J Intell Inf Syst (2010) 34:113–134
DOI 10.1007/s10844-009-0082-y

Answering questions with an *n*-gram based passage retrieval engine

**Davide Buscaldi · Paolo Rosso ·
José Manuel Gómez-Soriano · Emilio Sanchis**

Received: 6 December 2007 / Revised: 26 February 2009 / Accepted: 26 February 2009 /
Published online: 13 March 2009

Abstract In this paper, we present a Question Answering system based on redundancy and a Passage Retrieval method that is specifically oriented to Question Answering. We suppose that in a large enough document collection the answer to a given question may appear in several different forms. Therefore, it is possible to find one or more sentences that contain the answer and that also include tokens from the original question. The Passage Retrieval engine is almost language-independent since it is based on n-gram structures. Question classification and answer extraction modules are based on shallow patterns.

Keywords Question answering · Information retrieval and extraction · Passage retrieval

1 Introduction

A Question Answering (QA) system is an application that allows a user to question, in natural language, an unstructured document collection in order to look for the correct answer. QA is sometimes viewed as a particular form of Information

D. Buscaldi (✉) · P. Rosso · E. Sanchis
ELiRF Research Group - Departamento de Sistemas Informáticos y Computación,
Universidad Politécnica de Valencia, Valencia, Spain
e-mail: dbuscaldi@dsic.upv.es

P. Rosso
e-mail: prosso@dsic.upv.es

E. Sanchis
e-mail: esanchis@dsic.upv.es

J. M. Gómez-Soriano
GPLSI Research Group - Departamento de Lenguajes y Sistemas Informáticos,
Universidad de Alicante, Alicante, Spain
e-mail: jmgomez@dlsi.ua.es

Retrieval (IR) in which the amount of information retrieved is the minimal quantity of information that is required to satisfy user needs. It is clear from this definition that QA systems have to deal with more complicated problems than IR systems: first of all, what is the "minimal" quantity of information with respect to a given question? How should this information be extracted? How should it be presented to the user? These are just some of the many problems that may be encountered.

A QA system can usually be divided into three main modules: Question Classification and Analysis, Document or Passage Retrieval, and Answer Extraction. These modules have to deal with different technical challenges, which are specific to each phase.

Question Classification (QC) is defined as the task of assigning a class to each question formulated to a system. Its main goals are to allow the answer extraction module to apply a different Answer Extraction (AE) strategy for each question type and to restrict the candidate answers. For example, extracting the answer to "*What is Vicodin?*", which is looking for a definition, is not the same as extracting the answer to "*Who invented the radio?*", which is asking for the name of a person. The class that can be assigned to a question affects greatly all the following steps of the QA process and therefore it is of vital importance to assign it properly. A study by (Moldovan et al. 2003) reveals that more than 36% of the errors in a QA system are directly due to the question classification.

The approaches to question classification can be divided into two categories: pattern-based classifiers and supervised classifiers.

Most QC systems use patterns and heuristic rules (Hermjakob 2001; Voorhees 1999, 2000, 2001) that are based on the detection of interrogative pronouns ("*wh-words*") and specific trigger words. The main problem with these systems is the amount of work needed for pattern formulation and definition because those patterns must capture any (or almost any) possible query reformulation.

Machine learning-based QC systems (Li and Roth 2002; Hacioglu and Ward 2003) were designed to define flexible systems that are capable of adapting to new languages and/or domains easily, but they are not well-considered as the pattern-based ones due to the lack of training data.

In both cases, a major issue is represented by the taxonomy of classes that the question may be classified into. The design of a QC system always starts by determining what the number of classes is and how to arrange them. Hovy et al. (2000) introduced a QA typology made up of 94 question types. Most systems being presented at the TREC[1] and CLEF[2] QA competitions use no more than 20 question types.

Another important task performed in the first phase is the extraction of the *focus* and the *target* or *topic* of the question. The focus is the property or entity sought by the question. The target or topic is the event or object the question is about. For instance, in the question "*How many inhabitants are there in Rotterdam*?", the focus is "*inhabitants*" and the target "*Rotterdam*". Systems usually extract this information using light Natural Language Processing (NLP) tools, such as Part-of-Speech (POS) taggers and shallow parsers (chunkers). POS taggers label words with their lexical category, i.e. noun, verb, adjective, etc.

[1] http://www.clef-campaign.org

[2] http://trec.nist.gov

A Passage Retrieval (PR) system is an IR application that returns pieces of texts (passages) which are relevant to the user query instead of returning a ranked-list of documents.

QA-oriented PR systems present some technical challenges that require an improvement of existing standard IR methods or the definition of new ones. First of all, the answer to a question may be unrelated to the terms used in the question itself, making classical term-based search methods useless. These methods usually look for documents characterized by a high frequency of query terms. For instance, in the question "*What is BMW*?", the only non-stopword term is "*BMW*", and a document that contains the term "*BMW*" many times probably does not contain a definition of the company. Another problem is to determine the optimal size of the passage: if it is too small, the answer may not be contained in the passage; if it is too long, it may bring in some information that is not related to the answer, requiring a more accurate Answer Extraction module.

The Answer Extraction phase is responsible for extracting the answer from the passages. Every piece of information extracted during the previous phases is important in order to determine the right answer. The main problem that can be found in this phase is determining which of the possible answers is the *right* one, or the most informative one. For instance, an answer for "*What is BMW?*" can be "*A car manufacturer*"; however, better answers could be "*A German car manufacturer*", or "*A producer of luxury and sport cars based in Munich, Germany*". Another problem that is similar to the previous one is related to the normalization of quantities: the answer to the question "*What is the distance of the Earth from the Sun*?" may be "*149,597,871 km*", "*one AU*", "*92,955,807 miles*" or "*almost 150 million kilometers*". These are descriptions of the same distance, and the Answer Extraction module should take this into account in order to exploit redundancy. Most of the Answer Extraction modules are usually based on redundancy and on answer patterns (Abney et al. 2000; Clarke et al. 2001; Aceves et al. 2005).

2 Contributions and related work

In this section we analyze the related work and discuss the contributions of our approach with respect to the existing ones. The automated QA system described in this paper is an extension of the JIRS PR system described in (Gómez et al. 2007a, b).

The JIRS PR system constitutes the most important advance introduced by our QA system. It is based on *n*-grams similarity measures instead of classical weighting schemes that are usually based on term frequency, such as tf·idf (Salton and Buckley 1988), or on statistical analysis, such as BM25 (Robertson et al. 2000). A preliminary evaluation of our system and its impact on QA was presented in (Gómez et al. 2005).

Most QA systems are based on IR methods that have been adapted to work on passages instead of the whole documents (Magnini et al. 2001; Aunimo et al. 2005; Vicedo et al. 2003; Neumann and Sacaleanu 2005). The main problems with these QA systems derive from the use of methods which are adaptations of classical document retrieval systems, which are not specifically oriented to the QA task. Table 1 shows the search engines used in TREC 2006 and the type of documents retrieved by such documents. Only one search engine, Indri, was specifically designed for Passage Retrieval. The others are publicly available systems that rely on standard IR models (Vector Space Model for Lemur and Lucene, BM25 for Xapian). Hovy

Table 1 Analysis of search engines used by TREC 2006 participants

Search engine	Sentence or passage	Passage extraction	Whole document	Snippet	Total (by engine)
Lucene	2		2		4
Lemur		1	2		3
Indri	3				3
Prise	1				1
Inquery		1			1
Xapian		1			1
Lucene+Google		1		1	2
Google+Yahoo				1	1
Total (by doc. type)	6	4	4	2	

Participations are also classified depending on the type of document retrieved. Sentence or Passage: the Search Engine returns a sentence or a portion of a greater document. Passage Extraction: the Search Engine returns a full document, and subsequently the passage containing the answer is extracted from this document. Whole document: the search engine returns an entire document. Snippet: the search engine returns a summary of the document

et al. (2000) and Roberts and Gaizauskas (2004) show that off-the-shelf IR engines (MG and Okapi, respectively) often fail to find documents containing the answer when presented with natural language questions.

