International Journal on Artificial Intelligence Tools

PNFS: PERSONALIZED WEB NEWS FILTERING AND SUMMARIZATION

Xindong Wu

College of Computer Science and Information Engineering
Hefei University of Technology, Hefei, 230009, China
Department of Computer Science
University of Vermont, Burlington, VT 05405, USA
xwu@cs.uvm.edu

Fei Xie

Department of Computer Science and Technology
Hefei Normal University, Hefei, 230601, China
College of Computer Science and Information Engineering
Hefei University of Technology, Hefei, 230009, China
xiefei9815057@sina.com

Gongqing Wu

College of Computer Science and Information Engineering
Hefei University of Technology, Hefei, 230009, China
wugq@hfut.edu.cn

Wei Ding

Department of Computer Science
University of Massachusetts Boston, Boston, USA
ding@cs.umb.edu

Received (Day Month Year)
Revised (Day Month Year)
Accepted (Day Month Year)

Information on the World Wide Web is congested with large amounts of news contents. Recommending, filtering, and summarization of Web news have become hot topics of research in Web intelligence, aiming to find interesting news for users and give concise content for reading. This paper presents our research on developing the Personalized News Filtering and Summarization system (PNFS). An embedded learning component of PNFS induces a user interest model and recommends personalized news. Two Web news recommendation methods are proposed to keep tracking news and find topic interesting news for users. A keyword knowledge base is maintained and provides real-time updates to reflect the news topic information and the user's interest preferences. The non-news content irrelevant to the news Web page is filtered out. A keyword extraction method based on lexical chains is proposed that uses the semantic similarity and the relatedness degree to represent the semantic relations between words. Word sense disambiguation is also performed in the built lexical chains. Experiments on Web news pages and journal articles show that the proposed keyword extraction method is effective. An example run

of our PNFS system demonstrates the superiority of this Web intelligence system.

Keywords: Personalized News; Web News Filtering; Web News Summarization.

1. Introduction

Along with the rapid development of the World Wide Web, information on Web pages is rapidly inflated and congested with large amounts of news contents. To identify useful information that satisfies a user's interests, the filtering and summarization of personalized Web news have drawn much attention in Web intelligence. The filtering and summarization of personalized Web news refer to the recommendation, extraction, and summarization of interesting and useful information from Web pages, which can be widely used to promote the automation degree in public opinion investigation, intelligence gathering and monitoring, topic tracking, and employment services.

This paper presents a personalized news filtering and summarization (PNFS) system that works on news pages on the Web. The first task of our system is to recommend interesting news to users. We dynamically obtain Web news from the Google news website (http://news.google.com), and then recommend personalized news to the users according to their preferences. A news filter is applied in our system to provide high quality news content for analyzing. The second research component of the PNFS system is to summarize Web news. The summarization is given in the form of keywords based on lexical chains. Keywords offer a brief yet precise summary of the news content. Despite of the known advantages of keywords, only a minority of news Web pages have keywords assigned to them. This motivates our research in finding automated approaches to keyword extraction from Web news.

The main contributions of this paper are as follows. A Web news recommendation mechanism is provided according to the users' interests which makes our PNFS system specially designed for personalized news treatment. An embedded learning component interacts with the recommendation mechanism and models users' interests. A keyword knowledge base is stored to update the user's profile, and a keyword extraction algorithm is also provided to construct the lexical chains based on the word similarity and the relatedness degree to represent the semantic relations between words.

The rest of the paper is organized as follows. Section II reviews related work on personalized Web news recommendation, content extraction of Web news, and keyword extraction. The PNFS system architecture is given in Section III. Section IV introduces the proposed method of personalized Web news recommendation. Section V presents our algorithm for keyword extraction based on semantic relations and the experimental results. Section VI demonstrates an example run of the PNFS system. Finally, Section VII concludes the paper and discusses our future work.

2. Related Work

2.1. *Recommender Systems*

There are mainly three different techniques commonly used in recommendation systems: content-based recommendation, collaborative filtering, and hybrid recommendation.

The content-based approach recommends items based on the profile that is built by analyzing the content of articles that a user has read in the past. Syskill and Webert aimed to rate pages on the Word Wide Web and recommend them to a user by analyzing the content on each page.[1] Tan and Teo proposed a personalized news system where the profile is defined initially by a user and then learned from the user's feedback using neural networks.[2] The collaborative filtering approach uses the known preferences of a group of users to make recommendation for other users. Group-Lens[3] is a personalized news system using the collaborative filtering approach. Das *et al.* applied collaborative filtering techniques to Google news.[4] Yacut and Polat investigated how to produce high-quality referrals on hybrid collaborative filtering approaches from cross distributed data while maintaining the privacy.[5] Hybrid approaches combine content-based methods with collaborative filtering techniques, aiming to avoid the limitations of each approach and improve the recommendation performance.[6]

2.2. *Web News Extraction*

Web information extraction can be traced back to the integration research of heterogeneous data sources of structured and semi-structured data. A wrapper is viewed as a component in an information integration system to encapsulate accessing operations of multiple heterogeneous data sources, with which users can query on the integration system using a single uniform interface. As information extraction is the key function in a wrapper, the terms extractors and wrappers are often used interchangeably.

The targets of Web information extraction can be classified into three categories: records in a Web page, specific interesting attributes, and the main content of the page. Most Web information exploration systems for extracting records in a Web page work by automatically discovering record boundaries and then dividing them into items. With the rapid development of search engines and Web intelligence collection and analysis, the research of extracting specific interesting attributes, such as Web news titles and the main content of Web news from a Web page, has received much attention.[7]

Most Web information exploration systems use extraction rules that are represented as regular grammars, first order logic or a tag tree, with features as delimiter-based constraints. Those features include HTML tags, literal words, DOM tree paths, part-of-speech taggers, Word-Net semantic classes, tokens' lengths, link grammars, etc. W4F[8] uses DOM tree paths to address a Web page. The data to be

extracted is often collocated in the same path of the DOM tree, and it is convenient to address data with DOM tree paths, which make the rule processing much easier. Chakrabarti *et al.* took an extractive approach for title generation, which starts with URL tokens, HTML titles, keywords, and anchor text on incoming links etc.[9] Their approach combines information from external sources, and performs probabilistic parameter learning with a URL's HTML title, context/abstract, and vocabulary at the source level.

Wu *et al.* presented a news filtering and summarization (NFAS) system that works on Web pages.[10] The NFAS system consists of two main tasks. Given a URL from an end user or an application, the first task is to accurately identify whether the Web page is news or not, and if so filter the noise of the Web news, such as advertisements and non-relevant pictures. The second task is to summarize the Web news once it has been identified as a valid news page and has been filtered. The summarization is given in the form of lexical chains, based on keywords. This paper is built on the NFAS system. Web news pages are dynamically obtained and recommended to the users by their clicking histories. The keywords are extracted not only for summarizing the Web news but also capturing the main topics of the news content that the users have read, hence the keyword extraction algorithm in NFAS is improved for PNFS.

