user interests, which also were classified in different categories. But if we look at user interests in more detail (checking user keywords) we might say that the precision is 1.0. But in this report for the sake of simplicity we calculate precision and recall based on categories of user interests, which is not very detailed. On the other hand the keywords based algorithm cannot recall all the related articles because, if it does not find any match in the user's keywords and articles keywords, it concludes that the user is not interested in the article, which might not be correct. As a result most days this algorithm cannot recommend anything to the user. For example the keywords-based algorithm recommended one article on 09/08/2012, five articles on 10/08/2012, three articles on 11/08/2012 and nothing for the other days (within our testing period). Note that those articles will be recommended if one of their keywords matches the user's keywords more than two times. In other words, that keyword was repeated in the user's keywords list more than twice. If we change this condition to more than one the algorithm will recommend more articles but not enough to make the recall score

1.0 or even close to 1.0. Also, many of these new recommended articles can be found in the results of the other algorithm. In Tables 5.9, 5.10, and 5.11 we have shown the results of the second condition (if there is one or more matches in keywords) as an example.

User Id	14570	Recommended articles (category based)	Articles from 02/08 - 14/08		
News/Canada	22	30	30		
News/Health	3	10	10		
News/World	12	11	11		
News/Politic	7	8	8		
News/Technology	4	9	9		
News/Arts	0	0	2		
News/Business	1	6	6	Precision	1.00
News/Offbeat	0	0	6	Recall	0.98
Hockey	0	0	13		
Football	0	0	21		
Figure Skating	0	0	0		
Soccer	2	12	14		
Baseball	0	0	12		
Basketball	0	0	11		
Total # of visited articles	51	86	153		
User Id	14570	Recommended articles (keywords based)	Articles from 02/08 - 14/08		
News/Canada	22	3	30		
News/Health	3	3	10		
News/World	12	0	11		
News/Politic	7	2	8		
News/Technology	4	0	9		
News/Arts	0	0	2		
News/Business	1	0	6	Precision	0.90
News/Offbeat	0	0	6	Recall	0.10
Hockey	0	1	13		
Football	0	0	21		
Figure Skating	0	0	0		
Soccer	2	1	14		
Baseball	0	0	12		
Basketball	0		11		
Total # of visited articles	51	10	153		

Figure 5.3: Information related to user 14570.

In Table 5.3 again we checked for unique results of the algorithm and we found out that articles about soccer and hockey are not identical to those in the first table. According to the results this user has no interest in hockey based on his/her visited articles but keywords-based algorithm recommend an article about hockey to this user because one or some of the keywords were repeated in user keywords list more than twice. In Table 5.4 second part, 4 articles were recommended by the keywords-based algorithm, that are not in users interest categories. But apparently their keywords were repeated in user keywords list more than twice. This shows although the keywords-based algorithm might not be able to recommend articles to the user every day and some of its recommendation might be duplication of category-based algorithm, but it is capable of recommending articles from new categories to the

user, which might be actually interesting for the user. But because of the mentioned weaknesses of this algorithm it should be combined with another algorithm like the category-based algorithm in a recommender system.

User Id	14751	Recommended articles (category based)	Articles from 02/08 - 14/08		
News/Canada	36	30	30		
News/Health	0	0	10		
News/World	26	11	11		
News/Politic	18	8	8		
News/Technology	2	9	9		
News/Arts	1	2	2		
News/Business	24	6	6	Precision	1.00
News/Offbeat	2	6	6	Recall	0.92
Hockey	2	13	13		
Football	3	15	21		
Figure Skating	0	0	0		
Soccer	0	0	14		
Baseball	0	0	12		
Basketball	2	8	11		
Total # of visited articles	116	108	153		
User Id	14751	Recommended articles (keywords based)	Articles from 02/08 - 14/08		
News/Canada	36	2	30		
News/Health	0	2	10		
News/World	26	1	11		
News/Politic	18	0	8		
News/Technology	2	0	9		
News/Arts	1	0	2		
News/Business	24	0	6	Precision	0.44
News/Offbeat	2	0	6	Recall	0.08
Hockey	2	1	13		
Football	3	0	21		
Figure Skating	0	0	0		
Soccer	0	2	14		
Baseball	0	0	12		
Basketball	2	1	11		
Total # of visited articles	116	9	153		

Figure 5.4: Information related to user 14751.