There are other PR approaches that are based on NLP in order to improve the performance of the QA task (Greenwood 2004; Ahn et al. 2005; Hess 1996; Liu and Croft 2002). Some approaches include semantics in order to allow QA systems to answer specific types of questions (Narayanan and Harabagiu 2004). The main disadvantages of these approaches are that they are very difficult to adapt to other languages or to multilingual tasks, and they are usually slower than bag-of-words approaches (Bilotti et al. 2007). This issue is critical for commercial systems that can be accessed on the web, where short wait times are a key for success. Cao et al. (2005) argue that the use of NLP improves the results in the case of questions that are related to a specific domain, while pattern-matching approaches perform well only if it is possible to achieve great answer redundancy, such as in the web or large document collections. Although Roussinov et al. (2008) suggested that in web-based QA more sophisticated linguistic analysis could be the key to obtain an improvement over traditional keyword-based approaches, they did not demonstrate a decisive advantage of the former ones.

Web redundancy has been effectively exploited by 2004, Brill et al. (2001) and Buchholz (2001), to search the Web for the answer. They put the user question into a search engine (like Yahoo[3]) with the expectation of getting a passage that contains the same expression as the question or a similar one. To increase the possibility of finding relevant passages, they make reformulations of the question, i.e., they move or delete terms to search for other structures with the same question terms.

With the methods used by Brill et al. (2001) and Del Castillo et al. (2004) it would be very costly to perform all the possible reformulations since each reformulation must be searched for by the search engine. Our QA-oriented PR system makes

[3] http://www.yahoo.com

better use of the redundancy, taking into account all the possible reformulations of the question in order to effectively run the search engine with just one search.

Our PR method also has the advantage of being mostly language-independent because the question and passage processing does not use any knowledge about the lexicon or the syntax of the corresponding language. Our method proved to work very well in a language with few differences between the question and the answer sentences. We have tested the effectiveness of this method in several languages: English, three Romance languages (Italian, French and Spanish) (Gómez et al. 2005), Arabic (Benajiba et al. 2007) and Urdu (Gómez et al. 2007a). The test collections used for each language differ significantly in size and style, so we cannot determine if and how the characteristics of the languages (such as inflection) may affect the performance of the PR method.

3 The proposed approach

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

The user question is first handed over to the Question Classification and Analysis module and to the PR module. Then, the Answer Extraction module obtains the answer from the expected type, constraints and passages returned by these two modules. We describe in detail each module in the following subsections.

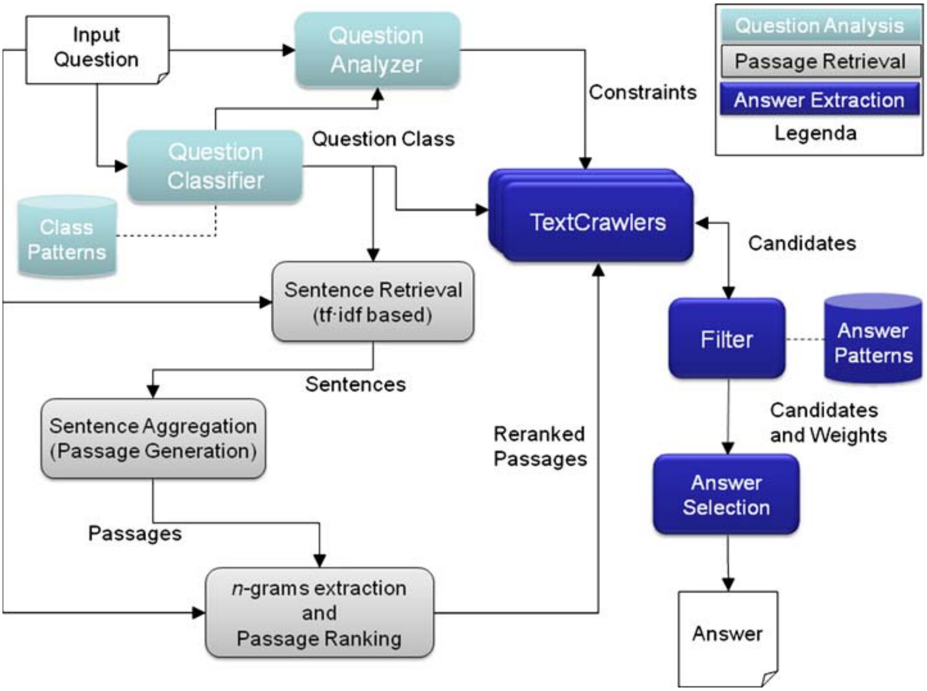

Fig. 1 Global architecture of our QA system

3.1 Question classification and analysis

The architecture of this module is shown in Fig. 2.

This module uses a pattern-based classifier; the patterns were derived by analyzing the CLEF QA test sets from 2003 to 2006. The system can handle questions for the classes shown in Table 2.

The patterns are organized in a three-level hierarchy, where one or more patterns define a category at each level of the hierarchy. The categories at level $i + 1$ are more specific than the category at level i. Each pattern is constituted by a regular expression, stored in an xml file together with the class names (as attributes). The xml format was adopted because it best fits the hierarchical organization of patterns. For instance, the portion of the pattern file containing the pattern for the QUANTITY.DIMENSION class is:

```
<pattern class = “QUANTITY”>
        ...
        <pattern class = “DIMENSION”>
                <ptrtext>How (high|wide|deep|far|long|heavy). + </ptrtext>
        </pattern>
        ...
</pattern>.
```

The questions that do not match any defined pattern are labeled with OTHER. This represents a label for a question that actually belongs to a class that has not been considered in the hierarchy. This is due to the fact that the categories were previously determined on the basis of the available answer extraction strategies. For instance, “Which *fruits* contain vitamin C?” can be classified as “FRUIT”. However, no strategy could be defined in the answer extraction module in order to discriminate fruit names from other names.

We evaluated the precision of the pattern-based classifier over the 2003–2004 Spanish test sets from CLEF, obtaining a 95.25% precision with respect to the classes defined above, including OTHER.

Together with the QC task, the system performs an analysis of the question in order to identify the constraints to be used in the AE phase. These constraints are made of sequences of words extracted from the POS-tagged query by means of POS patterns and rules. For instance, any sequence of nouns (such as “*ozone hole*” in the question “Where/AVQ is/VBZ the/DET ozone/NN hole/NN located/VBD”)

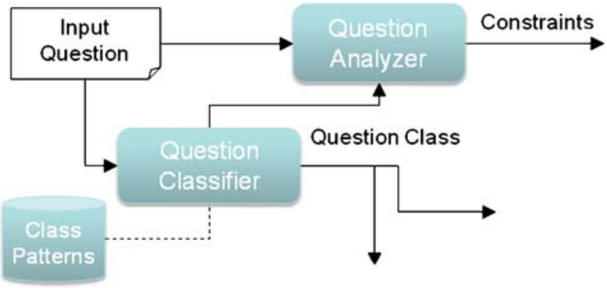

Fig. 2 Architecture of the question classification and analysis module

Table 2 Question classification categories

L1	L2	L3
Name	Acronym	
	Person	
	Title	
	Firstname	
	Location	Country
		City
		Geographical
Definition	Person	
	Organization	
	Object	
Date	Day	
	Month	
	Year	
	Weekday	
Quantity	Money	
	Dimension	
	Age	

is considered to be a relevant pattern. The POS-taggers used were the SVMTool (Giménez and Márquez 2004) for Spanish and the TreeTagger (Schmid 1994) for Italian and French.

We distinguish two classes of constraints:

1. a *target* constraint, which can be considered the object of the question;
2. *contextual* constraints, which hold the information that has to be included in the retrieved passage in order to be able to extract the correct answer.