2.3. *Keyword Extraction*

Research in keyword extraction began in early 1950's. Existing work can be categorized into two major approaches: supervised extraction and unsupervised extraction. Supervised methods view keyword extraction as a classification task, where labeled keywords are used to learn a model. This model is constructed using a set of features that capture the saliency of a word as a keyword. Turney designed a keyword extraction system GenEX based on C4.5.[11] Witten *et al.* used Naive Bayes to extract keywords, and designed the Kea system.[12] Supervised methods have some nice properties, for example, they can produce interpretable rules to explain the relations between features and keywords. However, they require a large amount of training data with known keywords. Furthermore, supervised methods are not very flexible because the training process on a specific domain tends to make the extraction adapt to that domain. Unsupervised keyword extraction removes the need for training data. Instead of trying to learn explicit features that contribute to the extraction of keywords, the unsupervised approach utilizes the structure of the text itself to extract keywords that depict the topic of the text. Mihalcea presented a graph-based ranking method to keyword extraction.[13] You *et al.* proposed a new candidate phrase generation method based on the core word expansion algorithm that greatly reduced the size of the candidate and introduced additional new features to improve the accuracy of the keyphrase extraction system.[14]

The study of Chinese keyword extraction began in recent years. Li *et al.* probed into keyword extraction using the Maximum Entropy model.[15] Because the param-

eter estimation of feature selection is not always accurate, their results had much room for improvement. Liu *et al.* mined a manually labeled keyword corpus which is from the People's Daily newspaper and attained the constructed rules for Chinese keyword extraction.[16] This approach needs a large number of labeled keywords. Suo *et al.*[17] presented a lexical-chain-based keyword extraction method for Chinese documents, and lexical chains were constructed based on the HowNet-based word semantic similarity.[18]. Word similarity is computed by HowNet, but the candidate words not in HowNet are filtered out in this approach.

In this paper, we present a new keyword extraction method for Web news based on semantic relations. In our method, semantic relations of the words not in HowNet are computed by a word co-occurrence model. Lexical chains are constructed to represent semantic relations and build semantic links between words.

2.3.1. *Lexical Chains*

Halliday and Hasan first defined the notion of cohesion as a device that sticks together different parts (i.e., words, sentences, and paragraphs) of the text to function as a whole.[19] Lexical chains are sequences of related words where the cohesion occurs among words. Morris and Hirst first introduced the concept of lexical chains to segment text. Later, lexical chains are used in many tasks, such as text retrieval and information extraction.[20]

The construction of lexical chains needs a thesaurus for determining relations between words. In this paper, we construct the lexical chains using the thesaurus-based word similarity and the word co-occurrence model.[21] Two thesauruses, including WordNet and HowNet, are respectively used to compute word similarity in English[22] and in Chinese[18]. The word co-occurrence model is adopted to solve the problem that is difficult to compute the semantic relations between words not in the thesaurus. Word co-occurrence is an important model based on statistics widely used in natural language processing that reflects the relatedness of the words in a document. The frequency of two words co-occurring in the same window unit (i.e., a sentence or a paragraph) can be computed without a thesaurus.

HowNet is a common-sense knowledge base that unveils inter-conceptual and inter-attribute relations of concepts as connoting in lexicons of the Chinese language and their English equivalents.[23] There are two important terms in HowNet: concept and sememe. A concept is the semantic description of phrases. Each phrase has several concepts. A concept is defined by a kind of knowledge representation language named sememe that is the smallest basic semantic unit.

Given two phrases W_1 and W_2, W_1 has n concepts, S_{11}, S_{12}, ..., S_{1n}, and W_2 has m concepts, S_{21}, S_{22}, ..., S_{2m}. The similarity between W_1 and W_2 is defined as follows:[18]

$$Sim(W_1, W_2) = \max_{i=1...n, j=1...m} Sim(s_{1i}, s_{2j}) \tag{1}$$

A concept is described by sememes. Sememe similarity is the basis of con-

cept similarity. Sememes in HowNet compose a hierarchical tree by the hypernym-hyponym relation. The semantic distance of the two sememes is defined as follows:

$$Sim(p_1, p_2) = \frac{\alpha}{d + \alpha} \tag{2}$$

where p_1 and p_2 represent two sememes, d is the length of p_1 and p_2 in the sememe hierarchical tree, and α is a parameter usually set to 0.5.

Since keywords are general notional words, only the similarities of notional words are considered in this paper. The concept descriptions of two notional words S_1 and S_2 comprise of four components: (1) first basic sememes of S_1 and S_2, with the similarity $Sim_1(S_1, S_2)$, (2) other basic sememes with the similarity $Sim_2(S_1, S_2)$, (3) relational sememes with the similarity $Sim_3(S_1, S_2)$, and (4) symbol sememes with the similarity $Sim_4(S_1, S_2)$. Then the similarity of the two notional words is defined as follows:

$$Sim(S_1, S_2) = \sum_{i=1}^{4} \beta_i \prod_{j=1}^{i} Sim_j(S_1, S_2) \tag{3}$$

where β_1, β_2, β_3, and β_4 are adjusted parameters that reflect the influences of the four similarity measures to the total similarity, and $\beta_1+\beta_2+\beta_3+\beta_4 = 1$. Because the first basic sememes of S_1 and S_2 describe the primary features of each concept, the value of β_1 is larger than the other three parameters. The description abilities of the first basic sememes, other basic sememes, relational sememes and symbol sememes that describe the same concept are in a decreasing order, hence $\beta_1 > \beta_2 > \beta_3 > \beta_4$. In our implementation, the values of $\beta 1$, β_2, β_3, and β_4 are usually set to 0.5, 0.2, 0.17 and 0.13 according to Ref. 18.

3. System Architecture

Figure 1 shows the PNFS system architecture. A new user is required to register with initially interesting topic categories or keywords. Once a registered user logs in, the system returns personalized Web news to the user. When the user clicks on his/her interesting news items, the recently browsing history is updated. The user can either browse the original news Web page or read the filtered news content with summarized keywords. A keyword model is maintained to store the topic-distinguished keywords and the keywords selected from the browsed news stories. The user can also modify the keyword model to improve the recommendation performance.The PNFS system consists of two phases.

Phase 1: Personalized Web News Filtering. There are two major tasks in the personalized news filtering phase. One is to filter out the news stories that are uninteresting to the user. Another is to filter out non-news parts on news Web pages. The personalized filtering subsystem has four components: a news aggregator, a news filter, a learning component, and a keyword knowledge base. The news aggregator automatically obtains content from news sources worldwide.[24] In this paper, we aggregate the world wide news from the Google News website. These news

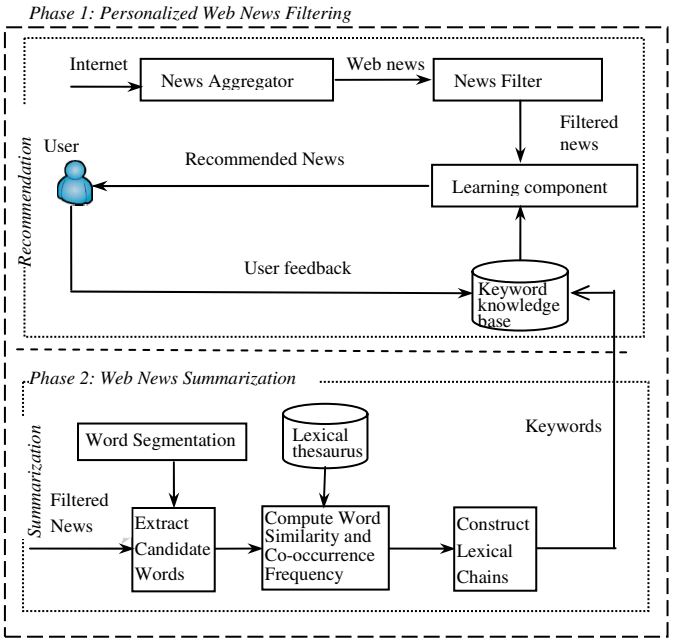


Fig. 1. The PNFS system architecture.

stories are automatically classified into different topic categories such as "world", "sports", "technology", and so on. Two learning algorithms including the k-nearest neighbor and the Naive Bayes are used to model the user's interest preference and recommend personalized news.

