The Designing of a Web Page Recommendation System for ESL

Chen-Chung Chi, Chin-Hwa Kuo and Chia-Chun Peng
CAN Lab., Dept. of CS&IE, Tamkang University, Taiwan, R. O. C.
ryanjih@mail.tku.edu.tw, chkuo@mail.tku.edu.tw, 693191420@s93.tku.edu.tw

Abstract

In this paper, a webpage reading recommendation system is constructed through the concept of meta search and article summary technique. The desinged system recommends webpages that are related to the current webpage, to provide the user with further reading material. Using article-searching mechanism, the ESL student can avoid using keyword-based search method, thereby greatly decreasing the time spent to look for related articles. The system provides related articles as well as information such as the difficulty of the articles, which would assist English learning, and harbor a more user friendly English learning environment. This in turn increases learning efficiency. A designed toolbar serves as the main medium of communication with the user. All the user has to do is install the toolbar on the browser to gain the assistance from the system.

Keywords: Web Page Recommendation System, English as a Second Language, Meta Search API, Article Summary

1. Introduction

Throughout the process of English instruction and learning, listening, speaking, reading, and writing are four of the main topics. In particular, reading is one of the most important aspects. To improve one's reading ability, and to increase the enjoyment of learning English, outside reading is commonly used in the English classroom as part of the curriculum. Traditional forms of outside reading now include books and magazine articles. The Internet includes a variety of newspaper articles and discussions covering various topics. This specialty has also allowed the Internet to become a source for outside material. However, as the articles on the Internet are very diverse, it is hard to search for specific articles. One way to narrow down is to use Google[6] or Yahoo[11], but these search engines are limited to a keyword search method. Whether or not the article is related to the keyword does not matter. For a learner who would like to look for an article whose content is related to the keyword, searching by these search engines could prove to be quite time consuming.

A reading recommendation system is designed to overcome the above weakness. The system consists of two subsystems, namely, the *article collecting subsystem* and the *article rating/recommendation subsystem*. The system provides related articles as well as information such as the difficulty of the articles, which would assist English learning, and harbor a more user friendly English learning environment. This in turn increases learning efficiency. The present paper is organized as follows. Design of the reading recommendation system of ESL is presented in Section 2. Implementation and simulation results are shown in Section 3. Finally, conclusion and future works are discussed in Section 4.

2. Design of the Reading Recommendation System

After the English learner have read an article and would like to search for related further readings: (Fig.1)

(1) Figure out the topic with a list of keywords.
(2) Keyword list would be used in the search engine.
(3) Return a list of results, which the learner would assess the relevance to what he is looking for.
(4) Make some changes to the list of keywords, until a satisfactory related article is found.

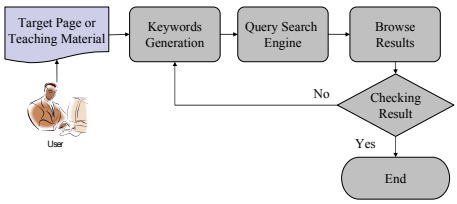

Fig. 1 Typical Situation of Finding Related Pages

In assessing the above situation, the first problem is how to figure out good keywords for an

ESL student, due to his unfamiliarity with English. Second, he would need to spend a lot of time of browsing through the huge amount of returned results. Therefore, the main purpose of our system is to filter out unrelated webpages by using the information provided by the search engine, thereby providing the learner with webpages with quality content.

The two main sections of the designed system are:

(1) The *article collecting subsystem*, which performs

- Text summarization of the webpages,
- Keywords extraction, and
- Meta search API.

(2) The *article rating/recommendation subsystem*, which conducts to

- Rate the comprehensibility and difficulty,
- Filter out unrelated articles, and
- Make some recommendations to the learner.

When a learner is interested in the article he is reading (named "target webpage"), and would like to read more related articles, the system's toolbar will search through its related webpage database to check for related information. If any result returns, the system will provide the learner with the related information directly. If there is no result returns, the article collecting subsystem will perform meta search for possible related webpage, and then saved in the database after analysis.

2.1. Article Collecting Subsystem

For a search engines, first task is to use a program such as "crawler" to collect a large number of websites before analyzing or recommending webpages. However, this task consumes process time and data storage space. Therefore a meta search is used to lower the number of webpages that need to be collected, and then saving process time and data storage space. The system flowchart is depicted in Fig. 2:

When a learner wants to find some webpages related with target webpage:

(1) Click on the "Search for Related Webpages" button on the toolbar.
(2) Webpage will be referenced as a "Target page" to search for related information in the database.
(3) If there is no related webpages in the database, Target webpage will be analyzed and then some keywords are produced.
(4) Finally, meta search API searches for related web pages, and saved into the related webpage database.