In Table 5.5 articles about soccer and health are recommended articles from new categories to the user.

User Id	15047	Recommended articles (category based)	Articles from 02/08 - 14/08		
News/Canada	18	30	30		
News/Health	0	0	10		
News/World	6	11	11		
News/Politic	8	8	8		
News/Technology	4	9	9		
News/Arts	0	0	2		
News/Business	3	6	6	Precision	1.00
News/Offbeat	1	6	6	Recall	0.82
Hockey	1	0	13		
Football	9	15	21		
Figure Skating	0	0	0		
Soccer	0	0	14		
Baseball	0	0	12		
Basketball	0	0	11		
Total # of visited articles	50	85	153		
User Id	15047	Recommended articles (keywords based)	Articles from 02/08 - 14/08		
News/Canada	18	1	30		
News/Health	0	1	10		
News/World	6	0	11		
News/Politic	8	2	8		
News/Technology	4	0	9		
News/Arts	0	0	2		
News/Business	3	0	6	Precision	0.44
News/Offbeat	1	0	6	Recall	0.08
Hockey	1	1	13		
Football	9	4	21		
Figure Skating	0	0	0		
Soccer	0	1	14		
Baseball	0	0	12		
Basketball	0	0	11		
Total # of visited articles	50	10	153		

Figure 5.5: Information related to user 15047.

In Table 5.6 the only new article that is recommended to the user, is about soccer, although the user has not read any of the articles in that category before.

User Id	15196	Recommended articles (category based)	Articles from 02/08 - 14/08		
News/Canada	36	30	30		
News/Health	3	10	10		
News/World	19	11	11		
News/Politic	52	8	8		
News/Technology	2	9	9		
News/Arts	3	2	2		
News/Business	1	6	6	Precision	1.00
News/Offbeat	2	6	6	Recall	1.00
Hockey	0	0	13		
Football	0	0	21		
Figure Skating	0	0	0		
Soccer	0	0	14		
Baseball	0	0	12		
Basketball	0	0	11		
Total # of visited articles	118	82	153		
User Id	15196	Recommended articles (keywords based)	Articles from 02/08 - 14/08		
News/Canada	36	2	30		
News/Health	3	3	10		
News/World	19	0	11		
News/Politic	52	2	8		
News/Technology	2	0	9		
News/Arts	3	0	2		
News/Business	1	0	6	Precision	0.89
News/Offbeat	2	1	6	Recall	0.11
Hockey	0	0	13		
Football	0	0	21		
Figure Skating	0	0	0		
Soccer	0	1	14		
Baseball	0	0	12		
Basketball	0	0	11		
Total # of visited articles	118	9	153		

Figure 5.6: Information related to user 15196.

In Table 5.7 new articles from soccer and basketball categories are recommended to the user by the keywords-based algorithm, while the algorithm only recommend a new article about soccer to the user in Table 5.8.

User Id	15786	Recommended articles (category based)	Articles from 02/08 - 14/08		
News/Canada	28	30	30		
News/Health	4	10	10		
News/World	5	11	11		
News/Politic	52	8	8		
News/Technology	1	9	9		
News/Arts	1	2	2		
News/Business	5	6	6	Precision	1.00
News/Offbeat	3	6	6	Recall	1.00
Hockey	0	0	13		
Football	0	0	21		
Figure Skating	0	0	0		
Soccer	0	0	14		
Baseball	0	0	12		
Basketball	0	0	11		
Total # of visited articles	99	82	153		
User Id	15786	Recommended articles (keywords based)	Articles from 02/08 - 14/08		
News/Canada	28	3	30		
News/Health	4	2	10		
News/World	5	0	11		
News/Politic	52	1	8		
News/Technology	1	0	9		
News/Arts	1	0	2		
News/Business	5	0	6	Precision	0.70
News/Offbeat	3	1	6	Recall	0.09
Hockey	0	0	13		
Football	0	0	21		
Figure Skating	0	0	0		
Soccer	0	2	14		
Baseball	0	0	12		
Basketball	0	1	11		
Total # of visited articles	99	10	153		

Figure 5.7: Information related to user 15786.