For example, in the question "*How many inhabitants were there in Sweden in 1989?*", *inhabitants* is the target constraint, while *Sweden* and *1989* are the contextual constraints.

There is always only one target constraint for each question, but the number of contextual constraints is not fixed. For instance, in "*Who is Jorge Amado?*" the target constraint is *Jorge Amado*, but there are no contextual constraints.

Usually, the first relevant pattern found from the beginning of the question is taken as the target constraint, but this is not always true. For instance, in "*How did Jimi Hendrix die?*" the target constraint is *die*. Special rules were used to handle these exceptions.

3.2 Passage retrieval

Most current PR systems are not oriented to the specific problem of QA because they only take into account the question keywords to obtain the relevant passages (i.e. passages with the correct answer). The JAVA Information Retrieval System (JIRS)[4] is a passage retrieval system based on an *n*-gram based model (Clustered Keyword Positional Distance model, CKPD).

[4] The passage retrieval engine JIRS can be obtained at the following URL: http://sourceforge.net/projects/jirs/.

An n-gram in JIRS is a sequence of n adjacent terms extracted from a sentence or a question. JIRS is based on the premise that in a sufficiently large document collection, question n-grams should appear near the answer at least once. JIRS is able to find question structures in a large document collection quickly and efficiently by using different n-gram models. In order to do this, JIRS uses a traditional PR system as the first step and then searches for all possible n-grams of the question in the retrieved passages and rates them depending on the number and the weight of the n-grams that appeared in these passages. The architecture of this module is shown in Fig. 3.

The user question is passed to the sentence retrieval engine, which returns a ranked list of sentences with question keywords that are ranked according to the tf·idf weighting scheme. The maximum size allowed for the list is 1,000 results. These sentences are used to form the passages that are ranked by JIRS using the CKPD model and used later in the Answer Extraction phase. The passages are formed by attaching to each sentence in the ranked list one or more contiguous sentences of the original document. Let a document $d$be a sequence of n sentences $d = (s_1, \ldots, s_n)$. If a sentence s_i is retrieved by the search engine, a passage of size $m = 2k + 1$ is formed by the concatenation of sentences $s_{(i-k)} \ldots s_{(i+k)}$. If $(i - k) < 1$, then the passage is given by the concatenation of sentences $s_1 \ldots s_{(k-+1)}$. If $(i + k) > n$, then the passage is obtained by the concatenation of sentences $s_{(i-k-n)} \ldots s_n$. For instance, let us consider the following text extracted from the Glasgow Herald 95 collection (GH950102-000011):

> “Andrei Kuznetsov, a Russian internationalist with Italian side Les Copains, died in a road crash at the weekend. He was 28. A car being driven by Ukraine-born Kuznetsov hit a guard rail alongside a central Italian highway, police said. No other vehicle was involved. Kuznetsov’s wife was slightly injured in the accident but his two children escaped unhurt.”

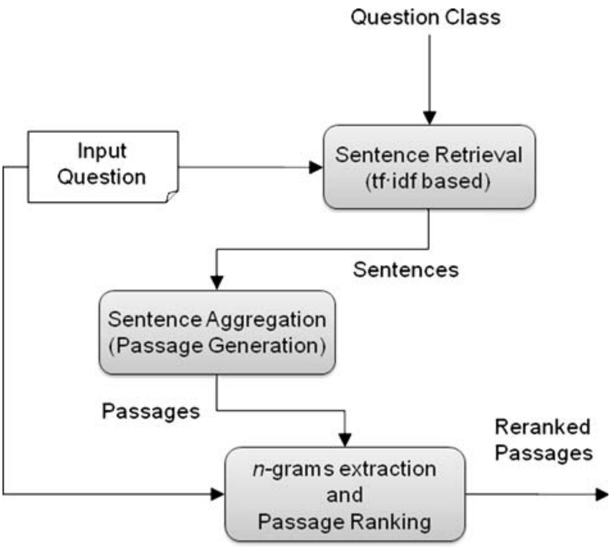
Question Class
Input Question
Sentence Retrieval (tf·idf based)
Sentences
Sentence Aggregation (Passage Generation)
Passages
n-grams extraction and Passage Ranking
Reranked Passages

Fig. 3 Architecture of the PR module

This text contains five sentences. Let us suppose that the question is "*How old was Andrei Kuznetsov when he died?*"; the search engine would return the first sentence as the best one (it contains "*Andrei*", "*Kuznetsov*" and "*died*"). If we set JIRS to return passages composed by three sentences, it would return "*Andrei Kuznetsov, a Russian internationalist with Italian side Les Copains, died in a road crash at the weekend. He was 28. A car being driven by Ukraine-born Kuznetsov hit a guard rail alongside a central Italian highway, police said.*". If we set JIRS to return passages composed by five sentences or more, it would return the whole text. This example also shows a case in which the answer is not contained in the same sentence, demonstrating the usefulness of splitting the text into passages.

Previous research work (Gómez et al. 2007a) demonstrated that almost 90% in answer coverage can be obtained with passages consisting of three contiguous sentences and taking into account only the first 20 passages for each question (Fig. 4). This means that the answer can be found in the first 20 passages returned by JIRS in 90% of the cases where an answer exists, if the passages are composed by three sentences.

We compared the answer coverage obtained by JIRS with the coverage that can be obtained by other IR engines that are based on the Vector Space Model: Lucene, a publicly available search engine, and IR-n, a passage retrieval engine (Llopis and Vicedo 2002). Figure 5 shows that JIRS obtains a better coverage even considering passages of just one sentence. The experiments carried out in (Gómez et al. 2007b) show that the *n*-gram model of JIRS can also be used to improve the Yahoo answer coverage, obtaining an improvement of approximately 20% with respect to the Yahoo search engine, using the questions of the Spanish CLEF 2005 test set.

The detection of sentence boundaries was carried out previously to the indexing phase. We used a heuristic based on the presence of uppercase letters or line

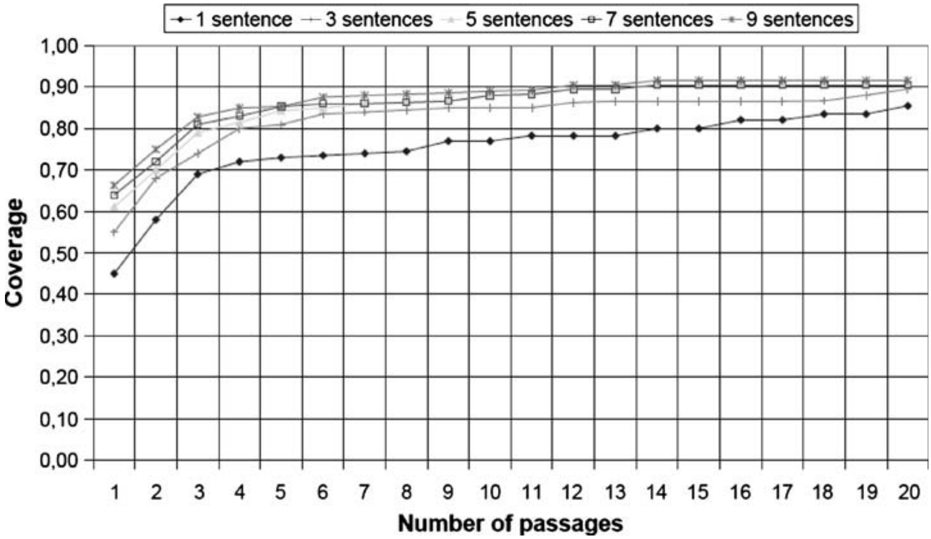

Fig. 4 Answer coverage provided by JIRS with up to 20 passages, each composed of up to nine sentences

