KPNM at the NTCIR-11 MobileClick Task

Dong Zhou, Zhao Wang, Ziyi Zeng, Tuo Peng
Key Laboratory of Knowledge Processing and Networked Manufacturing &
School of Computer Science and Engineering,
Hunan University of Science and Technology, Xiangtan, Hunan, China
+86 187 1133 1970
dongzhou1979@hotmail.com, 308384127@qq.com, 1193746367@qq.com,
543183993@qq.com

ABSTRACT
This paper describes KPNM's (Knowledge Processing and Networked Manufacturing lab at Hunan University of Science and Technology) participation in the Mobile Information Access ("MobileClick") task at the NTCIR-11. We chained simple techniques based on statistical models and heuristic rules to extract significant text units from the web pages retrieved by the given query. Then we ranked the text units by using the vector space model. The text units are regarded as iunits as the final results. This is our first attempt in this task. Due to the limited human resources and short of time, more measures should be considered in the future for generating the iunits particularly for use in the mobile devices.

Categories and Subject Descriptors
H.3.3 [**Information Systems**]: Information Storage and Retrieval - *Information Search and Retrieval;* I.2.7 [**Computing Methodologies**]: Artificial Intelligence - *Natural Language Processing.*

General Terms
Algorithms, Experimentation, Languages.

Keywords
Information Extraction, statistical models, segmentation, Vector Space Model.

1. INTRODUCTION
This year's participation in the NTCIR-11 MobileClick task [1] was motivated by two factors. One is we want to show how quickly we can chain simple techniques from the information extraction and the information retrieval fields to complete this task. Another factor is to train the undergraduate students working in our lab. This task is fully completed by the third-year undergraduate students, in a very short time.

Our participation is limited to the iUnit Retrieval Subtask. In summary we submitted three MANDATORY runs, all in English. Basically our system includes three steps, we firstly retrieve web pages for a given query, extract plain texts from the web pages, then significant text units are generated from the texts using statistical and rule-based methods. These text units include sentences, named entities and significant phrases. Next we ranked these text units according to the vector space model[2]. Cosine similarity was used to compute the scores of extracted text units w.r.t. the query. Finally we out the ranked text units as iunits results. The three MANDATORY runs submitted have the following features. Run1 uses sentences and text units, Run2 uses name entities and significant phrases, while Run 3 combines Run 1 and Run 2, for comparison purposes.

The Reminder of this paper is organized as follows. Section 2 presents the implementation details of the system used in this task. Section 3 concludes this paper and speculates on future work.

2. SYSTEM ARCHITECTURE
Figure 1 shows an overview of our system. The system can be divided into three stages: Web page retrieval & plain text extraction, significant text units extraction and ranking of the text units.

For each query that iunits need to be generated, we firstly extracted plain text information from the relevant html files. These files were provided by the NTCIR organizers, ranked according to their relevance to the source query. Then significant text units were extracted, this includes sentences, named entities and significant phrases. Every text unit is weighted by the vector space model, given the query. Lastly, these text units were ranked by the weights in the descent order. We now discuss each stage in full.

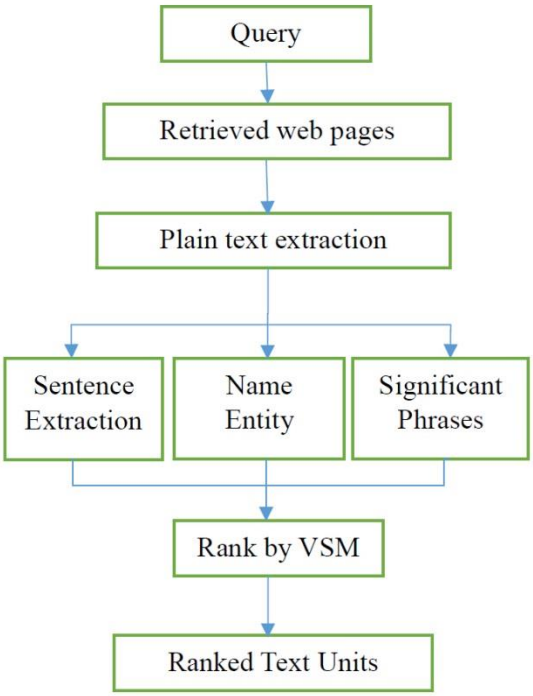

Figure 1. System Overview

2.1 Plain Text Extraction

The NTCIR organizers provided roughly more than 18,000 webpages for inits generation. These webpages are in HTML or XML format. In order to further processing of the information contained, we need to extract plain texts from these files. Here we used the HTML Parser version 2.0 [3] to parse the webpages and obtain the plain texts.

For every webpage, HTML Parser replenishes the informal labels and on the basis of labels' functions, takes out the content of the labels or put off the whole content. Particularly, for HTML label " <a >" ,we took out all information contained in the current and linked HTML pages. well , We then removed all hyperlinks, pictures, videos and similar information as advertisements were mostly embedded in these information units.

After the plain texts extracted, we rewrote them as the TREC format to be used in the information retrieval software. Heads of webpages were stored separately. It is worth to notice that although careful consideration was made when parsing the HTML files, still there are lots of noises in the text.

2.2 Text Units Extraction

2.2.1 Sentence Extraction

In the second step of the system, texts are further processed to be text units. The first group of text units is sentence. We have used two methods to extract sentences. The first one is based on heuristic rules and the second one is based on C99 [4].

For the first method, we uses MEDLINE data [5] as the training data. MEDLINE is a collection of 13 million plus citations into the biomedical literature maintained by the United States National Library of Medicine(NLM). We used some heuristic rules to determine sentence boundaries based on sets of tokens, a pair of flags, and boundary conditions. The important tokens are: Possible Stops, tokens that are allowed to be the final token in a sentence. Impossible Penultimates, tokens that may not be the penultimate (second-to-last) token in a sentence. Impossible Starts, tokens that may not be the first token in a sentence. The boundary conditions are determined according to Andrei Mikheev's paper [5].