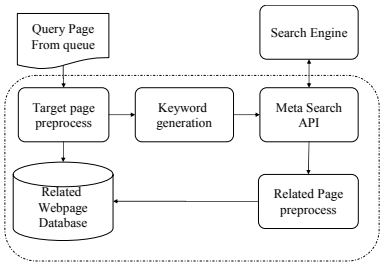

Fig. 2 Flowchart of the Related Page Subsystem

2.1.1 Target Page Preprocess: Fig.3. depicted description of each step of the process:

- Outlink Extraction:
  We uses a RE (Regular Expression) method to quicken the sentence comparison process.
- Page content extraction:
  To filter out the target webpage's advertisements and other unrelated data [9][10].
- Sentence Extraction:
  Sentences are extracted through the RE method.
- Keyword Extraction:
  Extracted keywords and stopwords list are compared. Keywords that appear in the stopwords list are eliminated. The times a keyword appears is counted.
- Vocabulary Difficulty score:
  A Bayes classifier [7] is used to test the lexical difficulty and found a correlation between the difficulty of the vocabulary and the lexical difficulty of the webpage content. We calculated the number of times a keyword appears in the database BNC[2]. The more the keyword appears, the less difficult it is.
- Page characteristics:
  Finally, some characteristics are obtained by the target webpage are the following: HTML tags, page content, outlink, sentences, difficulty score, and keyword vector.

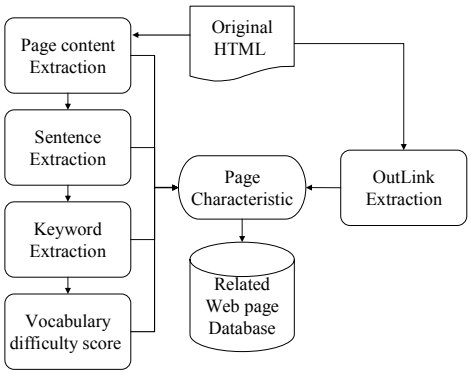

Fig.3 Target Page Preprocess

2.1.2 Keyword Generation: Text Summarization (TS) method is a way to translate the original text to a more simplified version. It still keeps information from the original text. relevance measure [12] takes the original article and each sentence in the article as term-frequency vectors, and compares the distance between the vector of the sentence and the vector of the article. Important sentences can be found in this way. On the other hand, a Singular Value Decomposition (SVD) method was proposed [12] for latent semantic analysis.

LexRank[5] is another method which uses graphics. It's originated from the "prestige" value in calculating social networks. It uses a node to represent each body, and a link to represent each relationship. LexRank correlates each sentence to a node, and the distance as the relationship, to come up with what is likely to be the central sentence. This central sentence will serve as the representation of the article. PageRank(Brin, Pag[8]) is proposed to calculate webpage prestige.

Keywords are extracted after preprocessing, every sen-tence is represented by a term-frequency vector. Every sentence set as a node, and each distance represented with a link, a relation network can be drawn up. The prestige value of each sentence can be calculated by LexRank[5].

If the word count for the summary is too large, this may take up much memory space and processing time. If it's too small, the quality of the list of recommendations may be affected. Therefore, following equations are considered to decide how many sentences are needed after summarization.

$$N = |S|(\alpha + (1-\alpha)\beta D) \quad (1)$$

$$D = 1 - \overline{C} \quad (2)$$

Where N is the number of sentences extracted. S is the sentence set from the articles. α, β are adjustable variables between 0~1. $\overline{C}$ is a clustering coefficient [3], originally used to determine the closeness of the nodes in a network. The entire system's clustering coefficient is the average of all points defined by [3].

The functions used in our system are based on two assumptions. (1)The longer the original article, the more content it contains. Therefore, that means more content will be in the summary. (2)If the clustering coefficient is low, then more sentences need to be picked to represent a target webpage.

This paper uses two factors α、β as the determinants of the number of sentences in the summary. α decides the effect of the original article's length on the length of the summary. β decides the effect of the clustering coefficient on the length of the summary. Finally the sentences extracted undergo TFIDF.

2.1.3 Meta Search API: It's commonly used in a meta search engine, combines the search results of many search engines. For instance, how to combine many searches as one meta search is shown in [1]. [13] innovates an agent platform with search engine as its base, also using the concept of meta search. Google Search API [14] uses SOAP and WSDL to collect search results.