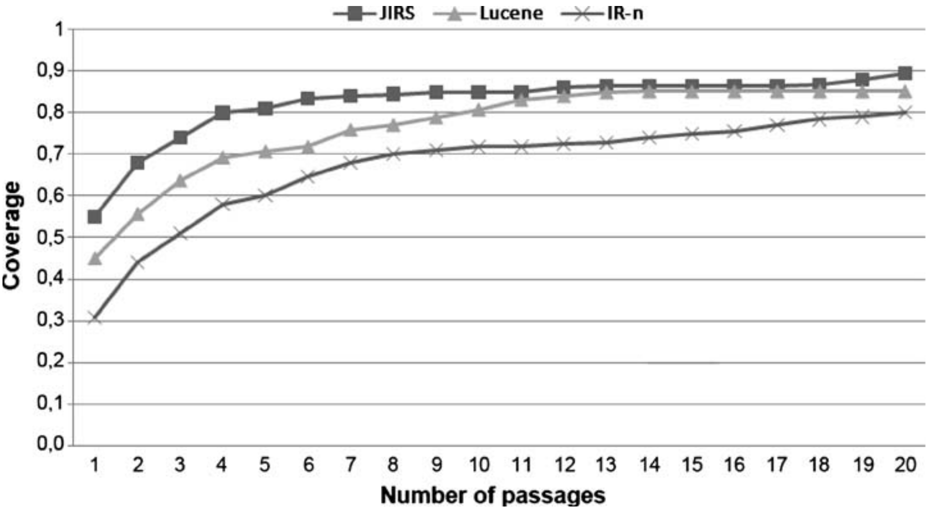

Fig. 5 Comparison of answer coverage obtained with up to 20 passages by JIRS, Lucene and IR-n

terminators after a period, an exclamation or a question mark. The errors in this phase are compensated by the way JIRS compose the passages.

After the composition of the passages, JIRS ranks them according to the *n*-grams that are included in each passage. The *n*-gram structures of every passage are extracted by the *n*-gram *Extraction* module. Only the *n*-grams that contain question terms are extracted. The weight of each passage is calculated according to the similarity between the question and the passage *n*-grams. The similarity of a passage with the question is greater if the passage shares longer *n*-gram structures with the question. The greater the similarity, the higher the passage is ranked by JIRS.

The weight of each term is calculated according to formula (1):

$$W_k = 1 - \frac{\ln(n_k)}{1 + \ln(N)} \tag{1}$$

where n_k is the number of sentences in which the term t_k appears, and N is the total number of sentences in the collection. If the term t_k occurs only once in the collection, its weight will be 1 (the maximum weight). We make the assumption that stopwords occur in every sentence (i.e., n_k takes the value of N); therefore, they will be assigned the minimum weight. Table 3 contains the number of stopwords for each language considered in JIRS.

The weight of an *n*-gram is calculated by the sum of its term weights:

$$h(x) = \sum_{k=1}^{n} w_k \tag{2}$$

where w_1,w_2, …,w_n are the term weights of the *n*-gram $x = t_1t_2\ldots t_n$.

Table 3 Number of stopwords in JIRS for each of the target languages

Language	No. of stopwords
Arabic	1,293
English	572
French	465
Italian	432
Spanish	191
Urdu	653

The similarity between the passage p and the query q is calculated using formula (3):

$$\mathrm{Sim}\,(p,q) = \frac{1}{\sum_{i=1}^{n} w_i} \times \sum_{\forall x \in P} h\,(x)\,\frac{1}{d\,(x, x_{\max})}. \tag{3}$$

Let Q be the set of n-grams of p composed only by question terms. Therefore, we define $P = \{x_1, x_2, \ldots, x_M\}$ as a sorted subset of Q that fulfills the following conditions:

$$\begin{aligned} &\forall x_i \in P : h\,(x_{i+1})\, i \in \{1, 2, \ldots M - 1\} \\ &\forall x, y \in P : x \neq y \Rightarrow T\,(x) \cap T\,(y) = 0 \\ &\min_{x \in P} h\,(x) \geq \max_{y \in Q-P} h\,(y) \end{aligned} \tag{4}$$

where $T(x)$ is the set of terms of the n-gram x.

The simplest measure of distance between two n-grams $d(x, x_{\max})$ in the text can be defined as the number of terms between them. However, this function has the disadvantage that it grows linearly and, therefore, the weight of the n-gram decreases too fast with respect to its distance from the heaviest n-gram. In order to address this issue, we use a logarithmic distance instead of the linear one. The distance function used in the CKPD model is the following:

$$d\,(x, x_{\max}) = 1 + \alpha \ln\,(1 + L) \tag{5}$$

where L is the number of terms between the n-gram $x_{\max}$ and the n-gram x of the passage ($x_{\max}$ is the n-gram with the maximum weight calculated in formula (2)). We have introduced the α constant to adjust the importance of the distance in the similarity equation. In previous experiments, we have determined that the best value for this constant is 0.1.

Example: Figure 6 shows an example with the question "*What is the capital of Croatia*?" and two passages returned by the search engine, both containing the keywords "*capital*" and "*Croatia*".

The first passage contains one n-gram, and its similarity value is simply the sum of its terms divided by the sum of the weights of all question terms. However, the second passage has two n-grams. The heaviest n-gram is "*the Croatia*" with a weight of 0.6. The other question n-gram is "*capital of*" with a weight of 0.3 and distance 7 from the heaviest n-gram. If we calculate the similarity value for both passages, we obtain a similarity value of 0.9 for the first passage and a similarity value of 0.7 for the second one.

In the CKPD model, if an n-gram does not contain any of the relevant terms, this n-gram is assigned a much smaller weight than another one that includes such a term.

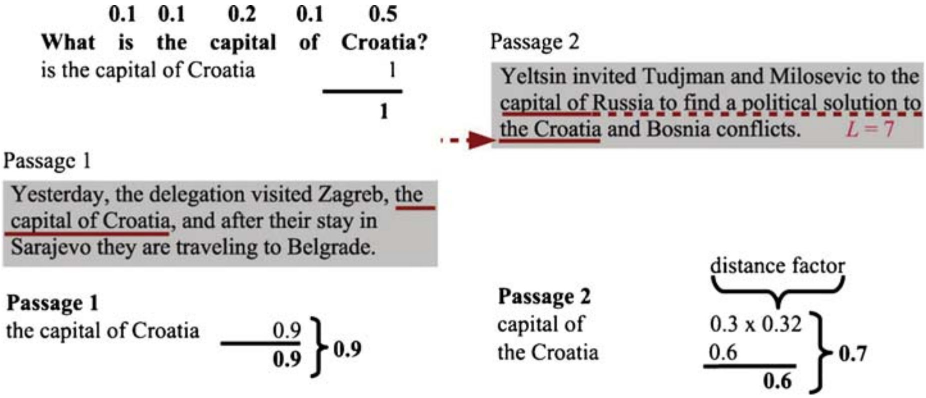
0.1 0.1 0.2 0.1 0.5
What is the capital of Croatia?
is the capital of Croatia
1
1
Passage 2
Yeltsin invited Tudjman and Milosevic to the capital of Russia to find a political solution to the Croatia and Bosnia conflicts.
L = 7
Passage 1
Yesterday, the delegation visited Zagreb, the capital of Croatia, and after their stay in Sarajevo they are traveling to Belgrade.
Passage 1
the capital of Croatia
0.9
0.9
0.9
distance factor
Passage 2
capital of
the Croatia
0.3 x 0.32
0.6
0.6
0.7

Fig. 6 Example of application of the JIRS n-gram model. In this example the constant α has been set to 1 for simplicity

Those n-grams that do not contain an irrelevant term (e.g. a stopword) have weights that are only reduced very slightly. Another characteristic of this model is that the similarity value is not affected by the term permutations. That is, the n-gram "*is the capital of Croatia*" is given the same weight as the n-gram "*the capital of Croatia is*" because it is composed of question terms. This aspect is very important for questions whose answers are usually formulated by means of permutations of question terms.