The keyword knowledge base stores two kinds of keywords including the general category keywords and the personalized interest keywords of a special user. The system periodically selects the category keywords for each news topic category from a large sample of stories. The selected category words are also used to represent the news story as a vector. The personalized keywords are selected from the browsed stories or designated when registering that reflect the topic preference of the user.

The news filter removes the non-news parts on the news Web page and provides higher quality content for recommendation and summarization than the original raw HTML Web page. This filtering stage is accomplished by the Web Information Extractor that retrieves the news Web page's title and news content by using pre-configured extraction rules. As with W4F[8], the PNFS system also adopts extraction rules based on the paths of the DOM tree of the news Web page. The Web Information Extractor uses extraction rules while it traverses the DOM tree of the Web page.

The learning component constantly learns the user interest model and recommends personalized news. There are two ways by which the learning component interacts with the recommendation system. One is by the user recently browsed

histories. The other is by the keywords that are automatically selected and can also be modified by the user.

Phase 2: Web News Summarization. The task of Phase 2 is to summarize and extract the keywords that capture the main topic of the news Web page. The purpose of keyword extraction is two-fold. First, it gives a concise form of the news to the user that saves the reading time. Second, the extracted keywords are also used to build a user interest model.

The filtered news content is segmented into words. Stop words are removed. Word frequencies are counted and the TFIDF[25] values are computed according to the corpus. Candidate words are identified by the TFIDF values. For the candidate words that occur in the thesaurus, word similarities are computed. Word co-occurrence frequencies are also calculated. Lexical chains are constructed by word similarities and word relatedness degree. Then keywords are extracted from the candidate words according to the TFIDF values and the semantic information in the lexical chains.

4. Personalized Web News Recommendation

4.1. *Recommendation Algorithms*

In this subsection, we present our proposed Web news recommendation algorithms. Firstly, we introduce a feature selection method to obtain the total word vocabulary. Then, we describe a recommendation algorithm to track the news events that the user would focus on based on the k-nearest neighbor algorithm. Finally, we provide a probability model to recommend the topic interesting news using the Naive Bayes algorithm. Many recommendation methods can be directly applied to Web news personalization. However, Web news has several characteristics including dynamic content, changing interests, multiple interests, novelty, and so on, that make some approaches better suited than other approaches.[26] Because collaborative methods suffer from the "latency" problem that needs some time to receive enough users' feedback, content-based approaches are better suited to the problem than collaborative approaches. In this paper, we focus on the content-based methods to recommend news by analyzing the user's browsing history.

We divide the recommendation news into two groups: previous news tracking and interesting topic news. The number of the recommended news stories for each group is defined by the user. We use the k-nearest neighbor algorithm[26] to track previously read news and find novel news as candidate news for the interesting topic news recommendation. The k-nearest neighbor algorithm identifies recently known stories that the user has read. It keeps tracking new stories that have the same event thread with recently read stories, and finds novel news.

After filtering out the non-news parts on the news page, each news article is converted to a vector using the Vector Space Model (VSM)[25]. In the VSM, document representation raises two issues: feature selection and term weighting. In this paper, an appropriate vocabulary is periodically selected from the recent news stories of

all categories as the word space. Algorithm 1 is the process of the feature selection. Firstly, for each document, the top k informative words are selected by the keyword extraction algorithm that will be given in Section V. Then, the topic words that appear frequently in the top m keyword lists are selected for each category. Finally, the vocabulary contains the topic words of all categories.

Algorithm 1 FeatureSelection(D, k, m, n)

Input: D: document set of all categories; k: the number of features selected for each document; m: the number of topic categories; n: the number of features selected for each topic category.

Output: F: selected feature set.

1: for each category C_i
2: for each document $d \in C_i$;
3: extract top k keywords from d;
4: sort all words in according to the number of times they appear in the top k keyword lists;
5: F_i=the n most frequent words in C_i
6: return $F = F_1 \cup F_2 \cup \ldots \cup F_m$;

In our PNFS system, the most recent 5000 documents per topic category are collected for the feature selection. The number of keywords extracted for each document is set to 50, and the number of features selected for each topic category is set to 1000. For example, the top 50 features selected for the technology topic category are as follows:

apple, company, google, iphone, user, mobile, vehicle, corp, software, app, video, technology, samsung, computer, internet, billion, safety, traffic, facebook, microsoft, credit, smartphone, service, ipad, version, price, web, office, honda, market, competitor, android, federal, gas, search, posted, major, executive, offline, site, incident, system, patent, security, youtube, device, model, industry, network, map

After feature selection, each news article is represented as a vector by the TFIDF term weighting scheme. The TFIDF value of term t_k in document d_j is defined as:

$$TFIDF(t_k, d_j) = TF(t_k, d_j) log \frac{N}{n_k} \tag{4}$$

where $TF(t_k, d_j)$ is the frequency of term t_k in d_j, N is the total number of documents in the corpus, and n_k is the number of documents in the corpus that contain term t_j. In order for the weight to fall in [0, 1], Eq.(4) is also normalized as:

$$w_{kj} = \frac{TFIDF(t_j, d_j)}{\sqrt{\sum_{s=1}^{|T|} TFIDF(t_s, d_j)^2}} \tag{5}$$

The cosine measure is used to compute the similarity of two vectors. Given two documents d_i and d_j, the cosine similarity between d_i and d_j is computed as:

$$Sim(d_i, d_j) = \frac{\sum_k w_{ki} w_{kj}}{\sqrt{\sum_k w_{ki}^2 \sum_k w_{kj}^2}} \tag{6}$$

In this paper, we define two similarity thresholds: t_1 and t_2 ($0 < t_1 < t_2 < 1$) to decide whether a news story is novel, interesting, or redundant. We calculate the similarities of the coming news story with the most recently rated stories and search k nearest neighbors. If one of the rated stories is closer than t_2, the coming story is labeled as redundant (the user has known it). If the average of the k similarities is less than t_1, the story is labeled as novel and selected as candidate news for the topic interesting news recommendation. If the average of the k similarities is larger than t_1 and less than t_2, the story is labeled as interesting; the larger the average of the similarities, the more interesting the story is. In our recommendation system, t_1 and t_2 are respectively set to 0.3 and 0.6, and k is 20 based on our empirical observations. Algorithm 2 describes the process of tracking previously read news and finding novel candidate news for recommendation.

