A Naïve Bag-of-Words Approach to Wikipedia QA

Davide Buscaldi and Paolo Rosso
Dpto. de Sistemas Informticos y Computación (DSIC),
Universidad Politcnica de Valencia, Spain
{dbuscaldi, prosso}@dsic.upv.es

August 18, 2006

Abstract

This paper presents a simple approach to the Wikipedia Question Answering pilot task in CLEF 2006. The approach ranks the snippets, retrieved using the Lucene search engine, by means of a similarity measure based on bags of words extracted from both the snippets and the articles in wikipedia. Our participation was in the monolingual English and Spanish tasks.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: H.3.1 Content Analysis and Indexing; H.3.3 Information Search and Retrieval; H.3.4 Systems and Software

General Terms

Measurement, Algorithms, Experimentation

Keywords

Question Answering, Wikipedia

1 Introduction

Question Answering (QA) based on Wikipedia (*WiQA*) is a novel task, proposed as a pilot task in CLEF 2006. Wikipedia recently caught the attention of various researchers [5, 1] as a resource for the QA task, in particular for the direct extraction of answers. WiQA is a quite different task, since it is aimed at helping the readers/authors of Wikipedia rather than finding answers to user questions. In the words of the organizers[1], the purpose of the WiQA pilot is "to see how IR and NLP techniques can be effectively used to help readers and authors of Wikipages get access to information spread thoughout Wikipedia rather than stored locally on the pages". An author of a given Wikipage can be interested in collecting information about the topic of the page that is not yet included in the text, but is relevant and important for the topic, so that it can be used to update the content of the Wikipage. Therefore, an automatic system will provide the author with information snippets extracted from Wikipedia with the following characteristics:

- *unseen*: not already included in the given source page;

[1] http://ilps.science.uva.nl/WiQA/Task/index.html

- *new*: providing new information (not outdated);
- *relevant*: worth the inclusion in the source page.

In our previous work, we analyzed the link between Wikipedia and the Question Answering task from another point of view: using Wikipedia in order to help finding answers [2]. Although the task is completely different from the WiQA, we achieved some experience in extracting useful informations from Wikipedia by means of a standard textual search engine such as Lucene[2].

In this paper we describe the system we developed for the WiQA pilot task, that is mostly based on the Lucene search engine and our previous experience in mining knowledge from Wikipedia.

2 System Description

In Figure 1 we show the architecture of our system. A WiQA topic (i.e., a title of a source page to

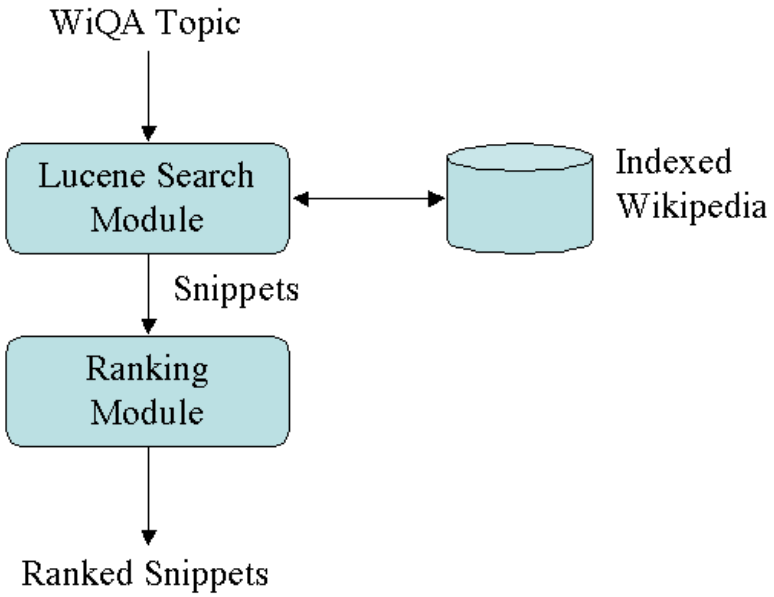

Figure 1: Architecture of our WiQA system.

be expanded) is passed as a phrase (e.g. "Patriarch of Alexandria") to the Lucene search module, which performs a standard search over the previously indexed collection of Wikipedia articles (the collections are the English and Spanish versions of the Ludovic Denoyer's Wikipedia XML corpora [3]). Lucene returns a set of snippets ranked accordingly to the usual $tf \cdot idf$ weighting.The snippets at this points include those belonging to the source page, which are removed from the result set. Subsequently, the *ranking module* rearranges the snippets in the optimal ranking, that is the final result. The following section describes how the ranking is done.