At the end of the passage ranking phase, the top m passages are passed to the Answer Extraction module. As shown in Fig. 4, we considered $m = 20$ to be enough to find an answer in 90% of the cases (i.e., 90% coverage). This is also an upper bound for the performance of the Answer Extraction module. If the QA system could obtain the same accuracy value of the answer coverage obtained by the PR module, it would mean that the ability of the Answer Extraction module to find answers is the same as that of a human.

3.3 Answer extraction

The architecture of this module is shown in Fig. 7.

The input of this module is constituted by the m passages returned by the PR module, the constraints and the expected type of the answer obtained through the Question Classification and Analysis module. A text analysis module, which we named *TextCrawler* because it slowly moves forward and back on the passage text, is instantiated for each of the m passages with a set of patterns for the expected type of the answer and a pre-processed version of the passage text. Some patterns may be used for all languages; for instance, when looking for proper names, the pattern is the same for all languages. The pre-processing of passage text consists in separating all the punctuation characters from the words and stripping off the annotations of the passage. It is important to keep the punctuation symbols because we have observed that they usually offer important clues for the individuation of the answer: for instance, it is more frequent to observe a passage containing "*The president of Mexico, Felipe Calderón*" than one containing "*The president of Mexico is Felipe Calderón*".

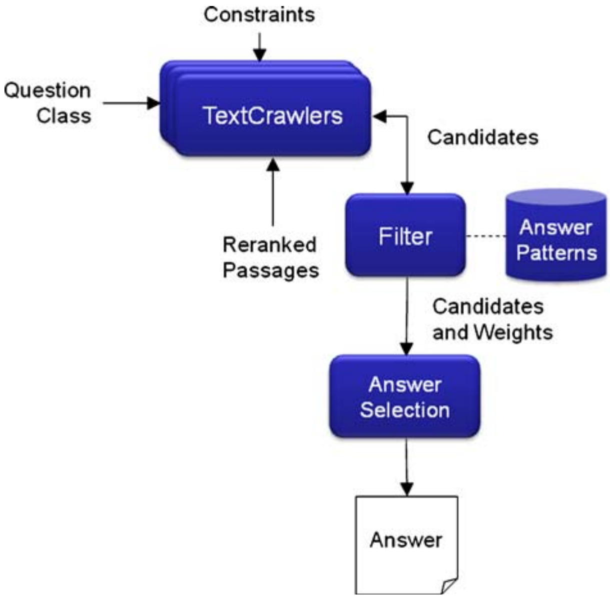

Fig. 7 Architecture of the AE module

The positions of the passages in which the constraints occur are marked before passing them to the TextCrawlers. Some spell-checking capability has been added in this phase by using the Levenshtein distance (Levenshtein 1966) to compare strings.

Each TextCrawler begins its work by searching for all the substrings of the passage that match the expected answer pattern and do not correspond to any constraint word. Thereafter it assigns a weight to each substring s that is found, depending on the positions of the constraints. Let us define $w_t(s)$ and $w_c(s)$ as the weights assigned to a substring s as a function, of its distance from the target constraints (6) and the context constraints (7) in the passage:

$$w_t(s) = \max_{0<k<|p(t)|} \left(\text{adjacent}\left(s, p_k(t)\right)\right) \tag{6}$$

$$W_c(S) \frac{1}{|c|} \times \sum_{i=0}^{|c|} \max_{0<j<|p(c_i)|} \left(\text{near}\left(s, p_j(c_i)\right)\right) \tag{7}$$

where c is the vector of contextual constraints, $p(c_i)$ is the vector of positions of the constraint c_i in the passage, t is the target constraint, and $p(t)$ is the vector of positions of the target constraint t in the passage. *Adjacent* and *near* are two proximity functions defined as:

$$\text{near}(s, p) = \exp\left(-\left(\frac{d(s,p)-1}{5}\right)^2\right) \tag{8}$$

$$\text{adjacent}(s, p) = \exp\left(-\left(\frac{d(s,p)-1}{2}\right)^2\right) \tag{9}$$

where the distance $d(s, p)$ between a candidate answer s and any constraint at position p in the passage is computed as:

$$d\,(s, p) = \min_{i=0, i=|s|} \sqrt{(s_i - p)^2} \quad (10)$$

where s_i indicates the position of the i-th word of the substring s. A high value of the proximity function adjacent(s, p) means that the substring s is adjacent to the word at the position p, and a high value for near(s, p) means that the substring s is "not far" from the word at position p. The 2 and 5 values roughly indicate the range within the position p where the words are considered really "adjacent" and "near" and have been chosen after some experiments with the CLEF 2003 QA Spanish test set. These two gradations were determined on the basis of the following observation: an answer usually appears *adjacent* to the target and *near* the contextual constraint. Since the candidates are weighted according to their proximity to question keywords, we used passages composed of three sentences, which provided a good compromise between keyword density and the probability to find a good candidate answer (c.f. Fig. 4).

We carried out some experiments in order to determine whether or not the exponential distance functions *near* and *adjacent* are better than a linear measure and whether or not the discrimination between *adjacent* and *near* is useful. The experiments were carried out on the Spanish 2005 and 2006 test sets. The results are shown in Table 4.

In the case of the 2005 Spanish test set (our first participation in CLEF) exponential measures allowed us to obtain a gain of 1.0% in accuracy over the linear measures; however, in the 2006 test set, the use of linear measures allowed to improve accuracy by 4.0%. If we calculate the results over both test sets, accuracy obtained by using linear measures is 1.5% better than exponential measures. Nevertheless, the differences observed in the single test sets clearly show that the effectiveness of the strategy used depends on the question set, and differences are not really significant. The results obtained with *near* and *adjacent* calculated in the same way (both functions are *near*) justify our design decision.

Finally, the weight is assigned to the substring s in the following way: if we find both the target constraint and the contextual constraints in the passage, the weight is calculated as the product of the weights obtained for every constraint; otherwise, it is constituted only by the weight obtained for the constraints found in the passage.

As an example of how the weighting works, let us consider the following question of type "QUANTITY":

How many inhabitants were there in Sweden in 1994?

Table 4 Comparison of results obtained using $1/d(s, p)$ instead of near and adjacent and the same formula for near and adjacent (both are the near function)

Test set	Accuracy (near and adjacent) (%)	Accuracy ($1/d(s, p)$) (%)	Accuracy (adjacent = near) (%)
Spanish 2005	33.5	32.5	29.0
Spanish 2006	35.0	39.0	32.5
2005 and 2006	34.2	35.7	30.7

Table 5 Adjacent and near values obtained for the passage "In 1994, there were 8 million inhabitants in Sweden and about 5 million inhabitants in Denmark"

S	Adjacent(s, $p_1(t)$)	Adjacent(s, $p_2(t)$)	Near(s, p(1994))	Near(s, p(Sweden))
8 million	1.0	0.00000478	0.527	0.852
5 million	0.00193	1.0	0.00790	0.697

The QA module returns *inhabitants* as the target constraint and *Sweden* and *1994* as contextual constraints. The following passage is returned by the PR module:

> In 1994, there were eight million inhabitants in Sweden and about five million inhabitants in Denmark.

Therefore, we obtain t = "*inhabitants*", c = ("*1994*", "*Sweden*") and the corresponding position[5] vectors p("*inhabitants*") = (8, 15), p("*1994*") = (2) and p("*Sweden*") = (10). The TextCrawler individuates two passage substrings matching the "QUANTITY" pattern, "8 million" and "5 million" ("*1994*" is discarded because it was recognized as corresponding to a contextual constraint). The distances are calculated as in formula (10), and the values obtained for near and adjacent functions are shown in Table 5. In Fig. 8 it can be observed how the weights vary in function of word positions in the passage.

The resulting weights are shown in Table 6. The substring "*8 million*" is selected as the candidate answer of the passage, and it is handed over to the filter module.

The *filter* module takes advantage of a knowledge base that is used to discard the candidate answers that are unlikely to match the correct answer. This knowledge base is made of two sets of patterns, one for the "allowed" candidate answer and another one for the "forbidden" ones.

