Personalized News Filtering and Summarization on the Web

Xindong Wu[1,2] Fei Xie[1,3] Gongqing Wu[1] Wei Ding[4]

[1] College of Computer Science and Information Engineering, Hefei University of Technology, Hefei, China
[2] Department of Computer Science, University of Vermont, Burlington, USA
[3] Department of Computer Science and Technology, Hefei Normal University, Hefei, China
[4]Department of Computer Science, University of Massachusetts Boston, Boston, USA

xwu@cs.uvm.edu, xiefei9815057@sina.com, wugg@hfut.edu.cn, ding@cs.umb.edu

***Abstract*—Information on the World Wide Web is congested with large amounts of news contents. Recommendation, filtering, and summarization of Web news have received much attention in Web intelligence, aiming to find interesting news and summarize concise content for users. In this paper, we present 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. A keyword knowledge base is maintained and provides a real-time update to reflect the general Web news topic information and the user's interest preferences. The non-news content irrelevant to the news Web page is filtered out. Keywords that capture the main topic of the news are extracted using lexical chains to represent semantic relations between words. An Example run of our PNFS system demonstrates the superiority of this Web intelligence system.**

***Keywords*—Component; Personalized News; Web News Filtering; Web News Summarization**

I. 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 Web pages. 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 interest 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 users' profile, and a keyword extraction algorithm is also provided to construct the lexical chains based on word sense disambiguation.

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.

II. RELATED WORK

A. Recommender Systems

There are mainly three different techniques commonly used in recommender 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 [14] aimed to rate pages on the Word Wide Web and recommend them to a user by analyzing the content on each page. Tan and Teo [19] 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.

The collaborative filtering approach uses the known preferences of a group of users to make recommendation for other users. Group-Lens [6] is a personalized news system using the collaborative filtering approach. Das *et al*. [3] applied collaborative filtering techniques to Google news.

Hybrid approaches combine content-based methods with collaborative filtering techniques, aiming to avoid the limitations of each approach and improve the recommendation performance [11].

B. 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.

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 [16] uses DOM tree paths to address a Web page. The data to be extracted are 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*. [2] took an extractive approach for title generation, which starts with URL tokens, HTML titles, keywords, and anchor text on incoming links etc. 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 news Web pages [22]. 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.

C. 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 [20] designed a keyword extraction system GenEX based on C4.5. Witten *et al*. [21] used Naive Bayes to extract keywords, and designed the Kea system. While supervised methods have some nice properties, like being able to produce interpretable rules to explain the associations between features and keywords, they require a large amount of training data. Many documents with known keywords are needed. Furthermore, supervised methods are not very flexible because training on a specific domain tends to customize the extraction process to that domain. Unsupervised keyword extraction removes the need for training dada. Instead of trying to learn explicit features that characterize keywords, the unsupervised approach exploits the structure of the text itself to determine keywords that appear "central" to the text. Mihalcea [12] presented a graph-based ranking method to keyword extraction.

The study of Chinese keyword extraction began in recent years. Li *et al*. [7] probed into keyword extraction using the Maximum Entropy model. Because the parameter estimation of feature selection is not always accurate, their results had much room for improvement. Liu *et al*. [10] mined a manually labeled keyword corpus which is from People's Daily newspaper and attain the constructed rules for Chinese keyword extraction. This approach needs a large number of labeled keywords. Suo *et al*. [18] presented a lexical-chain-based keyword extraction method for Chinese documents, and lexical chains were constructed based on the HowNet-based word semantic similarity proposed by Liu and Li [9]. 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.

D. Lexical Chains

The notion of cohesion is a device for "sticking together" different parts (i.e., words, sentences, and paragraphs) of the text to function as a whole. Lexical cohesion occurs not only between two terms, but also among sequences of related words, called lexical chains. Morris and Hirst [13] 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.

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 [15]. Two thesauruses, including WordNet and HowNet, are

respectively used to compute word similarity in English [8] and in Chinese [9]. The word co-occurrence model is adopted to solve the problem that it 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 [4]. 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 [9]:

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

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

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

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 [9].

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:

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

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 descriptional 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 β_1, β_2, β_3, and β_4 are usually set to 0.5, 0.2, 0.17 and 0.13 according to [9].