User Id	15850	Recommended articles (category based)	Articles from 02/08 - 14/08		
News/Canada	12	30	30		
News/Health	8	10	10		
News/World	8	11	11		
News/Politic	4	8	8		
News/Technology	6	9	9		
News/Arts	2	2	2		
News/Business	4	6	6	Precision	1.00
News/Offbeat	5	6	6	Recall	1.00
Hockey	0	0	13		
Football	1	21	21		
Figure Skating	0	0	0		
Soccer	0	0	14		
Baseball	0	0	12		
Basketball	0	0	11		
Total # of visited articles	50	103	153		
User Id	15850	Recommended articles (keywords based)	Articles from 02/08 - 14/08		
News/Canada	12	0	30		
News/Health	8	0	10		
News/World	8	0	11		
News/Politic	4	0	8		
News/Technology	6	0	9		
News/Arts	2	0	2		
News/Business	4	1	6	Precision	0.25
News/Offbeat	5	0	6	Recall	0.01
Hockey	0	2	13		
Football	1	0	21		
Figure Skating	0	0	0		
Soccer	0	1	14		
Baseball	0	0	12		
Basketball	0	0	11		
Total # of visited articles	50	4	153		

Figure 5.8: Information related to user 15850.

As we mentioned earlier in Tables 5.9, 5.10 and 5.11, we have run the keywords-based algorithm with the condition of finding more than one matching keyword. As a result the algorithm recommended many more articles to users (almost 3 times, see

Table 5.9 and 5.10). It is true that many of them are just duplicates of what we had in the category-based result set, but also we have more new articles as well (two more about health and one more about soccer; also we have one article about baseball, which was not recommended with the first condition). Also in Table 5.10 all the new articles are recommended based on the second condition (two articles about soccer, two articles about basketball, two articles about hockey, one about health and three about offbeat) and all of the results of the first condition were duplications.

User Id	16025	Recommended articles (category based)	Articles from 02/08 - 14/08		
News/Canada	19	30	30		
News/Health	0	0	10		
News/World	21	11	11		
News/Politic	1	8	8		
News/Technology	5	9	9		
News/Arts	0	0	2		
News/Business	9	6	6	Precision	1.00
News/Offbeat	2	6	6	Recall	1.00
Hockey	0	0	13		
Football	0	0	21		
Figure Skating	0	0	0		
Soccer	0	0	14		
Baseball	0	0	12		
Basketball	0	0	11		
Total # of visited articles	57	70	153		
User Id	16025	Recommended articles (keywords based)	Articles from 02/08 - 14/08		
News/Canada	19	9	30		
News/Health	0	3	10		
News/World	21	5	11		
News/Politic	1	2	8		
News/Technology	5	6	9		
News/Arts	0	0	2		
News/Business	9	1	6	Precision	0.75
News/Offbeat	2	1	6	Recall	0.46
Hockey	0	2	13		
Football	0	0	21		
Figure Skating	0	0	0		
Soccer	0	2	14		
Baseball	0	1	12		
Basketball	0	0	11		
Total # of visited articles	57	32	153		

Figure 5.9: Information related to user 16025.

