Finding answers in the Œdipe system by extracting and applying linguistic patterns

Romaric Besançon, Mehdi Embarek and Olivier Ferret
CEA-LIST
LIC2M (Multilingual Multimedia Knowledge Engineering Laboratory)
B.P.6 - F92265 Fontenay-aux-Roses Cedex, France
{besanconr,embarekm,ferreto}@zoe.cea.fr

Abstract

This article presents the version of the Œdipe question answering system that was used by the LIC2M for its participation to the monolingual track dedicated to the French language of the CLEF-QA 2006 evaluation. It focuses more precisely on the main new aspect of the Œdipe system, *i.e.* the learning and the application of patterns for extracting short answers for definition questions.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: H.3.1 Content Analysis and Indexing — Linguistic Processing; H.3.3 Information Search and Retrieval — Search process; H.3.4 Systems and Software — Performance evaluation (efficiency and effectiveness); H.3.7 Digital Libraries

General Terms

Measurement, Performance, Experimentation

Keywords

Question answering, Monolingual question answering, Definition questions, Answer Pattern extraction

1 Introduction

We develop a question-answering system, the Œdipe system, that can be considered in its present state as a baseline system, and we progressively extend it by adding new modules both for enlarging its capabilities and improving its results. We have adopted an incremental approach in order to carefully test the impact and the interest of each new added module. The first version of the Œdipe system was developed for the EQUER evaluation campaign [1] and was a passage-based system for French [2]. This first version was extended for CLEF-QA 2005 [6] to extract short answers from passages. Our results in this evaluation showed that our strategy for processing factoid questions was reasonably successful but that our few heuristics for answering definition questions were not sufficient. Hence, our focus of our participation to the French monolingual track of the CLEF-QA 2006 evaluation was to improve the extraction of short answers for definition questions.

2 Overview of the Œdipe system

As the core of the system that we used for the CLEF-QA 2006 evaluation is identical to our CLEF-QA 2005 system, we will only summarize its main features in this section. More details about it can be found in [6]. As illustrated by Figure 1, the architecture of the Œdipe system is a very classical pipeline architecture: the question is first submitted to a search engine to retrieve a restricted set of documents. Both the question and the documents go through a linguistic processing for normalizing their words and identifying their named entities. The question is then analyzed to determine the expected type of the answer and the focus of the question. A two-step gisting process is applied to the documents retrieved by the search engine to locate where answers are the more likely to be found: passages are first delimited by relying on the density of the words of the question in documents, which is a basic criterion but is not computationally expensive; passage answers with a maximal size of 250 characters are then extracted from these passages by computing a score in a sliding fixed-size window. Finally, short answers[1] are extracted from the passage answers depending on their expected type.