The second method uses the C99 algorithm to segment texts into sentences [3]. C99 focuses on the similarity between all parts of a text. It uses a similarity measure in conjunction with divisive clustering to perform linear text segmentation.

2.2.2 Named Entity Extraction

Named entity recognition (NER) is the process of finding mentions of specified things in running text. Here we used a hidden Markov model (HMM) to perform chunking over tokenized character sequences. Specifically the whole process is based on an encoding of chunking as a tagging problem. A character language model HMM then handles the tagging, using character language models for each tag (state) in the HMM, and a maximum likelihood bigram transition model.

2.2.3 Significant Phrases Extraction

The significant phrases are another important text units to constitute iunts. These are different from the named entities and should use different extraction strategies.

We recognize two types of significance of interest. Collocations, these are phrases which are seen together more than you would expect given an estimate of how frequent each token is and how often they are seen together. Relatively New terms that occur significantly more often in the foreground corpus than they would be expected to from the background corpus. All phrases are extracted by using a tokenized language model. The provides a dynamic sequence language model which models token sequences with an n-gram model, and whitespace and unknown tokens with their own sequence language models.

However, the statistically generated phases were not good enough. As we do not have more time before submission, this part may affect the results a lot. Applying semantic information to extract phrases would be an interesting future work.

2.3 Text Units Ranking

In the final step of the system , we used the vector space model to rank the text units , Each query is represented as a query vector $\underset{V}{\rightarrow}(q)$, and each text unit is represented as a document vector $\underset{V}{\rightarrow}(d)$. Each dimension the vector corresponds to a separate term. If a term occurs in the document or a query, its value in the vector is non-zero. Several different ways of computing these values and we used the TF-IDF weighting scheme.

Next, it is easy to calculate the cosine similarity between the two vectors, by using the following formular:

$$score = \frac{\underset{V}{\rightarrow}(q) \cdot \underset{V}{\rightarrow}(d)}{\left|\underset{V}{\rightarrow}(q)\right| \cdot \left|\underset{V}{\rightarrow}(d)\right|}$$

3. RESULTS

In this section we discuss our evaluation results. As shown in Table 1 and Table 2, under the metrics defined by the task organizer, the overall results are unsatisfactory. This is due to several reasons as described below.

The first and the most important reason is that our team is consists of only undergraduate students. Due to the lack of human resources and limit amount of time, we did not finished several modules in the chain. Secondly, the training data used in our methods is coming from the MEDLINE corpus, which is clearly different from the domain of the corpus provided for the task. The third reason is that too many noises included in the results, these should be removed before output.

There are also some lessons learned. Though the poor performance obtained, we proved that the model for generating iunits for the mobile environment should be different from the normal texts. Also fast chaining some existing techniques should be work if they are properly used. Lastly, simply combining the different text units are not sufficient to deliver good results.

4. CONCLUSION

In this paper, we described in detail our participation in the NTCIR-11 MobileClick task. This is our first time participating in this task and all the programs were finished by third-year undergraduate students. We chained simple techniques for a quick implementation. Semantic-based text units extraction and more accurate statistical results should be considered in the future for better iunits generation.

Table 1. Mean (Std) nDCG results for iUnit Rerieval task

TeamID	RunID	#retrieved	nDCG@5	nDCG@10	nDCG@80	nDCG@400
KPNM	1	3599.2400 (2790.3009)	0.0647 (0.1562)	0.0583 (0.1274)	0.0747 (0.0928)	0.1467 (0.1054)
KPNM	2	3590.6000 (2789.9506)	0.0681 (0.1565)	0.0609 (0.1276)	0.0763 (0.0935)	0.1480 (0.1059)
KPNM	3	9.3200 (17.6867)	0.0068 (0.0276)	0.0081 (0.0247)	0.0058 (0.0154)	0.0057 (0.0150)

Table 2. Mean (Std) Q measure results for iUnit Rerieval task

TeamID	RunID	Q@5	Q@10	Q@80	Q@400
KPNM	1	0.0520 (0.1215)	0.0361 (0.0828)	0.0149 (0.0281)	0.0125 (0.0211)
KPNM	2	0.0573 (0.1249)	0.0396 (0.0857)	0.0156 (0.0291)	0.0131 (0.0216)
KPNM	3	0.0067 (0.0260)	0.0042 (0.0147)	0.0008 (0.0027)	0.0007 (0.0024)

5. ACKNOWLEDGMENTS

The work was supported by the National Natural Science Foundation of China under grant No. 61300129, and a project Sponsored by the Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry, China under grant number [2013] 1792.

6. REFERENCES

[1] Kato, M. P., Ekstrand-Abueg M., Pavlu V., Sakai T., Yamamoto T., Iwata, M. *Overview of the NTCIR-11 MobileClick Task*. Proceedings of the 11th NTCIR conference, 2014.

[2] Manning, C. D., Raghavan, P., Schutze, H. 2009. *Introduction to Information Retrieval*, Cambridge University Press.

[3] http://htmlparser.codeplex.com

[4] Choi, F.Y.Y. 2000. *Advances in domain independent linear text segmentation*, Proceedings of the 1st North American Chapter of the Association for Computational Linguistics Conference, NAACL 2000, Seattle, Washington, pp.26–33.

[5] http://www.nlm.nih.gov/bsd/pmresources.html

[6] Mikheev, A.. 2002. *Periods, Capitalized Words, etc.* Computational Linguistics 28(3):289-318.