Can Automatic Abstracting Improve on Current Extracting Techniques in Aiding Users to Judge the Relevance of Pages in Search Engine Results?

Shao-Fen Liang Siobhan Devlin John Tait
University of Sunderland School of Computing & Technology
St. Peters Campus Sunderland, Tyne & Wear SR6 0DD
+44(0)191 5153410
{shaofen.liang, siobhan.devlin, john.tait}@sunderland.ac.uk

ABSTRACT

Current search engines use sentence extraction techniques to produce snippet result summaries, which users may find less than ideal for determining the relevance of pages. Unlike extracting, abstracting programs analyse the context of documents and rewrite them into informative summaries. Our project aims to produce abstracting summaries which are coherent and easy to read thereby lessening users' time in judging the relevance of pages. However, automatic abstracting technique has its domain restriction. For solving this problem we propose to employ text classification techniques. We propose a new approach to initially classify whole web documents into sixteen top level ODP categories by using machine learning and a Bayesian classifier. We then manually create sixteen templates for each category. The summarisation techniques we use include a natural language processing techniques to weight words and analyse lexical chains to identify salient phrases and place them into relevant template slots to produce summaries.

1. INTRODUCTION

Nowadays with massive amounts of information provided on line, conventional search engines are utilised to find web pages. These use sentence extraction techniques to produce snippet result summaries, which are however less coherent and readable than the original documents. Users have to spend more time thinking about each summary and finding desired pages because the summary may not express the contents of the page well. Unlike abstracting, extracting programs do not create new text. Therefore they are popular as they offer a relatively low cost and fast solution [6]. On the other hand, abstracting techniques first analyse the context of the document then rewrite it into an informative summary. We are arguing, however, that abstracting a summary will present document content better than snippet sentence extraction but the automatic abstracting technique has the problem of domain restriction. For solving this problem we need to classify web documents into several domains. Therefore the research aims to produce informative summaries to reduce web users' time on thinking about relevant pages from search results by constructing an automatic abstracting system to present search engine result summaries.

2. BACKGROUND

Currently researchers have investigated different methods for addressing the text categorisation problem: many of them have employed Machine Learning approaches such as Decision Trees [4], Bayesian classifiers [9], K-nearest neighbour [1] and Support Vector Machines [11] to induce the category for a document based on a set of training examples. Part of the research includes a text categorisation problem, which is to assign web documents into several categories, for summarisation use. Text classification is used commonly to help information retrieval with the indexing process [10], thus the documents are often classified prior to retrieval. The process in our project has different purpose, which is to help on our querying process. We retrieve documents first then classify them into different categories for summarising to help our summaries present specific characteristics of each category. Our training examples are retrieved from existing ODP categories to ensure the training set is of high quality. Moreover, we intend to cover the whole English web, thus our data will be very diverse. The above reasons and our use of Perl for implementation led us to decide to use a Bayesian classification method for text categorisation because it is fastest in these circumstances. The text categorisation problem is an important but minor element of this research, which is necessary to overcome before we can move onto the summarisation stage. We have used existing tools to achieve this objective and try to gain better performance for automatic abstracting summarisation.

Since the 1950s researchers [5] have paid great attention to helping readers extract meaningful content from an information source in a condensed form. Many groups [8] [2] [12] have produced different summarisers whose goals are to produce a condensed representation of the content of its input for human consumption. There are two approaches for producing such summarisers: shallow approaches typically produce extracts, usually by extracting sentences from source documents [7]. Deeper approaches usually involve Natural Language Generation from a semantic or discourse level representation. Although currently there is no commercial search engine using deeper approaches, we are assuming that applying a Natural Language Processing approach would achieve coherent textual summaries. Moreover, abstraction methods first build a semantic representation for sentences. Then new semantic representations are created by selection, aggregation and generalisation operations [6]. These steps are typically quite knowledge-intensive and domain independent. Although our aim is to summarise web documents, the nature of the web is that it covers many domains. Therefore it would be impossible to summarise using only one abstracting template because one template can only present one domain's documents. Thus first of all we need to construct sixteen templates manually, one for each category to express clear domain knowledge.