Algorithm 2 Tracking_News(t_1, t_2, k, n)

Input: t_1, t_2: similarity thresholds; k: the number of nearest neighbors; n: the number of tracking news stories to recommend;
Output: the top n tracking news stories and the candidate news for topic interesting news recommendation.

```
for each upcoming news story do
    calculate the similarities of the news story with the user's recently
     read stories and get k most nearest neighbors;
    if one of the k similarities is larger than t2
        label the upcoming story as redundant;
        continue;
    if the average of the k similarities is larger than t1
        put the new story into the tracking news queue;
        continue;
    if the average of the k similarities is less than t1
        put the new story into the candidate news queue;
recommend the top n stories in the tracking news queue in the
   descending order of the average similarity.
```

Although the k-nearest neighbor algorithm performs well in tracking news events and finding novel news, the recommended news stories are too specific that do not reflect the diversity of the user interests. Therefore, we use another learning model, Naive Bayes[27] to calculate the probability of a news story being interesting. Each news story is represented as a feature-value vector, where features are the keywords selected from the news story, and feature values are the word frequencies. The user topic preference is also represented as a vector where keywords are selected from the total browsed stories. The Naive Bayes classifier is built to calculate the topic distributions of the user's interests. The recent news stories aggregated from the Google News website with topic category labels are trained. When the keyword list that reflects the user interests is input, the classifier outputs the probabilities of the keyword list belonging to the categories. The keyword list of the user is maintained

by collecting from the clicked documents or the designation when the user firstly logs in the system. The advantages of the keyword list are two fold. Firstly, the topic probabilities calculated by the keyword list can provide different proportions of candidate news stories for each category. The user is usually interested in a broad range of topic news that requires a user interest modeling approach must be capable of representing multiple topics of interest. The probability-based classifier is a good choice to address this issue. Secondly, the keyword list also provides a concise representation of the user's interest preferences.

There are two probability models for the implementation of the Naive Bayes classifier, the multi-variate Bernoulli model and the multinomial model.[28] In the multi-variate model, a document is represented as a binary feature vector, where each feature value is 0 or 1 respectively indicating the absence or presence of a word in the document. In the multinomial model, a document is represented as a vector of word occurrences. In our PNFS system, the multinomial probability model is adopted that takes advantage of the word frequency information.

We assume that all words of a document are independent of each other given a class. The probability of a document d belonging to class c is computed as:

$$p(c|d) \propto p(c) \prod_{1 \leq k \leq n_d} p(t_k|c) \tag{7}$$

where $p(c)$ is the priori probability of a document occurring in class c, $p(t_k|c)$ is the conditional probability of term t_k given class c, and n_d is the number of terms in d that are part of the vocabulary selected. Both $p(c)$ and $p(t_k|c)$ are calculated from the training documents.

$p(c)$ is estimated as:

$$p(c) = \frac{N_c}{N} \tag{8}$$

where N_c is the number of training documents in class c, and N is the total number of training documents.

$p(t_k|c)$ is estimated as:

$$p(t_k|c) = \frac{N_{ck} + 1}{\sum_{s=1}^{|V|} N_{cs} + |V|} \tag{9}$$

where N_{ck} is the number of occurrences of t_k in the training documents from class c, $|V|$ is the number of terms in the vocabulary, and the Laplace smoothing is used to avoid the probability being zero.

For each news story d or the user preference u, we can calculate the probability of the vector belonging to a given topic class according to the Naive Bayes classifier.

Proposition 1 Assume that user u is independent to the news document d given the news topic classification model, where m is the number of news topic categories. The probability that document d is recommended to user u is computed

as follows:

$$p(u|d) = p(u) \sum_{j=1}^{m} \frac{p(c_j|u)p(c_j|d)}{p(c_j)} \tag{10}$$

Proof: According to the conditional probability formula, $p(u|d) = p(u,d)/p_d$.
By the total probability theorem, $p(u,d) = \sum_{j=1}^{m} p(u,d|c_j)p(c_j)$.
Then, $p(u,d) = \sum_{j=1}^{n} \frac{p(u|c_j)P(d|c_j)p(c_j)}{p(d)}$.
Since $p(u|c_j)p(c_j) = p(u)p(c_j|u)$ and $p(d|c_j)/p(d) = p(c_j|d)/p(c_j)$,
$p(u|d) = p(u) \sum_{j=1}^{m} \frac{p(c_j|u)p(c_j|d)}{p(c_j)}$.
For a given user, $p(u)$ is a constant value, so we can recommend d to u as follows:

$$p(u|d) \propto \sum_{j=1}^{m} \frac{p(c_j|u)p(c_j|d)}{p(c_j)} \tag{11}$$

The interesting topic news stories of the user are recommended by Algorithm 3.

Algorithm 3 Interesting_Topic_News(D^T, D^C, D^A, n)

Input: topic training documents D^T with categories C; the user clicked documents D^C; candidate documents D^A; n: the number of topic interesting news stories to recommend;

Output: the top n interesting topic stories.

```
V=the vocabulary selected by Algorithm 1 from D^T;
for each d ∈ D^T;
    represent d as a vector through vocabulary V;
NB=the Naive Bayes classifier trained on D^T;
u=the keyword vector collected from D^C;
for each c ∈ C
    calculate the probability of u belonging to c through NB;
for each candidate document d ∈ D^A
    for each c ∈ C
        calculate the probability of d belonging to c through NB;
    calculate the score of d by Eq.(11);
recommend the top n stories according to their scores.
```

4.2. *Interaction of the Learning Component with the Recommendation System*

The evaluation of a recommendation system is a huge project that needs a long time to collect the users' data. This is a common drawback in the traditional recommendation systems. The learning model is modified only when the performance of the recommendation system is evaluated.

In the proposed PNFS system, the learning component is interactive with the overall system by the keyword knowledge and the user-click behaviors. Keywords extracted from the news stories are automatically added into the keyword knowledge

base, including the general keywords that distinguish different topic categories and the personalized keywords that reflect the user's long-term topic preference. The keyword model is also open to users. That means the user can not only add their own keywords but also remove the automatically generated keywords. The modified keyword model by the user will immediately cause the change of the recommendation results. Our keyword model has two advantages. First, the recommendation system will also work if the user is not willing to modify the user profile. Second, the performance of the system will be improved by the interaction with end users.

5. Keyword Extraction Based on Semantic Relations

5.1. *Keyword Extraction Algorithm*

In this subsection, we propose our keyword extraction algorithm based on lexical chains. A Lexical chain is a sequence of words with related senses. For example, $LC : ws_{21}, ws_{43}, ws_{72}$ is a lexical chain, in which ws_{ij} is the jth sense of the word ws_i. The interpretation is composed of several disjoint lexical chains. All the possible interpretations form the interpretation space. The interpretation with the largest cohesion value represents the correct senses of the words in the text. The cohesion value of a lexical chain is defined as the sum of similarities between the words in the lexical chain.

Figure 2 gives an example of resolving word sense ambiguity. I_1 and I_2 are two interpretations in which each node represents a word with a particular sense. If the similarity value between two words is larger than a threshold value, then there is an edge connecting the two words. The weight of the edge is the similarity value.