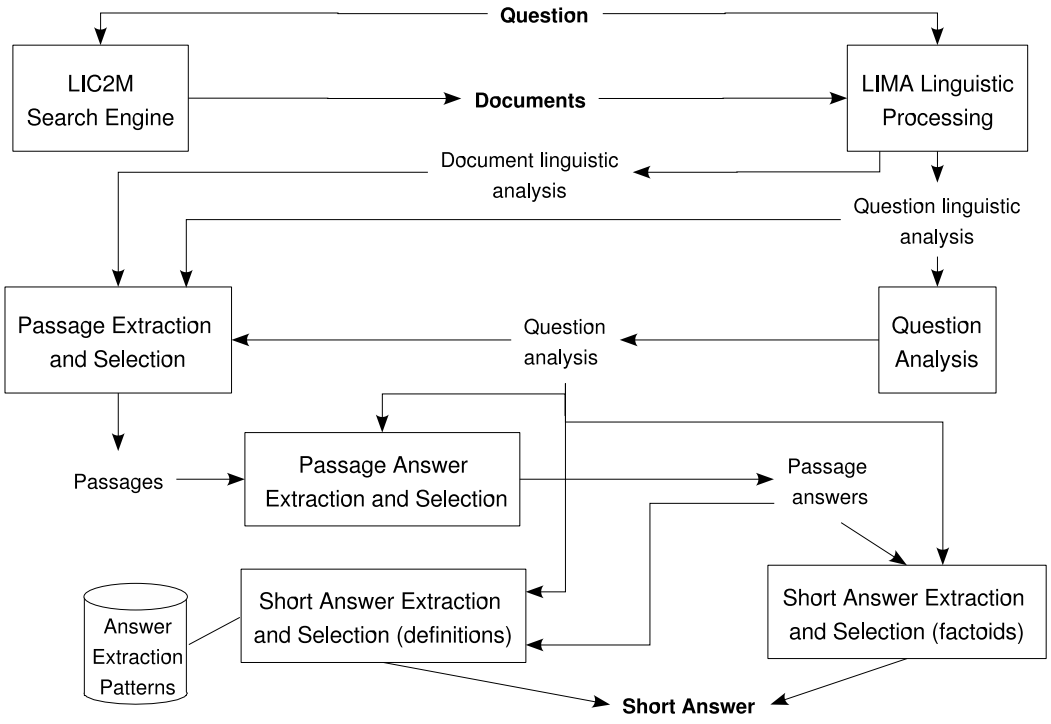

Figure 1: Architecture of the Œdipe system

More precisely, the Œdipe system is composed of the following modules:

Search Engine As for CLEF-QA 2005, we used the LIC2M search engine, that was already evaluated through the Small Multilingual Track of CLEF in 2003 [4] and 2004 [5]. Each question is submitted to the search engine without any pre-processing as the LIC2M search engine applies the LIMA linguistic processing to its queries. The only difference with CLEF-QA 2005 concerns the number of retrieved documents. In CLEF-QA 2005, a fixed number of documents (equal to 25) were selected. However, the LIC2M search engine is concept-based, which means that the documents it returns are not ranked according to a score but to the

[1] "Short answer" refers here to the exact answer and not to a fixed-size answer.

number of the concepts (*i.e.* multi-terms and named entities) of the query they contain. As a consequence, the results of the LIC2M search engine is a list of classes, each class being made of a set of documents that refer to the same set of concepts. As all the documents of a class are supposed to be equivalent, it is logically more justified to select all the documents of a class when it is possible. Moreover, we fixed a minimal number of 20 documents and a maximal number of 50 documents for each question. These constraints were fulfilled by applying the following algorithm:

$selectedDocuments \leftarrow \{\}$
$i \leftarrow 1$
while $card(selectedDocuments) < 20 \ \wedge \ i \leq card(classes)$ do
    $currentClass \leftarrow classes[i]$
    $i \leftarrow i + 1$
    if $card(selectedDocuments) + card(currentClass) \leq 50$
      then
        $selectedDocuments \leftarrow selectedDocuments \cup currentClass$
      else
        $randSelDocsNb = 50 - card(selectedDocuments)$
        $selectedDocuments \leftarrow selectedDocuments \cup$
                $random(currentClass, randSelDocsNb)$
    fi

where $random(s, n)$ is a function that randomly selects n elements from a set s. For CLEF-QA 2006, an average number of 33 documents by question were selected by this algorithm.

Linguistic Processing The linguistic processing of questions and documents is performed by the LIMA (LIc2m Multilingual Analyzer) linguistic analyzer [3]. Only a subset of its capabilities[2] is used for morpho-syntactic normalization of words, identification of content words and identification and typing of named entities. The named entities recognized by LIMA are restricted to the MUC named entities [10], that is to say *persons*, *locations*, *organizations*, *dates* and *times* and *numerical measures*, plus *events* and *products*.

Question Analysis In Œdipe, the question analysis module performs three different tasks:

- identifying the expected type of the answer;
- identifying the focus of the question ;
- identifying the significant words of the question and weighting their significance.

The result of the first task determines the strategy applied by Œdipe for extracting answers: if the expected type of the answer corresponds that a type of named entities that can be recognized by LIMA, Œdipe searches in the selected document passages for the named entity of that type whose context is the most compatible with the question. Otherwise, it assumes that the question is a definition question and applies specific linguistic patterns to extract possible answers (see Section 3).

The third task mainly consists in identifying the content words of a question and looking up their normalized information in a reference corpus to evaluate their specificity degree.