3. CLASSIFICATION FOR SUMMARISING

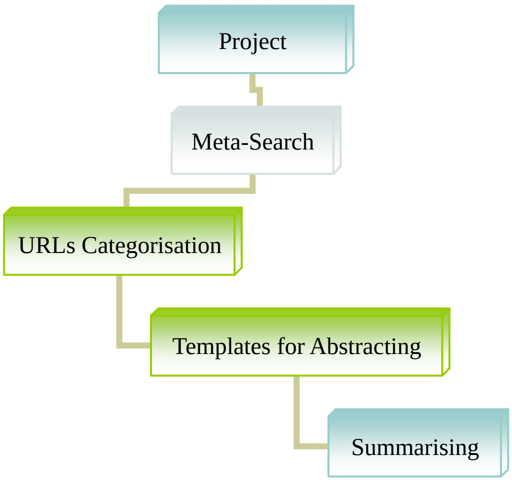

Figure1: Project Algorithm

The algorithm of our project has four steps: 1. Constructing a meta-search engine to retrieve the URL. 2. Categorising each URL into one of sixteen categories. 3. Producing templates for abstracting. 4. Summarising the web pages and returning the abstracted summary. (Figure 1) In the first step we construct a meta-search engine and employ 100 queries from TREC to retrieve the URLs. The reason for using these 100 queries is that TREC's web track has used a common collection and set of user queries with real users, which thus avoids personal bias in what is input which might otherwise confound the research result. We then check if the URL has been categorised by the Open Directory Project (ODP)[1]. The ODP is the largest, most comprehensive human-edited directory of the Web. It is constructed and maintained by a vast, global community of volunteer editors. We have chosen appropriate examples from the top level of ODP's categories, which has sixteen categories for further supervised learning use.

4. SUMMARISATION TECHNIQUES

Our summarisation approach is to first of all manually create templates for each of the sixteen categories then choose appropriate phrases to expand these templates into summaries. Summarising web pages poses many challenges, which are different from summarising plain text articles. Particularly our summaries need to be short to be displayed at a glance in a browser. Documents that contain too many words are problematic as they increase the size of the vector. Those containing very few words also present a difficult task. Some pages have little text but in addition include various elements such as tables, images, links and flashes. These elements are difficult to be used as summary material but they still present web pages well. In addition, for browsing and ease of navigation reasons, script language and HTML tags will also appear on the web pages. These factors become barriers for web summarising. To conquer these barriers, the process starts from retrieving URL then removing noise from the

[1] http://www.dmoz.org

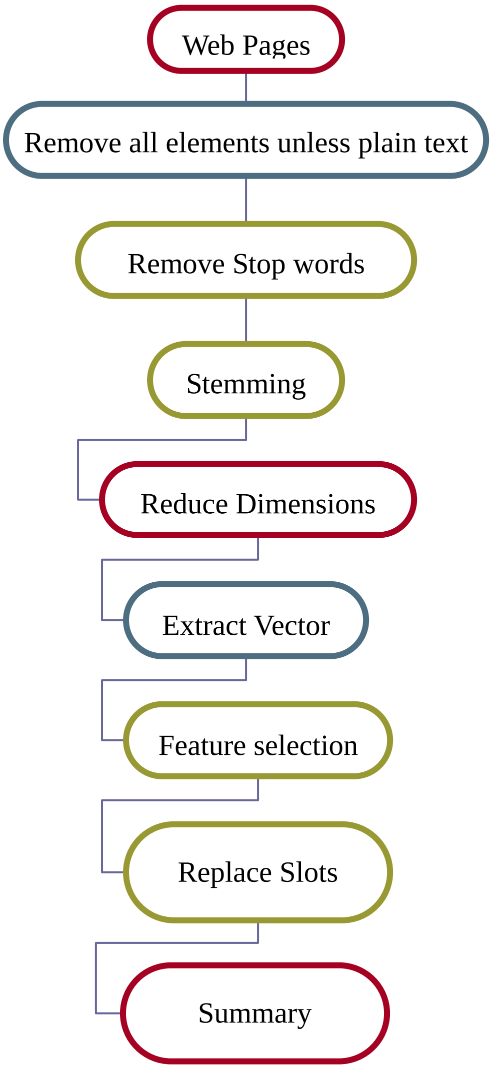

Figure 2: Process for summarisation

page. This can be done easily but transferring the remaining text into a bag of words vector representation presents many difficulties because the huge vector dimensions will take too long to execute. Therefore, first we have to reduce dimensions by using Term Frequency and Inverse Document Frequency weighting, and then normalise the vector. Finally, we analyse lexical chains, as used by Alfonseca [3], to extract important information from the source pages. Salient phrases will be identified and placed in the relative template slots to provide abstracted summaries. (Figure 2)