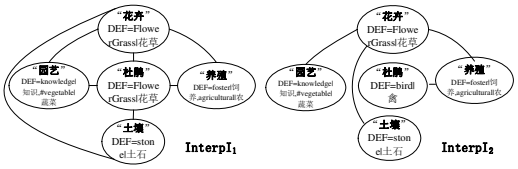


Fig. 2. An example of resolving word sense ambiguity.

For the words not in the thesaurus, we use the word co-occurrence model to compute the semantic relatedness degree between two words. There are two main measures to compute the word relatedness degree in the domain of information retrieval. One is the Dice coefficient.[29] Let x and y be two basic events in the probability space, representing the occurrences of words in a document. The Dice coefficient is defined as:

$$Dice(x, y) = \frac{2 \times p(x, y)}{p(x) + p(y)} \tag{12}$$

where $p(x, y)$, $p(x)$, and $p(y)$ are the joint and marginal probabilities of x and y.

Let $f(x)$ be the frequency of occurrences of x, that is, the number of sentences containing x, and $f(x, y)$ be the number of sentences containing both x and y. The Dice coefficient also equals $\frac{2\times f(x,y)}{f(x)+f(y)}$.

Another widely used measure is the mutual information.[30] It is defined as:

$$MI(x, y) = log\frac{p(x, y)}{p(x) \times p(y)} = log\frac{f(x, y) \times f_{total}}{f(x) \times f(y)} \tag{13}$$

where f_{total} is the total number of sentences.

In our keyword extraction task, the Dice coefficient measure is more appropriate than the mutual information measure. We consider the following extreme cases:

- When x and y are perfectly independent, i.e., $p(x, y) = p(x) \times p(y)$, the mutual information is equal to a constant while the Dice coefficient is determined by the frequencies of occurrences of x and y.
- When one word is fully determined by the other, i.e., $p(x, y) = p(x) = p(y)$, the Dice coefficient is equal to 1 whereas the mutual information is equal to $-logf(x)$ that grows with the inverse of the frequency of x.

Algorithm 4 KLC(d, n, m)

Input: d: Web news page; n: the number of candidate words;
m: the number of keywords extracted;
Output: the top m keywords.

1: Non-news content in the news Web page is filtered. Words are segmented and stemmed(for English words), and stop words are removed;
2: Compute the TFIDF of each word using Eq.(4);
3: Select the top n words by TFIDF as candidate words;
4: Build the disambiguation graph in which each node is a candidate word that is divided into several senses (concepts), and each weighted edge connects two word senses;
5: Perform the word sense disambiguation for each candidate word, and the one sense with the highest sum of similarities with other word senses is assigned to the word;
6: Build the actual lexical chains. An edge connects two words if the word similarity (using the assigned word sense) or the relatedness degree (the Dice coefficient value) exceeds the threshold t_3;
7: Compute the weight of each candidate word w_i as follows:
$Weight(w_i) = a \times TFIDF_i + b \times |chain_i| + c \times |related_i|$
where a, b, and c are parameters that can been adjusted. When a certain feature is used, the corresponding parameter is set to 1; otherwise, it is set to 0. $|chain_i|$ is the length of the chain in which w_i is, and $|related_i|$ is the number of related words linked with w_i;
8: Select the top m words as the keywords extracted from the candidate words by their weights.

It is obvious that the relatedness of two words is equal to 1 if one word occurs when and only when the other word occurs. The Dice coefficient measure satisfies this condition, while the mutual information measure does not. Our keyword extraction algorithm KLC (Keyword extraction based on Lexical Chains) is described as Algorithm 4, based on our KESR algorithm in the NFAS system.[10]

5.2. *Experimental Results on Web News Pages*

We select 120 news Web pages from the 163 website (http://news.163.com) as the experimental data to test performance of our method. We use ICTCLAS[31] to split Chinese documents into phrases. Keywords extracted are compared with the phrases in the news title and the phrases in the core hints provided by the editor. We use recall and precision as measures of extraction performance. The title recall R and the core hint precision P are defined as follows:

$$R = \frac{\#keywords\ matched\ with\ the\ title}{\#phrases\ in\ the\ title} \tag{14}$$

$$P = \frac{\#keywords\ matched\ with\ the\ core\ hint}{\#keywords\ extracted} \tag{15}$$

The parameter of n is 30 based on empirical studies. According to our experiments, n should be between 20 and 50; if it is smaller than 20, the advantages of semantic relations would not be evident, and if it is greater than 50, the importance of word frequency to the extracted keywords would be reduced.

The thresholds t_3 is set to 0.3 by some additional fine tuning in our experiments. The number of keywords extracted is selected as 3, 5, 7, and 10, respectively.

Experiment 1. In this experiment, we study the influence of selected features on the performance of keyword extraction. We first only use the TFIDF feature to score candidate words. Then, the $|chain|$ and $|related|$ features are respectively added to prove the improvement on the quality of keywords extracted.

Figures 3 and 4 show the precisions and recalls of KLC using three different feature sets to score candidate words when the number of keywords extracted is 3, 5, 7, and 10, respectively.

From Figures 3 and 4, we can see that both the $|chain|$ and $|related|$ features improve the quality of extracted keywords. The superiority increases with the number of keywords extracted decreased. The semantic relations of phrases are considered using the $|chain|$ and the $|related|$ features. The aim of additional semantic features is to extract the words with a low frequency but a great contribution to the text topic and to filter out the words with a high frequency but little contribution to the text topic, and the experiments have testified this design. It can also be seen that the $|related|$ feature outperforms the $|chain|$ feature. This is because that the $|related|$ feature reflects the direct related information of a candidate word, while the $|chain|$ feature reflects the total related information of the candidate words linked together.

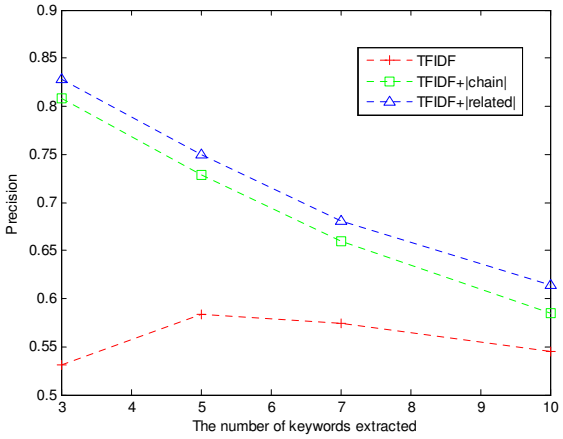


Fig. 3. The precisions of KLC with three different feature sets.

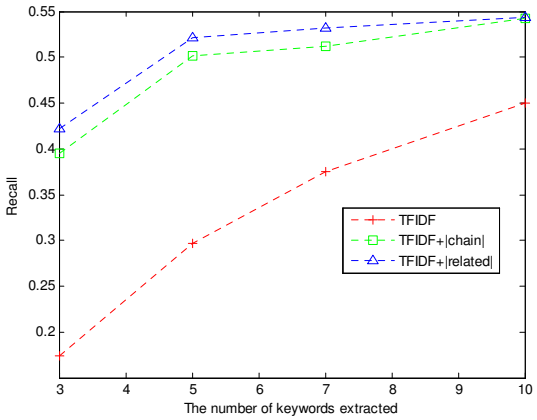


Fig. 4. The recalls of KLC with three different feature sets.