For instance, when looking for definitions, the filter rejects all definitions ending or starting with a stop-word like *at*, *for*, *of*, *who*, *which*. When looking for a country, the candidate answer must match with one of the country names in four languages that have been included in the knowledge base. Ideally, any knowledge base can be integrated at this point, simply by introducing a proper category in the question analysis module. When the filter rejects a candidate, the TextCrawler provides it with the next best-weighted candidate, if there is one.

Finally, when all TextCrawlers end their analysis of the text, the Answer Selection module selects the answer to be returned by the system. The following strategies have been developed:

- *Simple Voting* (SV): The returned answer corresponds to the candidate that occurs most frequently as passage candidate.
- *Weighted Voting* (WV): Each vote is multiplied by the weight assigned to the candidate by the TextCrawler and by the passage weight as returned by the PR module.
- *Maximum Weight* (MW): The candidate that maximizes the product of its candidate weight and the weight of the enclosing passage is returned.

[5] Note that the comma (,) is included in the position count.

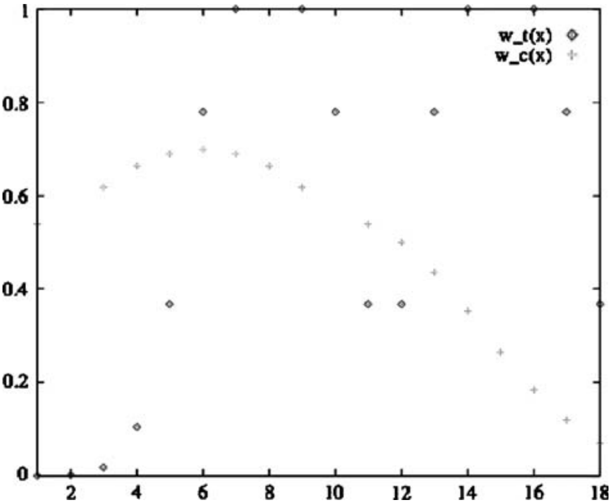

Fig. 8 Weights per word positions obtained for the passage "In 1994, there were 8 million inhabitants in Sweden and about 5 million inhabitants in Denmark." *si* marks the positions of the substrings found

- *Double Voting* (DV): The same as simple voting, but taking into account the second best candidates of each passage.
- *Top* (TOP): The candidate elected by the best weighted passage is returned.

SV is used for every "NAME" type question, whereas WV is used for all other types. For "NAME" questions, when two candidates obtain the same number of votes, the Answer Selection module looks at the DV answer. If there is still an ambiguity, then the WV strategy is used. For other types of question, the module uses the MW. TOP is used only when a confidence score is needed (for instance, to provide the user with some information on the reliability of the answer) in combination with other measures, depending on the number of passages returned and the averaged passage weight.

In our system, candidates are compared by means of a partial string match; therefore, *Boris Eltsin* and *Eltsin* are considered as two votes for the same candidate. Although it is possible to obtain bad matches, like *Tony Blair* and *Cherie Blair*, we obtained better results with relaxed matching over the Spanish 2003 test set (39.5% with complete match and 43% with partial match). We suppose that redundancy reduces the influence of matching errors most of the time. Later, the Answer Selection module returns the answer in the form that occurs most frequently.

Table 6 Weights obtained for the passage "In 1994, there were 8 million inhabitants in Sweden and about 5 million inhabitants in Denmark"

S	$W_t(s)$	$W_c(s)$	$W(s)$
8 million	1.0	0.689	0.689
5 million	1.0	0.352	0.352

wt target weight, *wc* contextual weight, *w*(*s*) final weight

4 Experiments

We carried out some experiments on the CLEF Spanish, Italian and French corpora. CLEF (Cross-Language Evaluation Forum) is the European reference workshop to evaluate IR and QA systems. The CLEF workshop is held every year to compare the participant systems in a wide range of categories, such as Mono-, Bi- and Multilingual Information Retrieval, Interactive Cross-Language Information Retrieval, Multiple Language QA, etc. An important extension of these tasks is the Multilingual QA, which accepts questions in any language on multilingual document collections. In this case it is important to use language-independent methodologies for the passage retrieval phase.

We submitted to the system the 1,800 questions from the CLEF 2004, 2005 and 2006 monolingual QA test sets of Spanish, Italian and French, plus 200 Spanish monolingual questions of the CLEF 2003 (Magnini et al. 2005; Vallin et al. 2006; Vicedo et al. 2003; Magnini et al. 2007), for a total of 2,000 questions. Two hundred of them are NIL questions (that is, questions that do not have an answer in the collection). Some questions are translations of questions in another language, and there are some questions that have been repeated in different editions of CLEF. Therefore, there are fewer unique questions.

JIRS was set up to index and search the standard CLEF collections. Table 7 shows the details of each of these collections.

Our system returns one answer and a document ID identifying the source document (not the passage) where the answer was found for each question. This document is used by CLEF evaluators to check whether or not a justification for the answer is present in the document. The answer is presented exactly as it appears in the passages, without rewordings, accordingly to CLEF guidelines.

The evaluation of the 2005 and 2006 results was provided directly by CLEF organizers, since we participated in the QA exercise. The answers for 2003 and 2004

Table 7 Characteristics of the indexed CLEF document collections

Language	Collection	Size (MB)	Number of docs	Avg. doc size (bytes)	Avg. doc size (terms)
Spanish	EFE 1994	469	215,738	2,281	337
	EFE 1995	531	238,307	2,336	345
	Spanish tot.	1000	454,045	2,308	341
French	Le Monde 94	143	44,013	3,417	510
	Le Monde 95[a]	141	47,646	3,108	472
	ATS 94	76	43,178	1,841	277
	ATS 95	78	42,615	1,910	288
	French tot.	438	177,452	2,569	386,75
English	LA Times 94	387	113,005	3,587	576
	Glasgow Herald 95	141	56,472	2,626	439
	English tot.	528	169,477	3,106	507,5
Italian	La Stampa 94	176	58,051	3,179	482
	AGZ 94	73	50,527	1,522	228
	AGZ 95	73	48,980	1,561	235
	Italian tot.	322	157,558	2,087	315

[a] Le Monde 95 was not included in 2005

test sets were evaluated by hand, respecting the CLEF QA guidelines that establish four grades of correctness for the questions:

- *R*—right answer: the returned answer is correct and the document ID corresponds to a document that contains the justification for returning that answer.
- *X*—incorrect answer: the returned answer is missing part of the correct answer, or includes unnecessary information. For instance: Q: "*What is the Atlantis*?" → A: "*The launch of the space shuttle*". The answer includes the right answer, but it also contains a sequence of words that is not needed in order to answer the question. This error is usually due to pattern matching mistakes.
- *U*—unsupported answer: the returned answer is correct, but the source document does not contain any information allowing a human reader to deduce that answer. For instance, assuming the question is "*Which company is owned by Steve Jobs*?" and the document contains only "*…Steve Jobs'latest creation, the Apple iPhone…*", and the returned answer is "Apple", it is obvious that this passage does not state that Steve Jobs owns Apple. Usually this happens when the answer has a high redundancy and there are many justification snippets available.
- *W*—wrong answer.

The metric used for the evaluation is accuracy, which is computed as the number of right answers divided by the total number of questions.

Table 8 shows the results obtained for each of the test sets, compared with those obtained by the best system on the same test sets in past editions of the CLEF QA tracks. Since the 2003 evaluation rules allowed the systems to return three answers for each question, only the first answer returned was considered for the calculation of the accuracy for the 2003 Spanish reference system. There are small differences between the systems used in 2005 and 2006, but they are limited to the adaptation of the system to the guidelines and input formats of each year.