2.1.4 Meta Score: Our system prioritizes articles in terms of the degree of similarity between the searched articles and target webpage. However, Meta search provides some useful information such as the prestige value. Therefore, this system calculates and defines meta score returned from a Meta search engine.

At First, the system extracts vocabulary from sentence, and calls Meta Search API for each keyword. Every sentence in the summary will be matched up with a post-search website.The system will then record order by which google returned this website and calculate its meta score.

2.2. Webpage Recommendation System

The main purpose of the system is to provide appropri-ate related article,to avoid any unrelated or difficult article is provided to the learner. In determining how appropriate the article is,this system takes into consideration the target webpage's similarity, comprehensibility, and meta score.

2.2.1 Reading ability checking: Webpages can be separated into different types. An ESL student would expect this system to provide articles and webpages with quality content. At this point of view, this system categorizes webpages into two types: hub (website) and authority (articles). Hub type contains links to many webpages (like a homepage). Authority type contains fewer links than a Hub, but more content.

If the number of links is less than 1.2 times the number of sentences extracted, this webpage is of the authority type. When the system calculates the similarity value, it will only consider authority type webpages.

2.2.2 Order of Recommendation and Similarity Value Mechanism: We will calculate the similarity values between all the collected related webpages and target webpages. When the learner decides that he is more interested in a subtopic after reviewing the list of

collected webpages, he can mark and adjust the weight value for a certain section or sentence in the original article or webpage. The system will check if there is any keyword in the marked section, and modify the calculations for the similarity value.

Finally, using the adjusted target webpage's TF value in relation to the original value, the order will be adjusted.

3. System Implementation and Experiments

This system has provided a user interface and toolbar (Fig. 3) to more readily assist the learner throughout the search. The learner can click left "Search for related webpages" button to search for a webpage. He/She can also click on "Search List" on the right side of the toolbar menu to call out webpages that have been previously searched. After the user logs in, there is a personalized search history on the left frame. By clicking a website on this list, the website will appear on the right frame of the browser. The status bar shows the status of webpage collection. If a link for "search results" appears, that means the webpage collection has been completed. Then a learner can directly click on the link and look at the results. If the link for "search results" does not appear, it means that the search has not been completed.

Fig. 3 also shows the list of recommendations. The first row is the target webpage, the second row is the recommended webpage, difficulty score and its summary.

We use the clustering coefficient method to determine the length of the summary. To support this point, we collected five articles of similar length, and selected sentences by five trials, then calculated the mean number of sentences selected. The results are shown Table 1.

Table 1. Experimental Results

Article No.	1	2	3	4	5
Clustering Coefficient	0.29	0.38	0.45	0.54	0.68
Reverse Clustering Coefficient	0.71	0.62	0.55	0.46	0.32
Mean number of sentences selected	4.2	3.8	3	3.2	2.2
Summary Similarity Value	54%	48%	18%	42%	51%
Recommendation Precision Value	78%	82%	16%	76%	80%