Experiment 2. Keywords are mainly the nouns in academic journals. However, verbs also play a key role in representing the news topics. In this experiment, we divide the candidate words into two sets. One consists of only nouns. The other contains both nouns and verbs.

Figures 5 and 6 show the precisions and recalls of KLC with different candidate word sets where both TFIDF and the $|related|$ features are used. The number of keywords extracted changes from 3 to 10.

It can be seen from Figures 5 and 6 that the quality of extracted keywords is improved after adding verbs into the candidate word set. The superiority increases with the number of extracted keywords. This demonstrates that when the number of keywords extracted is small, the most keywords are nouns. With the number of keywords increased, more verbs are extracted.

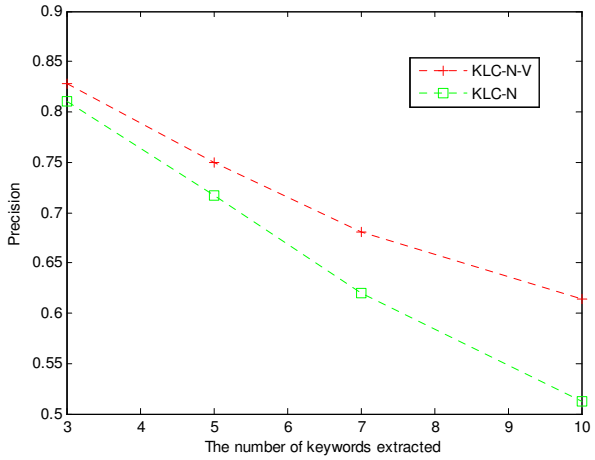


Fig. 5. The precisions of KLC with three different feature sets.

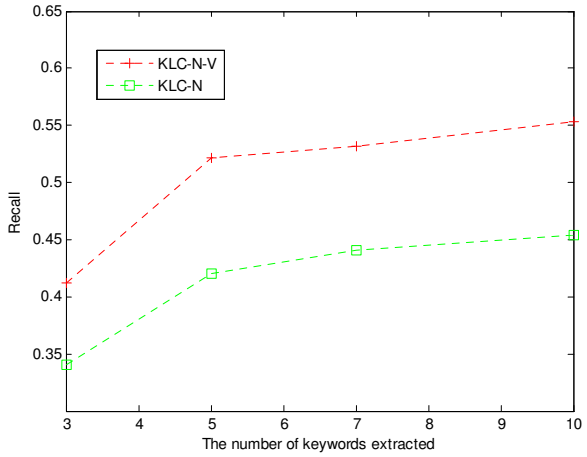


Fig. 6. The recalls of KLC with two different candidate word sets.

5.3. *Experimental Results on Journal Articles*

To study the performance of generalization, we also conduct experiments on journal articles. The corpus is collected by the Natural Language Processing group of the Fudan University International Database Center. We randomly select 200 Chinese journal articles with the keywords assigned by the authors. The precision P, recall R and F-measure are used as the evaluation metrics. They are defined as follows:

$$P = \frac{\#correct}{\#extracted} \tag{16}$$

$$R = \frac{\#correct}{\#labeled} \tag{17}$$

$$\text{F-measure} = \frac{2 \times P \times R}{P + R} \tag{18}$$

where *#correct* is the number of correctly extracted keyphrases, *#extracted* is the number of extracted keyphrases, and *#labeled* is the number of labeled keyphrases assigned by the authors.

Figures 7-9 show the comparative results of KLC and TFIDF on the journal articles corpus, where the numbers of changes on extracted keywords are 3, 5, 7, and 10. We can see that KLC always outperforms TFIDF.

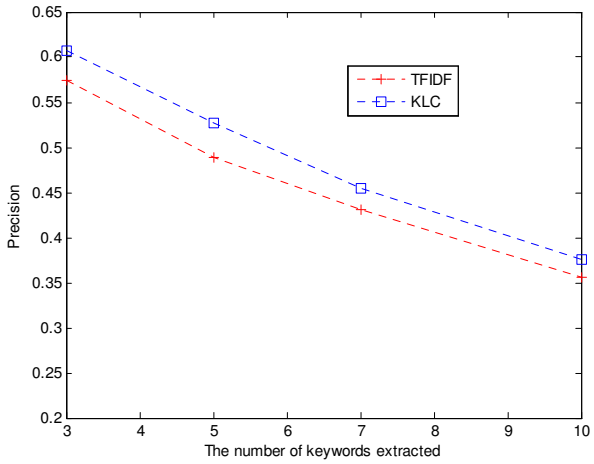


Fig. 7. The precisions of KLC and TFIDF on journal articles.

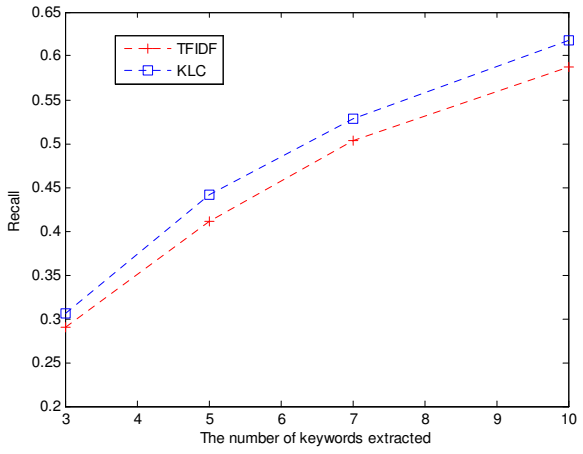


Fig. 8. The recalls of KLC and TFIDF on journal articles.

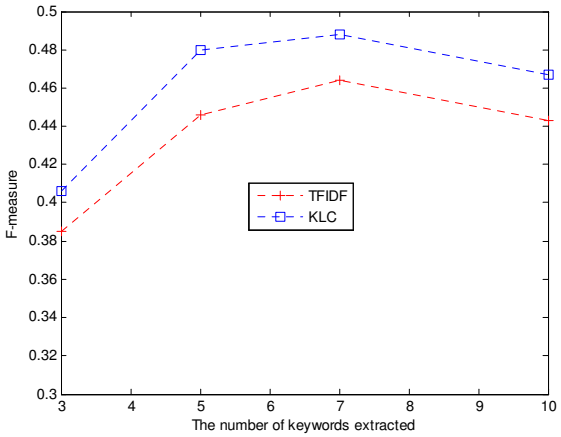


Fig. 9. The F-measure values of KLC and TFIDF on journal articles.

6. An Example Run

Figures 10-13 provide some screen shots of an example run. Figure 10 shows the interface of the personalized news filtering and summarization system (PNFS). There are three news navigators including recommended news, Google news, and browsed histories of the user. The recommendation list provides two categories of personalized news for users according to their registration information and the clicking histories. The news tracking reports the current popular news that is interesting to the user. For example, suppose the user has clicked the BBC news about hurricane Sandy hitting Cuba[a]. There are two (newly arrived) candidate news stories reporting Sandy. One is about Sandy moving toward central Bahamas[b]. The cosine similarity between the candidate news story and the clicked document is 0.52 which is lower than the redundance threshold 0.6 and higher than the interestingness threshold 0.3. Therefore, it is recommended to the user on the tracking news list. Another candidate news story from the Fox News website is about Sandy pounding Cuba[c]. The similarity is 0.67 which is higher than the redundance threshold. We think that the user has also known the news. Therefore, the second candidate news story is filtered out.

