A Passage Retrieval System for Multilingual Question Answering

José Manuel Gómez Soriano\textsuperscript{1}, Manuel Montes y Gómez\textsuperscript{2}, Emilio Sanchis Arnal\textsuperscript{1}, and Paolo Rosso\textsuperscript{1}

\textsuperscript{1} Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia, Spain, \{jogomez,esanchis,prosso\}@dsic.upv.es \\
\textsuperscript{2} Laboratorio de Tecnologías del Lenguaje, Coordinación de Ciencias Computacionales, Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE), Mexico, mmontesg@inaoeep.mx

Abstract. In this paper we present a new method to improve the coverage of Passage Retrieval (PR) systems when these systems are employed for the Question Answering (QA) tasks. The ranking of passages obtained by the PR system is rearranged to emphasize those passages with more probability to contain the answer. The new ranking is based on finding the n-gram structures of the question that are presented in the passage, and the weight of the passages increases when they contain longer n-grams structures of the question. The results we present show that the application of this method improves notably the coverage of the classical PR system based on the Space Vectorial Model.

1 Introduction

A QA system is an application that allows a user to question in natural language a non-structured document collection in order to look for the correct answer.

Recently, at the Cross-Language Evaluation Forum (CLEF)\textsuperscript{3} which is a reference workshop to evaluate IR/QA systems operating on European languages, the task of multilingual QA has been incorporated. For this task, a QA system must be capable to accept questions in several languages and to search for the answers in a set of multilingual document collections.

In the multilingual QA task, it is very interesting the use of methodologies of document (or passage) retrieval as independent as possible of the language. This is the case of some pattern matching approaches, in which it is not necessary the use of a-priori knowledge of the languages.

Document or passage retrieval is typically used as the first step in current question answering systems [1]. In most of the QA systems, classical PR systems are used \cite{2,3,4,5}. The main problem that these QA systems have is due to the fact they use PR systems which are adaptations of classical document retrieval.

\textsuperscript{3} http://clef.iei.pi.cnr.it/
systems instead of being oriented to the specific problematic of QA. These systems use the question keywords to find relevant passages. For instance, if the question is *Who is the President of Mexico?*, these systems return those passages which contain the words *President* and *Mexico*.

In [6,7] it is shown that standard IR engines often fail to find the answer in the documents (or passages) when the question is presented with natural language questions. In [8] is presented a study of the performance of a QA system using just the top 20 passages showing that these passages contain the answer for only 54% of the question set [9].

Other PR approaches are based on Natural Language Processing (NLP) [10,9,11,12]. These approaches have the disadvantage to be very difficult to be adapted to other languages or to multilingual tasks.

The strategy of [13,14,15] is to search the obviousness of the answer in the Web. They run the user question into a Web search engine (usually Google⁴) with the expectation to get a passage containing the same expression of the question or a similar one. They suppose that due to the high redundancy⁵ of the Web, the answer will be written in several different ways including the same form of the question.

To increase the possibility to find relevant passages they make reformulations of the question, i.e., they move or delete terms to search other structures with the same question terms. For instance, if we move the verb of the question *Who is the President of Mexico?* and we delete the question term *Who*, we can produce the query *the President of Mexico is*. Thanks to the redundancy, we might find a passage with the structure *the President of Mexico is Vicente Fox*. [14] makes the reformulations carrying out a Part Of Speech analysis of the question and moving or deleting terms of specific morphosyntactic categories. Whereas [13] makes the reformulations doing certain assumptions about the verb position and the prepositional phrases boundaries on the question. The problem of these systems is that all the possible reformulations of the question are not taken into account.

With the methods used by [14] and [13] it would be very costly to realize all the possible reformulations since every reformulation must be searched by the search engine.

Our QA-oriented PR system makes better use of the redundancy bearing in mind all the possible reformulations of the question efficiently running the search engine with just one question as it will be described in detail in the next section.

Our system has the advantage to be language independent because it is based on processing the question and the passages without using any knowledge about the lexicon and the syntax of the corresponding language. In a language with not many differences between the question and the answer sentences, our system should work very well.

---

⁴ www.google.com

⁵ Certain repetition of the information contained in the collection of documents or Web, which allows, in spite of the loss of a part of this one, to reconstruct its content
This paper presents our PR system for QA. In Sect. 2, we describe the general architecture of the system. In Sect. 3 we discuss the results which were obtained using the Spanish CLEF questions and corpus, whereas in the last section we draw conclusions and future works.