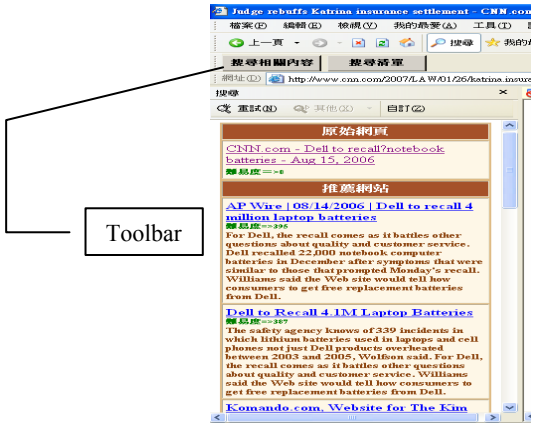

Fig. 3 Toolbar and UI of the search results of the reading recommendation system

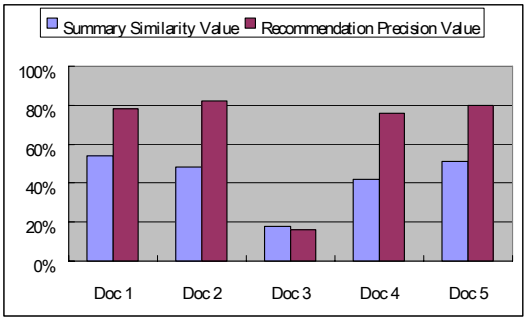

Fig. 4 Reverse Clustering Coefficient and Summary

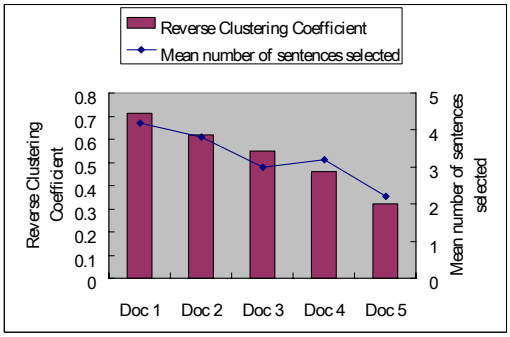

Fig 5 Comparison of Precision Value

As shown in Fig.4, there is a relationship between the number of sentences selected by learner from the summary and the reverse clustering coefficient of the article with the exception (article 3). The higher the reverse clustering coefficient, the more variety of topics the article has, causing the system need to select more representative sentences to represent it. However,

article 3 is a fable, so although it have a higher reverse clustering coefficient, but the system only selected a few sentences from the story's ending. This resulted in a clustering coefficient that is not low, but the summary's mean number of sentences does not correlate to the reverse clustering coefficient directly.

In the second part of the experiment (Fig. 5), the article summary and the selected part of the summary are compared. The sentence collection selected by the learner in Part 1 of the experiment is formed into a summary. Next we investigate the effectiveness of the system. Ten recommended webpages are shown to each of the five users that were in Experiment 1. These users determine if the webpage and original content are "related" or "not related". The results of this experiment are shown in Table 1. The analysis graph is shown in Fig. 4.

The precision value of first ten recommendations falls between 70% ~ 80%. Even if the summary and selected part of the summary have a low similarity value, the recommendations still have 70% precision value.

4. Conclusion

We have proposed a method based on meta search, combined with article summarization technique, to assist an ESL learner in deciding keywords, and to decrease the number of repeated search in the process, thereby increase the learning and reading efficiency of the learner. The special features of our system are (1) Use meta search and article summarization technology to construct an article search mechanism. (2) Apply a technique originally used for evaluating social networks, to article summarization.

This system regards keywords obtained from TFIDF as sentences, which are then sent to the meta search engine. Then, these keywords should contain information that would assist in the search. For instance: inputting the same set of keywords in different sequential orders for different searches would have different results. Therefore, how to order these keywords more effective after the summary would be a future area for research.

5. Reference

[1] A. Gulli & A. Signorini, "Building an Open Source Meta-Search Engine", *Special interest tracks and posters of the 14th international conference on World Wide Web WWW*, Chiba, Japan, May 2005, pages 1004 – 1005.

[2] BNC - British National Corpus, October 2006.

[3] D. J. Watts & S. H. Strogatz, "Collective dynamics of 'small-world' networks", NATURE, JUNE 1998, VOL 393, pages 440 – 442.

[4] DOM, http://www.w3.org/DOM/, 2006.

[5] G. Erkan & D. R. Radev, "LexRank: Graph-based Lexical Centrality as Salience in Text Summarization", *Journal of Artificial Intelligence Research*, 2004, 22:457–479.

[6] Google. http://www.google.com, 2006/11.

[7] K. Collins-Thompson & J. Callan, "A Language Modeling Approach to Predicting Reading Difficulty", *Proceedings of HLT/NAACL 2004*, Boston, USA, 2004.

[8] L. Pages, S. Brin, R. Motwani, & T. Winograd, "The pagerank citation ranking: Bringing order to the web". *Technical report*, Stanford University, CA, 1998.

[9] L. Yi, B. Liu & X. Li, "Eliminating Noisy Information in Web Pages for Data Ming", *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, Washington D. C., 2004, pages 296 – 305.

[10] S. Gupta, G. Kaiser, D. Neistadt & P. Grimm, "DOM-based Content Extraction of HTML Documents", *In WWW2003 proceedings of the 12 Web Conference*, Budapest, Hungary, May 2003, pages 207 - 214.

[11] Yahoo. http://www.yahoo.com, 2006/11.

[12] Y. Gong & X. Liu, "Generic Text Summarization Using Relevance Measure and Latent Semantic Analysis", *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval SIGIR '01*, New Orleans, Louisiana, United States, September 2001, pages 19 – 25.

[13] J. Chen & W. Liu "A Framework for Intelligent Meta-search Engine Based on Agent", *Proceedings of the Third International Conference on Information Technology and Applications (ICITA'05) Volume 2 - Volume 02 ICITA*, Washington D. C., July 2005, pages 276 – 279.

[14] Google SOAP Search API. http://code.google.com/apis/soapsear