III. SYSTEM ARCHITECTURE

Figure 1 shows the PNFS system architecture. A new user is required to register with an initial interesting topic category 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.

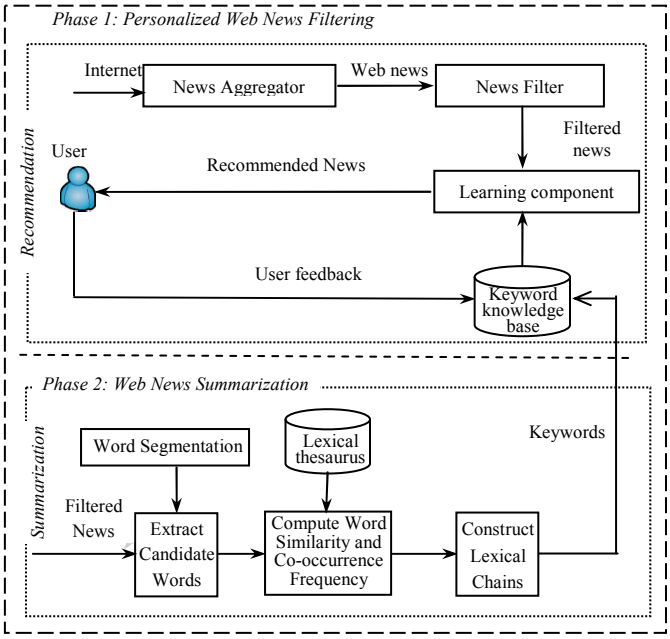


Figure 1. The PNFS system architecture

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. In this paper, we aggregate the world wide news from the Google News website. These news stories are automatically classified into different topic categories such as "world," "sports," "education," and so on. Two learning algorithms including the *k*-nearest neighbor and Naïve Bayes are used to model the user's 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 personalized keywords are selected from the browsed stories 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 [16], 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 values [17] 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 co-occurrence frequencies. Then keywords are extracted from the candidate words according to the TFIDF values and the semantic information in the lexical chains.

IV. Personalized Web News Recommendation

A. Recommendation Algorithms

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 [1]. 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 three groups: previous news tracking, interesting topics, and novelty news. The proportion of the recommended news for each group is defined by the user.

We use the *k*-nearest neighbor algorithm [1] to track previously read news and find novelty news. 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 it finds novel news.

After filtering out the non-news parts on the Web news page, each news article is converted to a TFIDF vector [17] as follows:

$$TFIDF_i = \frac{tf_i \times \log(N/n_i)}{\sqrt{\sum_j (tf_j \times \log(N/n_j))^2}} \qquad (4)$$

where tf_i is the frequency of word w_i in the given Web page, N is the number of documents in the corpus, and n_i is the number of documents in the corpus that contain word w_i.

The cosine measure is used to compute the similarity of two vectors. In this paper, we define three similarity thresholds: t_1, t_2, and t_3 ($0<t_1<t_2<t_3<1$) to decide whether a news story is interesting, not interesting, or novel. 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_3, the coming story is labeled as uninteresting (the user has known it). If the average of the k similarities is less than t_1, the story is labeled as novel; the smaller the average of the similarities is, the more novel the story is. If the average of the k similarities is larger than t_2, the story is labeled as interesting; the larger the average of the similarities is, the more the story is interesting. In our recommendation system, t_1, t_2, and t_3 are set to 0.05, 0.3, and 0.9, respectively, and k is 20 based on our empirical observations.

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 probability learning model, Naïve Bayes [5] to calculate the probability of news stories 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. For each news story (or the user preference), we can calculate the probability of the vector belonging to a given topic class according to the Naïve Bayes classifier.

Proposition 1 Assume that user u is independent to the news document d given the news topic classification model $C = \{c_1, c_2, ..., c_n\}$, where n is the number of news topic categories. The probability that document d is recommended to user u is computed as follows:

$$p(u \mid d) = p(u) \sum_{j=1}^{n} \frac{p(c_j \mid u) p(c_j \mid d)}{p(c_j)} . \qquad (5)$$

Proof: According to the conditional probability formula, $P(u \mid d) = p(u,d)/p(d)$, and by the total probability theorem, $p(u,d) = \sum_{j=1}^{n} p(u,d \mid c_j) p(c_j)$.