In the 2005 and 2006 Spanish task our system was outperformed only by the Inaoe systems (Pérez et al. 2006; Juárez et al. 2007), which used a definition database to answer definition questions and JIRS for other question types. Such systems obtained

Table 8 Accuracy obtained by our system over the test sets, compared with the best systems at the 2003, 2004, 2005 and 2006 CLEF QA tracks

Test set	*R*	*X*	*U*	Accuracy (%)	Reference system	R.S. accuracy (%)
Spanish 2003	86	2	0	43.0	Alicex032ms	26.5
Italian 2004	46	5	0	23.0	Irst04itit	28.0
French 2004	51	4	1	25.5	Gine042frfr	24.5
Spanish 2004	46	5	1	23.0	Aliv042eses	32.5
Spanish 2005	67	13	1	33.5	Inao051eses	42.0
Italian 2005	51	6	1	25.5	Tova052itit	27.5
French 2005	46	7	4	23.0	Syna051frfr	64.0
Spanish 2006	70	5	6	35.0	Pribe061eses	52.5
Italian 2006	53	6	5	28.2	UPV_061itit[a]	28.2
French 2006	60	10	1	31.6	Syna061frfr	67.89

R.S. Accuracy reference system accuracy

[a] *UPV_061itit* corresponds to our participation

Table 9 Results obtained over the test sets, according to answer types of questions

Test set	Loc (%)	Per (%)	Org (%)	Org_def (%)	Per_def (%)	Qty (%)	Tim (%)	Oth (%)	Overall (%)
Spanish 2003	62.2	60.4	30.7	30.8	n.a.	25.7	55.0	15.6	43.0
Spanish 2004	54.5	47.8	8.7	30.0	50.0	13.0	30.4	4.5	23.0
Italian 2004	36.0	32.1	35.3	0.0	33.3	16.6	48.0	3.6	23.0
French 2004	55.2	21.9	25.0	37.5	41.6	21.4	38.1	2.0	25.5
Spanish 2005	48.1	33.3	17.8	48.0	56.0	33.3	15.8	12.0	33.5
Italian 2005	26.1	25.8	12.5	44.0	56.0	4.0	20.0	3.7	25.5
French 2005	15.4	18.5	12.0	36.0	56.0	22.7	20.0	6.6	23.0

In 2006, CLEF organizers did not provide results grouped by answer types
n.a. not available

more than 80% accuracy over definition questions, whereas our system was not able to obtain more than 56% accuracy in the same type of questions.

Table 9 shows the accuracy for different expected types of answers. Table 10 contains also the results provided by CLEF for the NIL questions of the 2005 and 2006 tracks. The considered types are the ones officially indicated in the CLEF QA 2004 and 2005 editions, where *per* means "person name", *loc* means "location", *org* "organization", *org def* is "organization definition", *per def* stands for "person definition", *qty* means "quantity", *tim* "time expressions" and *oth* accounts for all other types. This classification was not provided to participants before the exercises.

The best results were obtained on the 2003 Spanish monolingual test set. This is due mainly to the fact that this collection contains a greater number of "location" and "person" questions with respect to the more recent test sets: these questions, which are generally considered to be easier, account for 46.5% of the Spanish 2003 test set versus an average 26.5% for the 2004 test sets.

The best results were obtained for the "location" type. This is mostly due to the knowledge base that provided the filter with semantic information and the ease in individuating candidate answers. On the other hand, the worst results were obtained for the "other" type of question, indicating that the pattern approach for Answer Extraction has severe limitations for questions where the answer cannot be identified by its typographical features. This is also the fundamental problem of open-domain Question Answering: currently available systems are not able to find answers for types of questions that have not been considered at design time. Their performance could be improved only by introducing new types of questions.

Table 10 Results obtained over the NIL questions for the 2005 and 2006 test sets

Test set	Prec.	Recall	NIL questions
Spanish 2005	0.19	0.30	20
Italian 2005	0.10	0.15	20
French 2005	0.06	0.10	20
Italian 2006	0.28	0.45	20
French 2006	0.34	0.30	30
Spanish 2006	0.33	0.65	40

Prec. number of times NIL was returned correctly/number of times NIL was returned, *Recall* number of times NIL was returned correctly/number of times NIL was the answer, *NIL questions* total number of NIL questions in the test set

5 Conclusions and further work

Giving computers the ability to answer "real" questions is still a long way ahead. Passage retrieval by itself may, in an interactive environment, represent the best choice for users. A potential contribution of this paper is to show how well clusters of keywords from questions and text passages match each other, especially in regard to distance between clusters and matching words, and using such measures to estimate accuracy of potential answers to questions. Our system takes advantage of a novel, language-independent, passage-based retrieval system that is paired with simple Question Classification and Analysis module and an Answer Extraction module. The Clustered Keywords Positional Distance model used for passage retrieval proved to be more effective than simpler keyword matching models. The system can be easily adapted to the cross-language task by providing it with one or more translations of the input question. It can also be expanded to more languages; however, it is necessary to be an expert in the target language in order to be able to define the patterns in an appropriate way. The obtained results show that further research and analysis should be carried out in order to identify which features of the different systems do better than others, and what factors lead to poor results. For instance, the effect of the adoption of different distance measures in the answer extraction phase is not clear.

As future work, we plan to investigate the use of distance measures based on logical or linguistic distance in the Answer Extraction module. We will modify the question analysis and Answer Extraction modules to tackle the questions that do not offer revelatory clues about the nature of the expected answer. Some additional categories could be added to the Question Classification and Analysis module; whereas a strategy based on NLP techniques like shallow parsing (or chunking) can be implemented in the Answer Extraction module such as to individuate answers that do not offer typographical characteristics that allow their identification. We would also like to compare the CKPD retrieval model to models that are based on deeper linguistic analysis. Another desirable future work could be to perform more experiments over the cross-language test sets, since our first participation in cross-language tasks (Spanish–English and vice versa) obtained the best results (Magnini et al. 2005).

Acknowledgements We would like to thank the TIN2006-15265-C06-04 research project for partially supporting this work.

References

Abney, S., Collins, M., & Singhal, A. (2000). Answer extraction. In *Proceedings of the sixth conference on applied natural language processing, applied natural language conferences* (pp. 296–301). Seattle, Washington: Morgan Kaufmann Publishers.

Aceves, R., Villaseñor, L., & Montes, M. (2005). Towards a multilingual QA system based on the web data redundancy. In *AWIC, 2005* (pp. 32–37). Lodz, Poland.

Ahn, K., Alex, B., Bos, J., Dalmas, T., Leidner, J. L., & Smillie, M. B. (2005). Cross-lingual question answering using off-the-shelf machine translation. In *Multilingual information access for text, speech and images, LNCS* (Vol. 3491, pp. 446–457). Springer.

Aunimo, L., Kuuskoski, R., & Makkonen, J. (2005). Finnish as source language in bilingual question answering. In *Multilingual information access for text, speech and images, LNCS* (Vol. 3491, pp. 482–493). Springer.

Benajiba, Y., Rosso, P., & Gómez, J. M. (2007). Adapting JIRS passage retrieval system to the Arabic. In *Proc. 8th int. conf. on comput. linguistics and intelligent text processing, CICLing-2007, LNCS* (Vol. 4394, pp. 530–541). Springer.

Bilotti, M. W., Ogilvie, P., Callan, J., & Nyberg, E. (2007). Structured retrieval for question answering. In *Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval (SIGIR'07)*, 23–27 July 2007 (pp. 351–358). Amsterdam, The Netherlands: ACM.

Brill, E., Lin, J., Banko, M., Dumais, S. T., & Ng, A. Y. (2001). Data-intensive question answering. In *Proceedings of the 10th text retrieval conference (TREC-10)* (pp. 393–400). Gaithersburg, Maryland.

Buchholz, S. (2001). Using grammatical relations, answer frequencies and the World Wide Web for TREC question answering. In *Proceedings of the 10th text retrieval conference (TREC-10)* (pp. 502–506). Gaithersburg, Maryland.