User Id	17047	Recommended articles (category based)	Articles from 02/08 - 14/08		
News/Canada	45	30	30		
News/Health	0	0	10		
News/World	24	11	11		
News/Politic	2	8	8		
News/Technology	1	9	9		
News/Arts	1	2	2		
News/Business	18	6	6	Precision	1.00
News/Offbeat	0	0	6	Recall	1.00
Hockey	0	0	13		
Football	0	0	21		
Figure Skating	0	0	0		
Soccer	0	0	14		
Baseball	0	0	12		
Basketball	0	0	11		
Total # of visited articles	91	66	153		
User Id	17047	Recommended articles (keywords based)	Articles from 02/08 - 14/08		
News/Canada	45	12	30		
News/Health	0	1	10		
News/World	24	5	11		
News/Politic	2	4	8		
News/Technology	1	3	9		
News/Arts	1	0	2		
News/Business	18	2	6	Precision	0.75
News/Offbeat	0	3	6	Recall	0.55
Hockey	0	2	13		
Football	0	0	21		
Figure Skating	0	0	0		
Soccer	0	2	14		
Baseball	0	0	12		
Basketball	0	2	11		
Total # of visited articles	91	36	153		

Figure 5.10: Information related to user 17047.

User Id	46261	Recommended articles (category based)	Articles from 02/08 - 14/08		
News/Canada	14	30	30		
News/Health	1	10	10		
News/World	11	11	11		
News/Politic	5	8	8		
News/Technology	14	9	9		
News/Arts	0	0	2		
News/Business	5	6	6	Precision	1.00
News/Offbeat	0	0	6	Recall	1.00
Hockey	0	0	13		
Football	0	0	21		
Figure Skating	0	0	0		
Soccer	0	0	14		
Baseball	0	0	12		
Basketball	0	0	11		
Total # of visited articles	50	74	153		
User Id	46261	Recommended articles (keywords based)	Articles from 02/08 - 14/08		
News/Canada	14	1	30		
News/Health	1	1	10		
News/World	11	2	11		
News/Politic	5	4	8		
News/Technology	14	2	9		
News/Arts	0	0	2		
News/Business	5	0	6	Precision	0.71
News/Offbeat	0	1	6	Recall	0.14
Hockey	0	2	13		
Football	0	0	21		
Figure Skating	0	0	0		
Soccer	0	1	14		
Baseball	0	0	12		
Basketball	0	0	11		
Total # of visited articles	50	14	153		

Figure 5.11: Information related to user 46261.

5.3.2 Combination of algorithms results

In order to show the performance of the combination of category-based and keywords- based algorithms, we tested our system on a specific date (10/08/2012) for two different users and the result are demonstrated in Tables 5.12 and 5.13. In these two examples our category-based algorithm selects all the possible articles on that date to be recommended to the user so precision and recall scores are 1.0 the articles that are chosen by keywords-based algorithm is the same as the category-based algorithm (Table 5.12), or contain something new (Table 5.13). In both cases because we are going to use the combination of both results, we can consider precision and recall score to be 1.0. Because all the articles that are related to user's interests are retrieved and all the articles that are retrieved are related to user's interest.

Previously we said that we would ignore one set of the identical articles and then sort them based on user interest. In that case it is probable that the new articles that are recommended by the keywords-based algorithm will never be recommended to the user (because they are going to be sorted as the least favorite of the user) unless there is a small number of articles that are recommended by the keywords-based algorithm and some of them are similar to the category-based algorithm result or all of them are new articles, just like our second example.

User Id	14541	Recommended articles (category based)	Recommended articles (keywords based)	Combination	Articles on 10/08/2012
News/Canada	7	2	0	2	2
News/Health	2	2	0	0	2
News/World	5	0	0	0	0
News/Politic	4	2	0	2	2
News/Technology	0	0	0	0	1
News/Arts	3	0	0	0	0
News/Business	2	0	0	0	0
News/Offbeat	1	1	0	0	1
Hockey	2	1	0	0	1
Football	2	4	0	1	4
Figure Skating	1	0	0	0	0
Soccer	27	5	5	5	5
Baseball	0	0	0	0	1
Basketball	0	0	0	0	1
Total # of visited articles	56	17	5	10	20
	Precision	1.00			
	Recall	1.00			

Figure 5.12: Results related to the combination of both algorithms for user 14541.