Then, $p(u \mid d) = \sum_{j=1}^{n} \frac{p(u \mid c_j) p(d \mid c_j) p(c_j)}{p(d)}$.

Since $p(u \mid c_j) p(c_j) = p(u) p(c_j \mid u)$, and

$p(d \mid c_j) / p(d) = p(c_j \mid d) / p(c_j)$,

$$p(u \mid d) = p(u) \sum_{j=1}^{n} \frac{p(c_j \mid u) p(c_j \mid d)}{p(c_j)}.$$

For a given user, $p(u)$ is a constant value, so we can recommend d to u using the formula:

$$p(u \mid d) \propto \sum_{j=1}^{n} \frac{p(c_j \mid u) p(c_j \mid d)}{p(c_j)}. \quad (6)$$

Given the similarity thresholds t_1, t_2, t_3, the number of most nearest neighbors k, where t_1, t_2, t_3, and k are decided by experiments, the algorithm outputs recommended stories according to the user's rated histories in the most recent past. We formalize the recommendation algorithm as follows.

1: **FOR** each upcoming news story **DO**
2: calculate the similarities of the news story with the user's recently rated stories and get k most nearest neighbors;
3: **IF** one of the k similarities is larger than t_3
4: label the upcoming story as uninteresting;
5: **CONTINUE**;
6: **IF** the average of the k similarities is larger than t_2
7: put the new story into the interesting queue;
8: **CONTINUE**;
9: **IF** the average of the k similarities is less than t_1
10: put the new story into the novelty queue;
11: recommend the stories in the interesting queue in the descending order of the average similarity;
12: recommend the stories in the novelty queue in the ascending order of the average similarity;
13: recommend the remaining stories according to the probability calculated by formula (6).

B. 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 category news 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 keywords automatically generated. 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.

V. Keyword Extraction Based on Semantic Relations

A. Keyword Extraction Algorithm

According to the word similarity and lexical chains reviewed in Section II, our keyword extraction algorithm KLC (Keyword extraction based on Lexical Chains) is designed as follows, based on our KESR algorithm in the NFAS system [22].

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 formula (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) exceeds the threshold t_4 or the word co-occurrence frequency exceeds the threshold t_5.
7: Compute the weight of each candidate word w_i as follows:

$$Weight(w_i) = a \times TFIDF_i + b \times |chain_i| + c \times |related_i| \quad (7)$$

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.

B. Experimental Results and Analysis

There are no standard news Web pages for keyword extraction. We select 120 news Web pages with core hints from the 163 website (http://news.163.com) as the experimental data to test performance of our method. We use ICTCLAS [23] 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:

$$R = \frac{\#\ keywords\ matched\ with\ the\ title}{\#\ phrases\ in\ the\ title} \quad (8)$$

$$P = \frac{\#keywords\ \ matched\ with\ \ the\ core\ hint}{\#keywords\ extracted} \quad (9)$$

The parameter of n is selected 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_4 and t_5 are selected 0.3 and 2 respectively by some additional fine tuning in our experiments. The number of keywords extracted is selected 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 2 and 3 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.