2 System Architecture

The architecture of our PR system is shown in Fig 1.

![Diagram of the PR system](image)

**Fig. 1.** Main diagram of the PR system

Given an user question, it will be transferred to the search engine and n-grams extraction modules. Passages with the relevant terms (no stopwords) are found by the search engine using the classical IR system. Sets of unigrams, bigrams, ..., n-grams are extracted from the extended passages and from the user question. In both cases, \( n \) will be the number of question terms.

With the n-gram sets of the passages and the user question we will make a comparison in order to obtain the weight of each passage. The weight of a passage is related to the greater n-gram structure of the question which can be found in the passage itself. The weight of a passage will be longer if the passage contains greater n-gram structures of the question. This weight will be calculated using (1).

\[
Sim(d, q) = \frac{\sum_{j=1}^{n} \sum_{x \in Q_i} h(x, D_j)}{\sum_{j=1}^{n} \sum_{x \in Q_i} h(x, Q_j)} .
\]  

(1)
Where $Sim(d, q)$ is a function which measures the similarity of the set of \( n \)-grams of the question \( q \) with the set of \( n \)-grams of the passage \( d \). \( Q_j \) is a set of \( j \)-grams that are generated from the question \( q \) and \( D_j \) is the set of \( j \)-grams of the passage \( d \) to compare with. That is, \( Q_1 \) will contain the question unigrams whereas \( D_1 \) will contain the passage unigrams, \( Q_2 \) and \( D_2 \) will contain the question and passage bigrams respectively, and so on until \( Q_n \) and \( D_n \).

The result of (1) is equal to 1 if the longest \( n \)-gram of the question is in the set of passage \( n \)-grams.

For instance, if we ask "Who is the President of Mexico?" the system could retrieve two passages: one with the expression "...Vicente Fox is the President of Mexico..." and the other one with the expression "...José Luis Rodríguez Zapatero is the President of Spain...". Of course, the first passage must have more importance because it contains the 5-gram "is the President of Mexico", whereas the second passage only contains the 4-gram "is the President of", since the "is the President of Spain" 5-gram is not in the original question.

The function \( h(x, D_j) \) measures the relevance of the \( j \)-gram \( x \) with respect to the set of passage \( j \)-grams, whereas the function \( h(x, Q_j) \) is a factor of normalization. The function \( h(x, D_j) \) is defined by (2).

$$ h(x, D_j) = \begin{cases} 1 & \text{if } x \in D_j \\ 0 & \text{otherwise} \end{cases} \quad (2) $$

That is, (2) returns 1 if the \( j \)-gram \( x \) belongs to the set of passage \( j \)-grams and 0 otherwise. In this way, the denominator function \( h(x, Q_j) \) returns always 1 since the \( j \)-gram \( x \) always is in the set of \( j \)-grams \( Q_j \) and, therefore, the (1) can be simplified as:

$$ Sim(d, q) = \frac{\sum_{j=1}^{n} \sum_{x \in Q_j} h(x, D_j)}{\sum_{j=1}^{n} j} \quad (3) $$

The (3) has the disadvantage that it gives the same weight to all question terms, no matter if they are relevant words or not. Therefore, if we find the "the President of Mexico" 4-gram, it would have the same weight than "is the President of". In spite of that, this approximation improves the results considerably with respect to the classical vectorial model.

Of course, it would be interesting to give more weight to those \( n \)-grams which contain more relevant words. For this purpose, we would have to redefine (2) as:

$$ h(x, D_j) = \begin{cases} \sum_{k=1}^{[x]} w_k & \text{if } x \in D_j \\ 0 & \text{otherwise} \end{cases} \quad (4) $$

where \( w_1, w_2, \ldots, w_{[x]} \) are the associated weights of the terms of the \( j \)-gram \( x \). The associated weights should give an incentive to those terms which do not appear much in the document collection. Moreover, the weights should also
discriminate the terms against those (e.g., stopwords) which often occur in the
document collection. The next function was used in the experiments in order to
assign a weight to a term:

\[ w_k = 1 - \frac{\log(n_k)}{1 + \log(N)} \]  \hspace{1cm} (5)

where \( n_k \) is the number of passages in which the associated term to the weight
\( w_k \) appears and \( N \) is the number of system passages. We make the assumption
that stopwords occur in every passage (i.e., \( n_k \) takes the value of \( N \)). For instance,
if the term appears once in the passage collection, its weight will be equal to 1
(the greatest weight). Whereas if it is a stopword its weight will be the lowest.