User Id	16025	Recommended articles (category based)	Recommended articles (keywords based)	Combination	Articles on 10/08/2012
News/Canada	19	2	0	2	2
News/Health	0	0	1	1	2
News/World	21	0	0	0	0
News/Politic	1	2	0	2	2
News/Technology	5	1	0	1	1
News/Arts	0	0	0	0	0
News/Business	9	0	0	0	0
News/Offbeat	2	1	0	1	1
Hockey	0	0	0	0	1
Football	0	0	0	0	4
Figure Skating	0	0	0	0	0
Soccer	0	0	1	1	5
Baseball	0	0	0	0	1
Basketball	0	0	0	0	1
Total # of visited articles	57	6	2	8	20
		Precision	1.00		
		Recall	1.00		

Figure 5.13: Results related to the combination of both algorithm for user 16025.

5.3.3 Precision and Recall of the algorithms

The following graphs demonstrate the precision and recall score of all 10 users for our algorithms. The keywords-based algorithm's precision and recall scores are much lower than for the category-based algorithm (especially in relation to recall scores), which was expected because the goal of the keywords-based algorithm is to introduce users to the articles that although they relate to subjects that they do not read regularly (or they have never read), might be interesting for them.

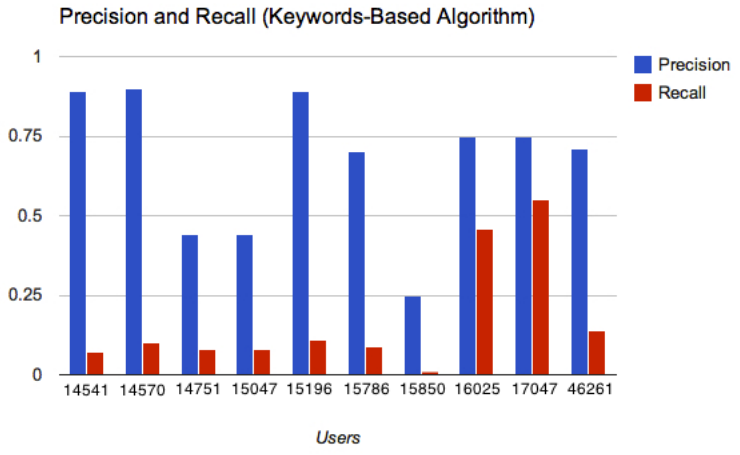

Figure 5.14: Precision and recall of 10 users (Keywords-based Algorithm)

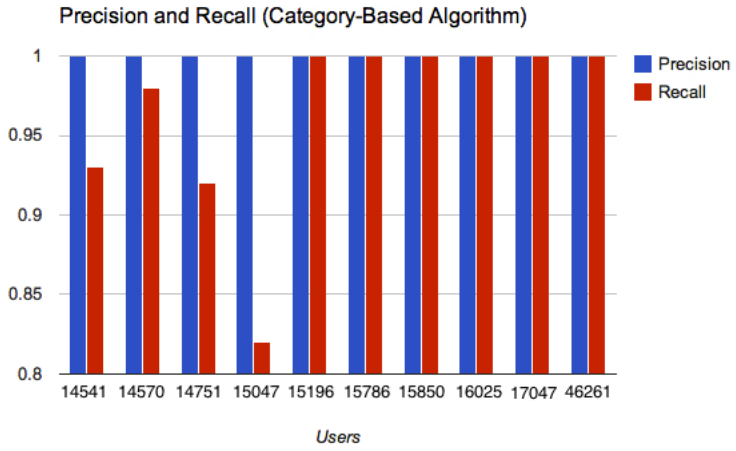

Figure 5.15: Precision and recall of 10 users (category-Based Algorithm).