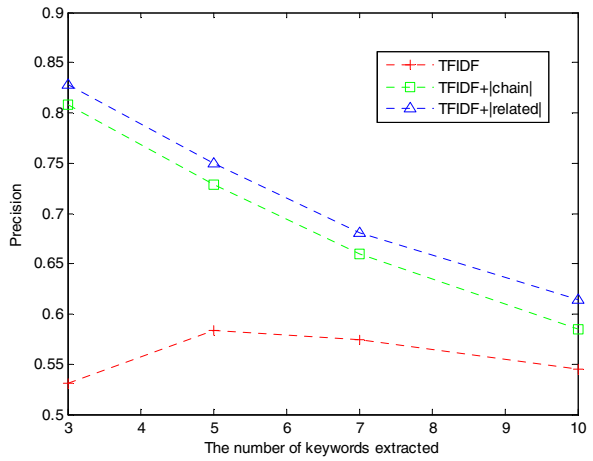


Figure 2. The precisions of KLC with three different feature sets

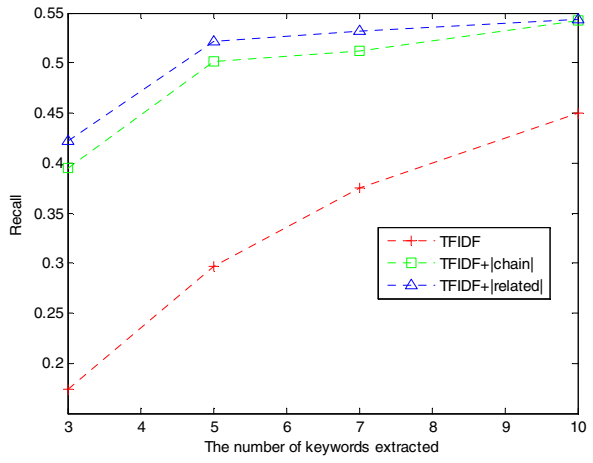


Figure 3. The recalls of KLC with three different feature sets

From Figures 2 and 3, we can see that both the |chain| and |related| features improve the quality of keywords extracted. 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.

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.

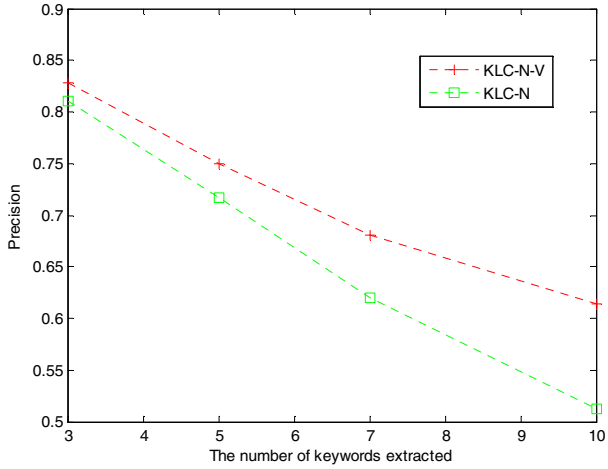


Figure 4. The precisions of KLC with two different candidate word sets

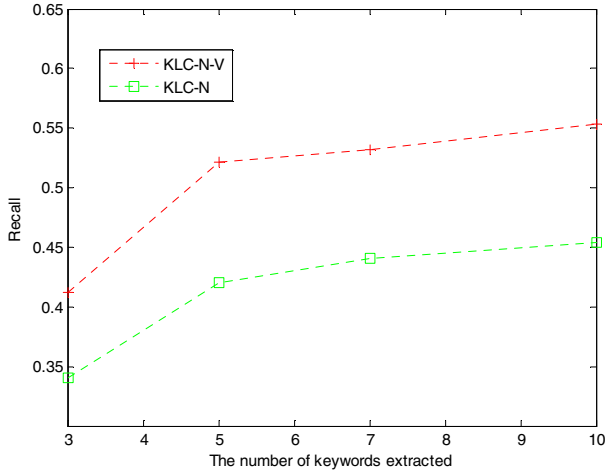


Figure 5. The recalls of KLC with two different candidate word sets

Figures 4 and 5 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 4 and 5 that the quality of keywords extracted is improved after adding verbs into the candidate word set. The superiority increases with the number of keywords extracted increased. 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.

VI. An Example Run

Figures 6-8 provide some screen shots of an example run. Figure 6 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 recommended news provides personalized news for users according to their preference. The Google news in PNFS provides general news avoiding missing important news. The browsing history is recorded so that the users know the recent progress of the event they pay attention to. The browsing history is also used to update the user preference profile. 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.