5. EVALUATION

In evaluating our system we will employ two methods: one is human judgement and the other is baseline comparison. For the human judgement, we will choose five testers. Each will be given a test sheet, which prints an input query and ten output summaries in two styles. One style is from our abstraction system and the other is an extraction style from the Google search engine. We will then ask the testers to read each summary and assign them a score for comprehensibility. They will also be asked to tick a comparison box stating which of the two styles of summary is easier to understand. The test sheets are produced before our testing process starts and the testing process is conducted off-line because we want to avoid human computer interaction becoming one of the variables that might affect the test result. The other way to evaluate our system is to compare our summaries with automatically created baseline summaries. The National Institute of Standards and Technology (NIST) has founded the Document Understanding Conference (DUC)[2] to promote advances in summarization techniques and enable

[2] http://www-nlpir.nist.gov/projects/duc/

researchers to participate in large-scale experiments. They have established an evaluation road map for summarization research. We will use the data from DUC to test the proposed approach and send the result to DUC for evaluating.

6. CONCLUSION

In this paper, our research project has been described. We hope to present more readable and easy to understand summaries to help users judge relevance. Our approach contains two major parts: text classification and automatic summarisation, where classification is used to overcome the domain restriction on automatic summarisation.

The initial idea of just sixteen summary templates may be too coarse to cover all domains of web documents. We hope that during the project development period, the sixteen templates can be produced automatically, thereby enabling the quantity of templates to be extended dramatically.

References

[1] B.B. Wang; R.I. McKay; H.A. Abbass; M. Barlow. Learning text classifier using the domain concept hierarchy. Communications, Circuits and Systems and West Sino Expositions, IEEE 2002 International Conference on, Volume: 2, 29 June-1 July 2002 pp: 1230 -1234.

[2] D. McDonald; H. Chen, Summarization and question answering: Using sentence-selection heuristics to rank text segments in TXTRACTORC. Proceedings of the second ACM/IEEE-CS joint conference on digital libraries.

[3] E. Alfonseca. A WordNet interface to APL2. ACM Series-Proceeding-Article, pp7-16, 2002.

[4]H. Liu; J.X. Yu; H. Lu; J. Chen. Unifying decision tree induction and association based classification. Systems, Man and Cybernetics, 2002 IEEE International Conference on, Volume: 7, 6-9 Oct. 2002.

[5]H.P. Luhn. The automatic creation of literature abstracts. IBM journal of Research and Development, Vol. 2, No, 2, pp:159-165, 1958.

[6] I. Mani. Automatic summarization. John Benjamins Publishing Company, 2001.

[7] J. Goldstein; V. Mittal; J. Carbonell; J. Callan. Creating and evaluating multi-document sentence extract summaries. Proceeding of the ninth international conference on information and knowledge management, November 2000.

[8]J. Kupiec; J. Pedersen; F. Chen. A trainable document summarizer. In Annual ACM Conference on Research and Development in Information Retrieval, pp: 68-73, 1995.

[9]K.M.A. Chai; H.T. Ng; H.L. Chieu. Bayesian online classifiers for text classification and filtering.

[10] R. Hoch. Using IR techniques for text classification in document analysis. Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval, pp 31-40, 1994.

[11] T. Joachims. Text categorization with

support vector machines: learning with many relevant features. In European Conference on Machine Learning, pp 137-142, 1998.

[12]Y. Gony; X. Liu. Generic text summarization using relevance measure and latent semantic analysis. In Annual ACM Conference on Research and Development in Information Retrieval, pp: 19-25, 2001.