5.4 Discussion

Based on the results of the experiments, the category-based algorithm recommends to the user almost all the possible relevant articles. However, the algorithm cannot introduce new articles from new categories that might be interesting for users. On the other hand, we can cannot entirely rely on the keywords-based algorithm, because this algorithm has difficulty in recommending all the relevant articles. So using these two algorithms together could be a very good choice.

Chapter 6

Conclusion and Future Work

Content-based recommender systems employ users' past activities to predict interests in order to suggest new items to them. This method of filtering is popular in news recommendation and has been widely used. We aimed to create a content-based recommender system for a news website like CBC News that performs reasonably with a small amount of data. The system can be used with other similar websites as well. Furthermore, instead of user clicks on articles, we considered user comments as proof of user interest in a specific subject that is stronger evidence. Moreover, we combined this category-based algorithm with another algorithm that attempts to match user keywords and article keywords. We called this the keyword-based algorithm. We combined these two algorithms to meet different users' needs.

Our experiments show that the category-based algorithm is capable of recommending almost all of the possible relevant articles to the user but cannot introduce users to new articles from new categories that might be interesting for them. On the other hand, our recommender system cannot only rely on the keywords-based algorithm. This algorithm has difficulty in recommending all of the relevant articles, however it is capable of introducing our users to new articles from new categories. Thus using these two algorithms together was found to be a very good choice, based

on experiment's results.

As we discussed through this report there are many techniques that can be used to implement a recommender system. To extend and improve our work we can combine our content-based algorithms with a collaborative algorithm. Also, in the case of the keywords-based algorithm it might be a good idea to identify the synonyms and consider them in the algorithm. Identifying synonyms will help to find more matches for each keyword so the recommendation will become more accurate (having more evidence gives us stronger proof). Further, it might be interesting to add news trend information to our results. News trend or hot news are those articles that form short-term user interests. For example, during the Olympic games many users appeared interested in articles related to the games and events about this topic. This is despite the fact that they might have no history related to any kind of sports in our database. Some of these users might not be interested in reading about sports after the Olympics. So, it is important to determine the articles that are related to news trends and suggest them to users.

Bibliography

[1] L. Stead, M. Rosenstein, and G. Furnas, W. Hill, *Recommending and Evaluating Choices in a Virtual Community of Use*, in Proc. Conf. Human Factors in Computing Systems, 1995.

[2] N. Iakovou, M. Sushak, P. Bergstrom, and J. Riedl, P. Resnick, *GroupLens: An Open Architecture for Collaborative Filtering of Netnews*, in Computer Supported Cooperative Work Conf, 1994

[3] R. Burke. *Hybrid recommender systems: Survey and experiments. User Modeling and User-Adapted Interaction*, pages 331 370, November 2002.

[4] Ben Schafer, Joseph Konstan, and John Riedi. *Recommender systems in e-commerce.*, In Proceedings of the 1st ACM conference on Electronic commerce, EC 99, pages 158 166, New York, NY, USA, 1999. ACM.

[5] M. Balabanovi and Y. Shoham. *Fab: content-based, collaborative recommendation*, Communications of the ACM, pages 66 72, 1997.

[6] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*, Addison-Wesley, Ed., 1999.

[7] N. Belkin and B. Croft, *Information Filtering and Information Retrieval*, Comm. ACM, vol. 35, no. 12, pp. 29-37, 1992.

[8] Michael J. Pazzani, and Daniel Billsus, *Content-Based Recommendation Systems*, pp. 325-341, the Adaptive Web, Peter Brusilovsky, Alfred Kobsa, And Wolfgang Nejdl (Ed.), Lecture Notes in Computer Science, Springer- Verlag, Berlin, Germany, Vol. 4321, May 2007, ISBN 978-3-540-72078-2.

[9] G. Adomavicius and A. Tuzhilin, *Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions*, Knowledge and Data Engineering, IEEE Transactions on, vol. 17, no. 6, pp. 734 - 749, june 2005.