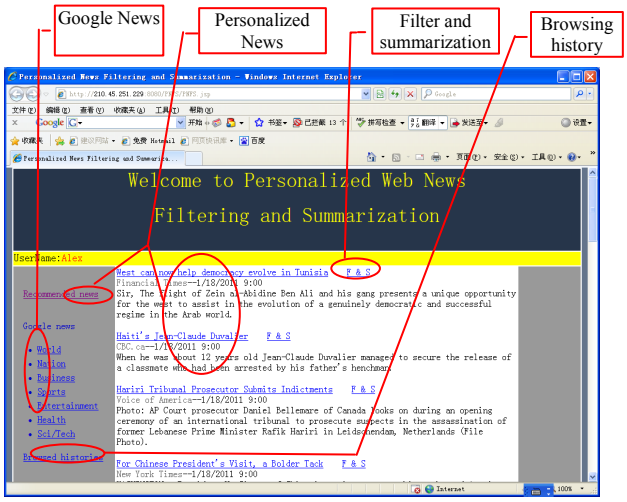


Figure 6. The PNFS system interface

Figure 7 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 8 gives the filtered news page and the extracted keywords with lexical chains.

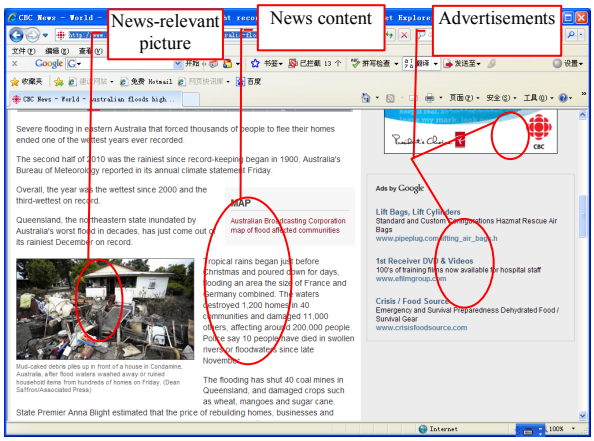
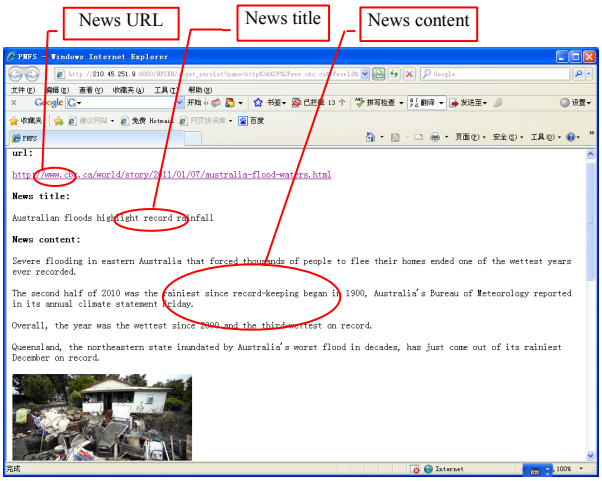
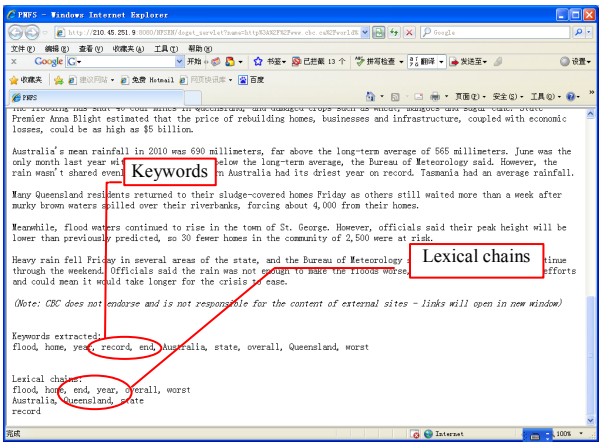


Figure 7. The original news about Australia floods



(a)



(b)

Figure 8. The filtered and summarized news

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

1) flood, home, end, year, overall, worst;

2) Australia, Queensland, state;

3) record.

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 in lexical chains.

VII. 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 Word Wide Web news from the Google news website and recommends personalized news to users according to their preference. Two learning strategies are used to model the user interest preference including the *k*-nearest neighbor and Naive Bayes. The recommender not only keeps track of the past news-reading events and finds novel news stories, but also recommends the news that reflects the user topic preference. 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. Future work can seek to construct better lexical chains and make a full use of the chains. How to utilize an emphasized format and the news related news for summarization is another research issue to be explored.