3 Bag-of-Words Ranking

In our previous works on Wikipedia and QA we attempted to emulate a human use of Wikipedia by means of simple Lucene queries based on pages' titles. The WiQA task presents almost the

[2]http://lucene.apache.org

same characteristics, every topics being itself a page title. Our approach is straightforward and needs only to pass to Lucene a phrase query containing the source page title.

The user behaviors emulated by our system are the following:

1. The user searches for pages containing the title of the page he is willing to expand;
2. The user analyzes the snippets, discarding the ones being too similar to the source page.

In the first case, passing the title as phrase to Lucene is enough for most topics. We observed that some topics needed a better analysis of title contents, such as the 7th of the English monolingual test set:"*Minister of Health (Canada)*", however these cases were not so many with respect to the total number of topics.

In the second case, the notion of *similarity* between a snippet and a page is the key for obtaining unseen (and relevant) snippets. Note that we decided to bound together relevance and the fact of not being already present in the page; we did not implement any method in order to determine whether the snippets contain outdated informations or not. In our system, the similarity between a snippet and the page is calculated by taking into account the number of terms they share. Therefore, we define the similarity $f_{sim}(p, s)$ between a page p and a snippet s as:

$$f_{sim}(p, s) = \frac{|p \cap s|}{|s|} \tag{1}$$

Where $|p \cap s|$ is the number of terms contained in both p and s, and $|s|$ is the number of terms contained in the snippet. This measure is used to rearrange the ranking of snippets by penalizing those being too similar to the source page. If w_l is the standard $tf \cdot idf$ weight returned by Lucene, then the final weight w of the snippet s with respect to page p is:

$$w(s, p) = w_l \cdot (1 - f_{sim}(p, s)) \tag{2}$$

The snippets are then ranked according to the values of w.

4 Results

The obtained results are shown in Table 1. The most important measure is the average yield, that is, the average number of "good" snippets per topic among top 10 snippets returned. The best average yield of the systems participating to the task was up to 3.4 for the English monolingual subtask. The *Average yield* is calculated as the total number of important novel non-repeated

Table 1: Results obtained by our system.

Language	Average yield	MRR	Precision
English	2.6615	0.4831	0.2850
Spanish	1.8225	0.3661	0.2273

snippets for all topics, divided by the number of topics. The *MRR* is the Mean Reciprocal Rank or the first important non-repeated snippet. The precision is calculated as the number of important novel non-repeated snippets, divided by the total number of snippets per topic. Our system returned always a number of snippets ≤ 10.

5 Conclusions and Further Work

The results obtained in Spanish were worse than those obtained in English. We suppose this is due to the smaller size of the Spanish Wiki (117MB in contrast to 939MB): information is spread

over a smaller number of pages, reducing the chance of obtaining valuable snippets. Our next steps in the WiQA task will be the inclusion of a similarity distance based on n-grams, resembling the one we already used for the JIRS Passage Retrieval engine [4]. We also hope to be able to participate in future cross-language tasks.

Acknowledgements

We would like to thank R2D2 CICYT (TIC2003-07158-C04-03) and ICT EU-India (ALA/95/23/2003/077-054) research projects for partially supporting this work.

References

[1] David Ahn, Valentin Jijkoun, Gilad Mishne, Karin Mller, Maarten de Rijke, and Stefan Schlobach. Using wikipedia at the trec qa track. In *TREC 2004 Proceedings*, 2004.

[2] Davide Buscaldi and Paolo Rosso. Mining knowledge from wikipedia for the question answering task. In *LREC 2006 Proceedings*, July 2006.

[3] Ludovic Denoyer and Patrick Gallinari. The Wikipedia XML Corpus. *SIGIR Forum*, 2006.

[4] Jose Gómez, Manuel Montes, Emilio Sanchis, and Paolo Rosso. A passage retrieval system for multilingual question answering. *LNAI-Springer Verlag*, 3658, 2005.

[5] Lucian Vlad Lita, Warren A.Hunt, and Eric Nyberg. Resource analysis for question answering. In *ACL 2004 Proceedings*. Association of Computational Linguistics, July 2004.