3 Preliminary Results

The experiments detailed in this paper will be evaluated using a metric know as
coverage (for more details see [7]).

Let \( Q \) be the question set, \( D \) the passage collection, \( A_{D,q} \) the subset of \( D \)
containing correct answers to \( q \in Q \), and \( R_{D,q,n} \) be the top \( n \) ranked documents
in \( D \) retrieved by the search engine given a question \( q \).

The coverage of the search engine for a question set \( Q \) and a document
collection \( D \) at rank \( n \) is defined as:

\[ \text{coverage}(Q, D, n) \equiv \frac{|\{q \in Q | R_{D,q,n} \cap A_{D,q} \neq \emptyset\}|}{|Q|} \]  \hspace{1cm} (6)

Coverage gives the proportion of the question set for which a correct answer
can be found within the top \( n \) documents retrieved for each question.

Some experiments were carried out on the CLEF Spanish corpus which is
which we used are those of the 2003 Spanish QA task.

Due to the fact that whether the answers were included in the passages or not had to be manually verified, for the preliminary experiments we only analyzed the first 50 questions of the corpus. In the Fig. 2 is possible to appreciate the substantial improvement of our two models with respect to classical vector
model.

This figure shows that the percentage of found answers increases as the number
of considered passages does. The 75% of answers is found by both the simple
ngram model (2) and the termweight ngram model (4). We can appreciate that
the termweight ngram model is lightly better when the number of observed pas-
sages is higher whereas the simple ngram model works better when the number of
paragraphs is smaller. This is because the simple ngram model returns, in the first
paragraphs, the n-grams which contain most of the words of the question. Therefore,
it is more likely that these passages could contain the longest n-grams. As the
simple ngram model does not prioritize any word, when we increase the number
of paragraphs, we start to find passages with smaller n-gram structures. Moreover,
the shortest n-grams could be composed by words which are not necessarily relevant. On the other hand, the termweight ngram model has a smaller coverage in the first passages because it prioritizes more the weight of the terms than the n-gram structures which contain little relevant terms. In fact, the last model could provide relevant passages (with respect to the question) which do not necessarily contain the correct answer. The termweight ngram model improves in the final part due to the number of relevant longer structures decreases and the weight of the terms becomes more important.

In order to study the importance of redundancy and the coverage which our system could obtain, the 200 questions corpus was used. Fig. 3 shows the results which were obtained with the termweight ngram model for passages of 1 and 3 sentences.

In this figure we can appreciate that the coverage with passages of 3 sentences is higher than passages of 1 sentence (about 85% for the first 20 passages). This is due to the fact that often the answer is in the previous or following sentence.

Another important characteristic of our system is the redundancy of the correct answers which is about 7 (with passages of 3 sentences) and 5.5 (when only one sentence is in the passage). These results make our system suitable for those answer extraction methods based on redundancy [4,13,14,16,3,5,17].
4 Conclusions and Future Work

The n-grams comparison method which allowed us to obtain passages that contain the answer worked very well with the CLEF Spanish corpus, giving a very good coverage with a high redundancy (i.e., the correct answer was found more than once in the returned passages). Moreover, our system, does not make use any linguistic information and it is language independent. Therefore, we suppose it should allow to tackle the problem of the Multilingual QA since it will be able to distinguish what translations are better considering their n-gram structure in the corpus and it will discriminate the bad translations that are very unlikely that they appear. Our further interest is to prove the above assumption using as input several automatic translations and merging the returned passages. Those passages obtained with bad translations will have less weight than those which correspond to the correct ones.

As future work it would be necessary to see if this behavior stays for other corpora (e.g. the TREC\textsuperscript{6} English corpus and the CLEF transcribed dialogue corpus), as well as other languages. In addition, we would be interested in incorporating our system to a Multilingual QA system, to implement the modules of question analysis and extract the answer.

\textsuperscript{6} Text REtrieval Conference (http://trec.nist.gov/)
Acknowledgements

We would like to thank CONACyT for partially supporting this work under the grant 43990A-1 as well as R2D2 CICYT (TIC2003-07158-C04-03) and ICT EU-India (ALA/95/23/2003/077-054) research projects.

References