ACKNOWLEDGMENTS

This research has been supported by the National Natural Science Foundation of China (NSFC) under awards 60828005 and 60975034, the US National Science Foundation (NSF) under grant CCF-0905337, the National 973 Program of China under award 2009CB326203, 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 award JEB1103.

REFERENCES

[1] D. Billsus and M. Pazzani, "Adaptive news access," In: P. Brusilovsky, A. Kobsa, and W. Nejdl (eds.): *The Adaptive Web: Methods and Strategies of Web Personalization*, Springer, 2007.

[2] D. Chakrabarti, R. Kumar, and K. Punera, "Generating succinct titles for web URLs," *KDD-2008*, pages 79-87, Las Vegas, Nevada, USA, August 24-27, 2008.

[3] 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*, page 271-280. New York, USA, 2007.

[4] Z. Dong and Q. Dong, "HowNet and the computation of meaning," *Singapore: World Scientific Publishing Company*, 2006.

[5] D.J. Hand and K. Yu, "Idiot's Bayes: not so stupid after all?" *Internat. Statist. Rev.* 2001, 69, 385-398.

[6] J. Konstan, B. Miller, D. Maltz, J. Herlocker, L. Gordon, and J. Riedl, "GroupLens: Applying collaborative filtering to Usenet news," *Communications of the ACM 40*, 3: 77-87, 1997.

[7] S. Li, H. Wang, S. Yu, C. Xin, "Research on maximum entropy model for keyword indexing," *Chinese Journal of Computers*, 27(9): 1192-1197, 2004.

[8] D. Lin, "An information-theoretic definition of similarity," in *Proceedings of the 15th International Conference on Machine Learning*, pages 296-304, Madison, Wisconsin, August 1998.

[9] Q. Liu and S. Li, "Word similarity computing based on How-net," *Computational Linguistics and Chinese Language Processing*, 7(2): 59-76, 2002.

[10] Y. Liu, X. Wang, Z. Xu, B. Liu, "Ming constructing rules of Chinese keyphrase based on rough set theory," *Acta Electronica Sinica*, 35(2): 371-374, 2007.

[11] 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*, pages 187-192, Edmonton, Canada, 2002.

[12] R. Mihalcea, "Graph-based ranking algorithms for sentence extraction, applied to text summarization," in *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 58-65, Madrid, Spain, 2004.

[13] J. Morris and G. Hirst, "Lexical cohesion computed by thesaural relations as an indicator of the structure of text," *Computational Linguistics*, 17(1): 21-48, 1991.

[14] M. Pazzani, J. Muramatsu, and D. Billsus, "Syskill & Webert: Identifying interesting web sites," *AAAI/IAAI*, pages 54-61, 1996.

[15] H. Peat and P. Willet, "The limitations of term co-occurrence data for query expansion in document retrieval systems," *Journal of American Society for Information Science*, 42(5): 378–383, 1991.

[16] A. Saiiuguet and F. Azavant, "Building intelligent web applications using light weight wrappers," *Data and Knowledge Engineering*, 36(3): 283–316, 2001.

[17] G. Salton, A. Wong, and C. Yang, "On the specification of term values in automatic indexing," *Journal of Documentation*, 29(4): 351–372, 1973.

[18] H. Suo, Y. Liu, and S. Cao, "A keyword selection method based on lexical chains," *Journal of Chinese Information Processing*, 20(6): 25–30, 2006.

[19] A. Tan and C. Tee, "Learning user profiles for personalized information dissemination," in *Proceedings of the IEEE International Joint conference on Neural Networks*, pages 183-188, May 1998.

[20] P. D. Turney, "Learning to extract keyphrases from text," National Research Council, Canada, *NRC Technical Report ERB-1057*, 1999.

[21] I. H. Witten, G.W. Paynter, E. Frank, C. Gutwin, and C.G. Nevill-Manning, "KEA: Practical automatic keyphrase extraction," in *Proceedings of the 4th ACM Conference on Digital Libraries*, pages 254-256, Berkeley, California, US, 1999.

[22] 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): 68-76, 2010.

[23] 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*, pages 63-70, August 2003.