The topic interesting news reflects long-term preferences of the user. If the user has not signed in, he or she can only access the general Google News. The clicking history list records all the news stories that the user has browsed. The user's interesting topics are dynamically learned based on the clicking histories and the registration information. If the history list is empty that means the user has not clicked any news page, and the user's interest topics are collected from the regis-

[a]http://www.bbc.co.uk/news/world-latin-america-20078215.
[b]http://www.voanews.com/content/powerful-sandy-reaches-cuba/1532936.html.
[c]http://www.foxnews.com/weather/2012/10/25/sandy-makes-landfall-in-cuba/#ixzz2AK2mntG8.

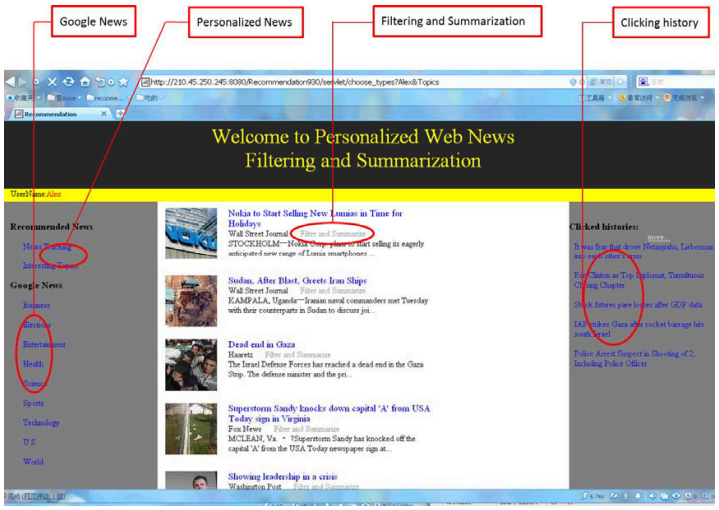


Fig. 10. The PNFS system interface.

tration information. This can avoid the cold start problem of the recommendation system.

Figure 11 shows the registration interface when the user firstly logs in the system. He or she needs to fill in some basic information, such as username, password, email address, and so on. There are two parts of registration information collected to reflect the user's interest preferences including interest topics and the interest keywords. There are totally nine topics in the topic list. The user can select one or more topics he or she is interested in. The user can also designate some keywords he or she focuses on. The input keywords are represented as a vector and then sent to the Naive Bayes classifier to obtain the topic category to which the keywords belong.

The users can either browse the original news Web page or read the filtered and summarized news content by clicking on the Filtering and Summarization link.

Figure 12 shows a partial original Web news page about Australia floods from CBC news (http://www.cbc.ca/). A lot of non-news content, such as advertisements and other non-relevant links exist on the original page.

A news filter is used to extract the news content and relevant pictures and filter out other parts that are not relevant to the news. Finally, the summarization component extracts keywords and their lexical chains from the news article. Figure 13 shows the filtered news page and the extracted keywords with lexical chains.

The extracted keywords are "flood", "home", "year", "record", "end", "Australia", "state", "overall", "Queensland", and "worst". There are three lexical chains that link the extracted keywords, which are:

1) flood, home, end, year, overall, worst;
2) Australia, Queensland, state;
3) record.

Fig. 11. The registration interface.

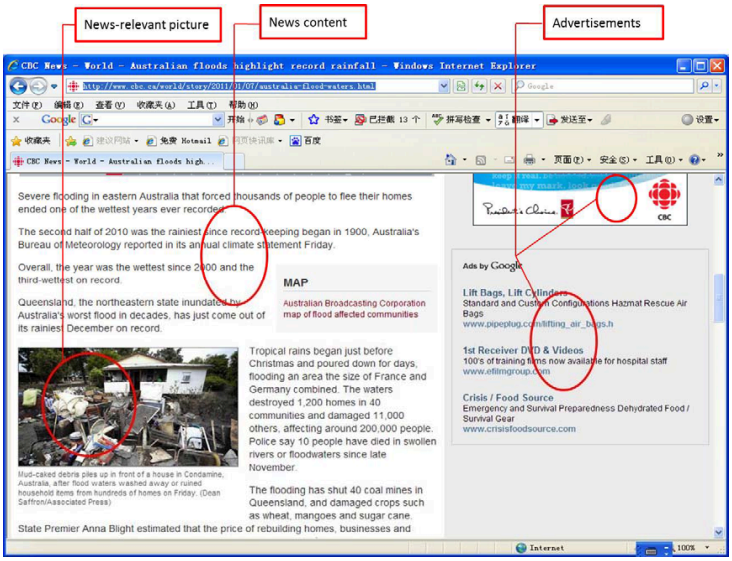
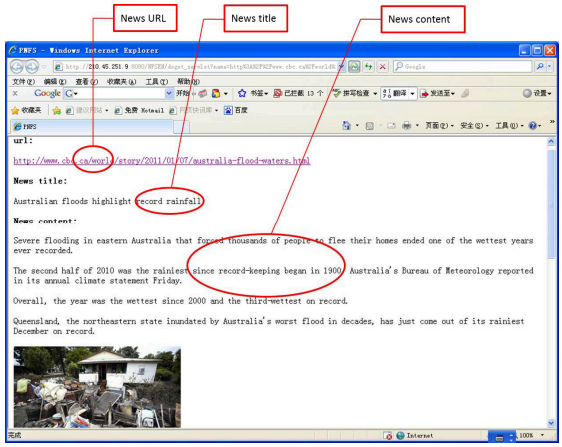


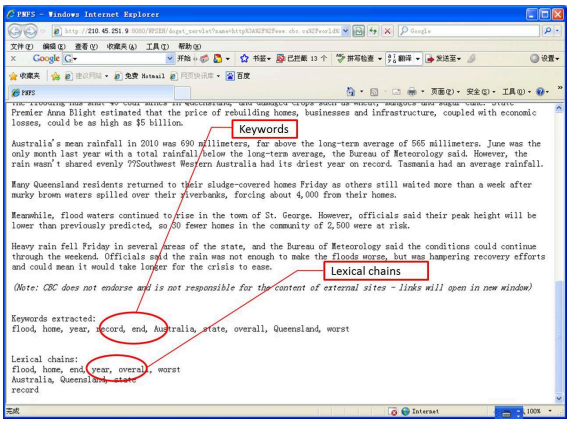
Fig. 12. The original news about Australia floods.

By a comparison between the news content and keywords extracted, we find that the summarization results are very close to the original news story.

Our PNFS system significantly differs from existing commercial news systems such as Google News. Google News is an automated news aggregator provided by Google Inc., and does not provide filtering and summarization functions. PNFS takes Google News as input, filters out non-news content, and summarizes the news



(a)



(b)

Fig. 13. The filtered and summarized news

in lexical chains.