[10] J. Liu, P. Dolan and Elin R. Pedersen, *Personalized news recommendation based on click behavior*, in Proceedings of the 15th international conference on Intelligent user interfaces, New York, NY, USA, 2010, pp. 31 40.

[11] Cohen, W., Hirsh, H., *Joins that Generalize: Text Classification Using* WHIRL In: Proceed- ings of the Fourth International Conference on Knowledge Discovery Data Mining, New York, NY (1998) 169 -173

[12] Yang, Y. *An Evaluation of Statistical Approaches to Text Categorization Information Retrieval* 1(1) (1999) 67-88

[13] K.Lang, *Newsweeder:Learning to Filter Netnews,in 12th int'l Conf. Machine Learning, 1995.*

[14] R.J. Mooney and L. Roy, Content-Based Book Recommending Using Learning for Text Categorization*, ACM SIGIR '99 Workshop Recommender System: Algorithms and Evaluation, 1999.*

[15] M. Pazzani and D. Billsus, Learning and Revising User Profiles: The Identification of Interesting Web Sites*, Machine Learning, vol. 27, pp. 313-331, 1997.*

[16] R. Di Massa, M. Montagnuolo and A. Messina, Implicit news recommendation based on user interest models and multimodal content analysis*, in Proceedings*

of the 3rd international workshop on Automated information extraction in media production, New York, NY, USA, 2010, pp. 33-38.

[17] Stuart E. Middleton, Nigel R. Shadbolt, David C. De Roure, Capturing interest through inference and visualization: ontological user profiling in recommender systems*, In K-CAP '03: Proceedings of the 2nd international conference on Knowledge capture (2003), pp. 62-69.*

[18] Stuart E. Middleton and David C. De Roure and Nigel R. Shadbolt, Capturing Knowledge of User Preferences: Ontologies in Recommender Systems*, In Proceedings of the First International Conference on Knowledge Capture (K-CAP 2001), Oct 2001, pp. 100-107, ACM Press*

[19] Toine Bogers, Antal van den Bosch, Comparing and evaluating information retrieval algorithms for news recommendation*, In Proceedings of the 2007 ACM conference on Recommender systems (2007), pp. 141-144.*

[20] Pattie Maes, Agents that reduce work and information overload Commun.*, ACM, Vol. 37, No. 7. (July 1994), pp. 30-40.*

[21] Ah-Hwee Tan, C. Teo, Learning user profiles for personalized information dissemination*, Neural Networks Proceedings, 1998. IEEE World Congress on Computational Intelligence. The 1998 IEEE International Joint Conference on, Vol. 1 (May 1998), pp. 183-188 vol.1.*

[22] Kazunari Sugiyama, Kenji Hatano, Masatoshi Yoshikawa, Adaptive web search based on user profile constructed without any effort from users*, In Proceedings of the 13th international conference on World Wide Web (2004), pp. 675-684.*

[23] Steve Wedig, Omid Madani, A large-scale analysis of query logs for assessing personalization opportunities*, In KDD '06: Proceedings of the 12th ACM*

SIGKDD international conference on Knowledge discovery and data mining (2006), pp. 742-747.

[24] $http://en.wikipedia.org/wiki/MySQL$

[25] $http://en.wikipedia.org/wiki/JavaServer_Pages$

[26] $http://en.wikipedia.org/wiki/Precision\_and\_recall$

[27] $http://www.cbc.ca/news/$

Vita

Candidate's full name: Mahta Moattari
University attended :
Master of Computer Science, University of New Brunswick, Fredericton, NB, Canada, January 2011- May 2013
Bachelor of Computer Science and Information Technology, Amirkabir University of Tech.(Transfer), Tehran, Iran, January 2006- August 2010
Bachelor of Computer Science and Information Technology, Eastern Mediterranean University, Famagusta, Cyprus, September 2004- January 2006
Conference Presentations:
Poster presentation about Content-Based Recommender System for News Sites, The 10th Annual Research Expo, April 12th 2013