Cao, J., Roussinov, D., Robles-Flores, J. A., & Nunamaker, J. F., Jr. (2005). Automated question answering from lecture videos: NLP vs. pattern matching. In *Proceedings of the 38th Hawaii international conference on system sciences (HICSS 2005)*. Big Island, Hawaii, USA: IEEE Computer Society.

Clarke, C., Cormack, G., & Lynam, T. (2001). Exploiting redundancy in question answering. In *24th ACM SIGIR conference* (pp. 358–365).

Del Castillo, A., Gómez, M. M., & Villaseñor-Pineda, L. (2004). QA on the web: A preliminary study for Spanish language. In *Proceedings of the fifth Mexican international conference in computer science (ENC'04)* (pp. 322–328). Colima, Mexico.

Giménez, J., & Márquez, L. (2004). SVMTool: A general POS Tagger generator based on support vector machines. In *Proceedings of 4th LREC*. Lisbon, Portugal.

Gómez, J. M., Buscaldi, D., Bisbal, E., Sanchis, E., & Rosso, P. (2005). A multilingual question answering system using an **n**-grams based passage retrieval. In *Proc. workshop on natural language processing for information retrieval, 2nd Indian int. conf. on artificial intelligence (IICAI-2005)* (pp. 686–672). Pune, India.

Gómez, J. M., Buscaldi, D., Rosso, P., & Sanchis, E. (2007a). JIRS Language-independent Passage Retrieval system: A comparative study. In *Proc. 5th int. conf. on natural language processing (ICON-2007)*, 4–6 January. Hyderabad, India.

Gómez, J. M., Rosso, P., & Sanchis, E. (2007b). Re-ranking of Yahoo snippets with the JIRS Passage Retrieval system. In *Proc. workshop on cross lingual information access (CLIA-2007), 20th int. joint conf. on artificial intelligence (IJCAI-07)*, 6–12 January 2007. Hyderabad, India.

Greenwood, M. A. (2004). Using pertainyms to improve passage retrieval for questions requesting information about a location. In *Proceedings of the 27th annual international ACM SIGIR conference on research and development in information retrieval (SIGIR 2004)*. Sheffield, UK.

Hacioglu, K., & Ward, W. (2003). Question classification with support vector machines and error correcting codes. In *Proceedings of the 2003 conference of the North American chapter of the association for computational linguistics on human language technology: Companion volume of the proceedings of HLT-NAACL 2003–Short papers - Volume 2 (Edmonton, Canada, May 27–June 1, 2003)* (pp. 28–30). North American Chapter Of The Association For Computational Linguistics. Association for Computational Linguistics, Morristown, NJ. doi:10.3115/1073483.1073493.

Hermjakob, U. (2001). Parsing and question classification for question answering. In *Proceedings of the ACL 2001 workshop on open-domain question answering* (pp. 17–22). Toulouse, France.

Hess, M. (1996). The 1996 international conference on tools with artificial intelligence (TAI 96). In *Proc. conference on research and development in information retrieval (SIGIR 1996)*. Zürich, Switzerland.

Hovy, E., Gerber, L., Hermjakob, U., Junk, M., & Lin, C. (2000). Question answering in webclopedia. In *Proceedings of the ninth text retrieval conference (TREC-9)*. Gaithersburg, Maryland.

Juárez, A., Téllez, A., Delicia, C., Montes, M., Villaseñor, L. (2007). Using machine learning and text mining in question answering. In *7th workshop of the cross-language evaluation forum (CLEF 2006), LNCS* (Vol. 4730). Springer 2007.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics, Doklady, 10*, 707–710.

Li, X., & Roth, D. (2002). Learning question classifiers. In *Proc. international conference on computational linguistics (COLING 2002)*. Taipei, Taiwan.

Liu, X., & Croft, W. (2002). Passage retrieval based on language models. In *Proceedings of the eleventh international conference on information and knowledge management (CIKM 02)* (pp. 375–382). McLean, Virginia.

Llopis, F., & Vicedo, J. L. (2002). IR-n: A passage retrieval system at CLEF-2001. Revised papers from the second workshop of the cross-language evaluation forum on evaluation of cross-language information retrieval systems (September 03–04, 2001). In C. Peters, M. Braschler, J. Gonzalo, & M. Kluck (Eds.) *Lecture notes in computer science* (Vol. 2406, pp. 244–252). London: Springer.

Magnini, B., Negri, M., Prevete, R., & Tanev, H. (2001). Multilingual question/answering: The DIOGENE system. In *Proceedings of the 10th text retrieval conference (TREC-10)*. Gaithersburg, Maryland.

Magnini, B., Vallin, S., Ayache, C., Erbach, G., Peñas, A., De Rijke, M., et al. (2005). Overview of the CLEF 2004 multilingual question answering track. In *Multilingual information access for text, speech and images, LNCS* (Vol. 3491, pp. 371–391). Springer 2005.

Magnini, B., Giampiccolo, D., Forner, P., Ayache, C., Osenova, P., Peñas, A., et al. (2007). Overview of the CLEF 2006 multilingual question answering track. In *Evaluation of multilingual and multi-modal information retrieval, LNCS* (Vol. 4730, pp. 223–256). Springer.

Moldovan, D. I., Pasca, M., Harabagiu, S. M., & Surdeanu, M. (2003). Performance issues and error analysis in an open-domain question answering system. *ACM Transactions on Information Systems, 21*, 133–154. doi:10.1145/763693.763694.

Narayanan, S., & Harabagiu, S. (2004). *Question answering based on semantic structures, international conference on computational linguistics (COLING 2004)* (pp. 693–702). Geneva, Switzerland.

Neumann, G., & Sacaleanu, B. (2005). Experiments on robust nl question interpretation and multi-layered document annotation for a cross-language question/answering system. In *Multilingual information access for text, speech and images, LNCS* (Vol. 3491, pp. 411–422). Springer 2005.

Pérez, M., Montes, M., López, A., & Villaseñor, L. (2006) The role of lexical features in question answering for Spanish. In *Accessing multilingual information repositories: 6th workshop of the cross-language evaluation forum, CLEF 2005, LNCS* (Vol. 4022). Revised Selected Papers. Springer 2006.

Roberts, I., & Gaizauskas, R. J. (2004). Evaluating passage retrieval approaches for question answering. In *Advances in information retrieval, 26th European conference on IR research (ECIR 2004)* (pp. 72–84). Sunderland, UK.

Robertson, E., Walker, S., & Beaulieu, M. (2000). Experimentation as a way of life: Okapi at TREC. *Information Processing & Management, 36*(1), 95–108. doi:10.1016/S0306-4573(99)00046-1.

Roussinov, D., Fan, W., & Robles-Flores, J. (2008). Beyond keywords: Automated question answering on the web. *Communications of the ACM, 51*(9), 60–65. doi:10.1145/1378727.1378743.

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management, 24*(5), 513–523. doi:10.1016/0306-4573(88)90021-0.

Schmid, H. (1994). Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the conference on new methods in language processing*. Manchester, UK.

Vallin, S., Magnini, B., Giampiccolo, D., Aunimo, L., Ayache, C., Osenova, P., et al. (2006). Overview of the CLEF 2005 multilingual question answering track. In *Accessing multilingual information repositories, LNCS* (Vol. 4022, pp. 307–331). Springer 2006.

Vicedo, J. L., Izquierdo, R., Llopis, F., & Munoz, R. (2003). Question answering in Spanish. In *Working notes of the Cross-Lingual Evaluation Forum (CLEF 2003)*. Trondheim, Norway.

Voorhees, E.M. (1999). The TREC-8 question answering track report. In *Proceedings of the eighth text retrieval conference (TREC-8)*. Gaithersburg, Maryland.

Voorhees, E. M. (2000). Overview of the TREC-9 question answering track. In *Proceedings of the ninth text retrieval conference (TREC-9)*. Gaithersburg, Maryland.

Voorhees, E. M. (2001) Overview of TREC 2001. In *Proceedings of the tenth text retrieval conference (TREC-10)*. Gaithersburg, Maryland.