7. Conclusions

In this paper, we have presented the recommendation and summarization components of our personalized news filtering and summarization (PNFS) system. For the recommendation component, we have designed a content-based news recommender that automatically obtains World Wide Web news from the Google news website and recommends personalized news to users according to their interest preference. Two learning strategies are used to model the user interest preference including the k-nearest neighbor and the Naive Bayes. The recommender not only keeps track

of the past news-reading events and finds novel candidate news stories, but also recommends the news that reflects the user's long-term topic interests. To better analyze the news content, a news filter is used to filter out the advertisements and other irrelevant parts on the news Web page.

For the summarization component, a new keyword extraction method based on semantic relations has been presented in this paper. Semantic relations between words based on lexical thesaurus and word co-occurrence are studied, and lexical chains are used to link the relations. Keywords of high quality are extracted based on the information in the lexical chains. There is rich information in lexical chains. In this paper, the lexical chains are built within a document. Future work can seek to construct lexical chains across documents and make a full use of the chains for recommendation. How to utilize the semantic information between the keywords extracted from the clicked news stories and find the most representative keywords to model the user's topic interests is another research issue to be explored.

Acknowledgments

This research has been supported by the National 863 Program of China under grant 2012AA011005, the National Natural Science Foundation of China (NSFC) under grants 61229301, 61273297 and 61273292, the Fundamental Research Funds for the Central Universities of China (2011HGZY0003), and the Jiangsu Provincial Key Laboratory of E-business, Nanjing University of Finance and Economics under grant JEB1103.

References

1. M. Pazzani and D. Billsus, Syskill & Webert: Identifying interesting web sites, in *Proceedings of the thirteenth national conference on Artificial intelligence* (1996), pp. 54–61.
2. A. Tan and C. Tee, Learning user profiles for personalized information dissemination, in *Proceedings of the IEEE International Joint conference on Neural Networks* (Anchorage, Alaska, USA, 1998), pp. 183–188.
3. J. Konstan, B. Miller, D. Maltz, J. Herlocker, L. Gordon and J. Riedl, GroupLens: Applying collaborative filtering to Usenet news, in *Communications of ACM* 40(3)(1997) 77–87.
4. A. S. Das, M. Datar, A. Garg and S. Rajaram, Google news personalization: scalable online collaborative filtering, in *Proceedings of the 16th International Conference on World Wide Web* (New York, USA, 2007), pp. 271–280.
5. I. Yacut and H. Polat, Privacy-preserving hybrid collaborative filtering on cross distributed data, in *Knowledge and Information Systems* 30 (2) (2012) 405–433.
6. P. Melville, R.J. Mooney and R. Nagarajan, Content-boosted collaborative filtering for improved recommendations, in *Proceedings of the 8th National Conference on Artificial Intelligence* (Edmonton, Canada, 2002), pp. 187–192.
7. X. Yu and W. Lam, Probabilistic joint models incorporating logic and learning via structured variational approximation for information extraction, in *Knowledge and Information Systems* 33 (2) (2012) 415–444.

8. A. Saiiuguet and F. Azavant, Building intelligent web applications using light weight wrappers, in *Data and Knowledge Engineering* 36 (3) (2001) 283–316.
9. D. Chakrabarti, R. Kumar and K. Punera, Generating succinct titles for web URLs, *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Las Vegas, Nevada, USA, 2008), pp. 79–87.
10. X. Wu, G. Wu, F. Xie, Z. Zhu, X. Hu, H. Lu and H. Li, News filtering and summarization on the web, *IEEE Intelligent Systems* 25 (5) (2010) 68–76.
11. P. D. Turney, Learning to extract keyphrases from text, *National Research Council, NRC Technical Report ERB-1057* (Canada, 1999).
12. H. Witten, G. W. Paynter, E. Frank, C. Gutwin and C. G. Nevill-Manning, KEA: Practical automatic keyphrase extraction, *Proceedings of the 4th ACM Conference on Digital Libraries* (Berkeley, California, US, 1999), pp. 254–256.
13. R. Mihalcea, Graph-based ranking algorithms for sentence extraction, applied to text summarization, *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics* (Madrid, Spain, 2004), pp. 58–65.
14. W. You, D. Fontaine, and J. Barths, An automatic keyphrase extraction system for scientific documents, *Knowledge and Information Systems* (available online, 2012), doi: 10.1007/s10115-012-0480-2.
15. S. Li, H. Wang, S. Yu and C. Xin, Research on maximum entropy model for keyword indexing, *Chinese Journal of Computers* 27(9) (2004) 1192–1197.
16. Y. Liu, X. Wang, Z. Xu and B. Liu, Ming constructing rules of Chinese keyphrase based on rough set theory, *Acta Electronica Sinica* 35(2) (2007) 371–374.
17. H. Suo, Y. Liu and S. Cao, A keyword selection method based on lexical chains, *Journal of Chinese Information Processing* 20(6) (2006) 25–30.
18. Q. Liu and S. Li, Word similarity computing based on How-net, in *Computational Linguistics and Chinese Language Processing* 7(2) (2002) 59–76.
19. M. Halliday and R. Hasan, *Cohesion in English* (Longman, London, 1976).
20. J. Morris and G. Hirst, Lexical cohesion computed by thesaural relations as an indicator of the structure of text, in *Computational Linguistics* 17(1) (1991) 21–48.
21. H. Peat and P. Willet, The limitations of term co-occurrence data for query expansion in document retrieval systems, in *Journal of American Society for Information Science* 42(5) (1991) 378–383.
22. D. Lin, An information-theoretic definition of similarity, in *Proceedings of the 15th International Conference on Machine Learning* (Madison, Wisconsin, August 1998), pp. 296–304.
23. Z. Dong and Q. Dong, *HowNet and the Computation of Meaning* (World Scientific Publishing Company, Singapore, 2006).
24. M. H. Alam, J. W. Ha, and S. K. Lee, Novel approaches to crawling important pages early, in *Knowledge and Information Systems* 33 (3) (2012) 707–734.
25. G. Salton, A. Wong and C. Yang, On the specification of term values in automatic indexing, *Journal of Documentation* 29(4) (1973) 351–372.
26. D. Billsus and M. Pazzani, Adaptive news access, in *The Adaptive Web: Methods and Strategies of Web Personalization* (Springer, 2007), eds. P. Brusilovsky, A. Kobsa and W. Nejdl.
27. D. J. Hand and K. Yu, Idiot's Bayes: not so stupid after all? *Internat. Statist. Rev.* 69 (2001) 385–398.
28. A. McCallum and K. Nigam, A comparison of event models for naive bayes text classification, in *AAAI/ICML Workshop on Learning for Text Categorization* (1998), pp. 41–48.
29. W. Frakes, *Information retrieval: data structures and algorithms* (Prentice Hall, 1992),

eds. W. Frakes and R. Yates.
30. K. Church, W. Gale, P. Hanks and D. Hindle, Using statistics in lexical analysis, in *Lexical Acquisition: Using On-line Resources to Build a Lexicon* (New Jersey, 1991), eds. U. Zernik and L. Hillsdale, pp. 115–165.
31. H. Zhang, Q. Liu, X. Cheng, H. Zhang and H. Yu, Chinese lexical analysis using hierarchical hidden markov model, in *Proceedings of the Second SigHan Workshop* (2003), pp. 63–70.