The second task is a new capability of Œdipe developed for answering definition question. In the version of Œdipe used for CLEF-QA 2006, it was achieved only for definition questions but from a more general viewpoint, determining the focus of a question could also be useful for improving the processing of factoid questions. The focus of a question is more precisely the part of the question that is expected to be present close to the answer. As for identifying the expected type of the answer, this task is achieved through the application of a specific set of rules, implemented as finite-state automata. Here are two examples of such rules:

[2]The LIMA analyzer can also perform term extraction or syntactic analysis but these capabilities are not exploited in Œdipe yet.

```
[Qu']::[est] [ce] [@Que] [$L_DET] *{1-30} [\?]:FOCUS:
[Qui]::[être$L_V] *{1-30} [\?]:FOCUS
```

The first rule identifies *Atlantis* as the focus of the definition question *Qu'est-ce que l'Atlantis ?*[3] while the second rule extracts *Hugo Chavez* as the focus of the question *Qui est Hugo Chavez ?*[4]

Passage Extraction and Selection This module first delimits candidate passages by detecting of the areas of documents with the highest density of words of the question. Then, a score is computed for each delimited passage by mainly relying on the number and the significance of the words of the question it contains. Candidate passages are ranked according to this score and passages whose the score is lower than a fixed threshold are discarded.

Passage Answer Extraction and Selection A passage answer is extracted from each selected passage by moving a window over the passage and computing a score at each position of the window according to its content and the expected type of the answer. The size of this window is equal to the size of the passage answers to extract (250 characters in our case) and its position depends on the question type: each content word of the target passage for a definition question ; each of its named entities that is compatible with the expected answer type for factoid questions. A fixed number of passage answers are selected according to their score.

Short Answer Extraction and Selection When the expected type of the question is a named entity, each passage answer is centered on a named entity of that type. Hence, the extracted short answer is directly that named entity. Its score is equal to the score of the passage answer. The way short answers are extracted for definition questions, which was the focus of our work for CLEF-QA 2006, is presented in the next section. As for passage answers, each short answer has a score and the short answers extracted from all the selected passage answers are ranked according to this score.

3 Answering definition questions

As mentioned before, the main improvement of the Œdipe system for CLEF-QA 2006 concerns its ability to answer questions whose the expected answer is not a named entity, and more specifically definition questions such as *What is X?* or *Who is X? What/Who* questions have proved to be difficult both because they are generally short and the search of an answer cannot be focused by a specific type of elements such as a named entity. Hence, trying to answer to this kind of questions with a basic approach leads to poor results as it was illustrated by Œdipe's results at CLEF-QA 2005.

Most of the question answering systems rely on a set of handmade linguistic patterns to extract answers for that kind of questions from selected sentences, as illustrated by [15]. Some work was also done to learn such patterns from examples, following some work in the Information Extraction field. One of the first attempt in this way was the work of Ravichandran and Hovy [14] which combines the use of the Web and suffix trees. Mining the Web for learning such patterns was also the solution adopted by [8]. [11] tested several machine learning algorithms for extracting patterns and [7] proposed a new algorithm for learning probabilistic lexico-syntactic patterns, also called *soft patterns*.

Work such as [14, 11] has proved that building a set of question-answer examples is quite easy, especially from the Web. It is why we have chosen for Œdipe to rely on lexico-syntactic patterns learnt from examples for answering definition questions. Moreover, this approach appears to be both more flexible and less costly when a question answering system must be extended to new domains.

[3] *What is Atlantis?*
[4] *Who is Hugo Chavez?*

3.1 Learning of definitional patterns

The algorithm we used to learn linguistic patterns for extracting answers to definition questions is an extension of the Ravichandran and Hovy's algorithm [14]. This extension was proposed by Ravichandran in [13] and learns multilevel patterns instead of surface patterns, that is to say patterns that can refer to different levels of linguistic information. More generally, this algorithm can be used for inducting patterns from texts dedicated to the extraction of various kinds of elements: semantic relations for populating knowledge bases [12, 9], answers in question answering systems or factual relations between entities in information extraction.

In our case, the induction of patterns is done from a set of example answers to definition questions. The basic element of a pattern can be the surface form of a word, its part of speech (POS) or its lemma. These three levels of information are obtained by applying the LIMA linguistic analyzer to the sentences containing the example answers. More precisely, the overall procedure for building up a base of patterns dedicated to the extraction of definitional answer is the following:

1. Building a corpus of example answer sentences. Unlike [14] or [8], our corpus of example answers was not built from the Web but came from the results of the EQUER evaluation and from the previous CLEF-QA evaluations. For each definition question of these evaluations, all the sentences containing a right answer to the question were extracted.

2. Application of the LIMA linguistic analyzer to all the answer sentences to get their three levels of linguistic information.

3. Abstraction of answer sentences. This abstraction consists in replacing in each answer sentence the focus of the question by the tag <*focus*> and the short answer to the question by the tag <*answer*>.

4. Application of the multilevel pattern-learning algorithm between each pair of sentences (see below).

5. Selection of the top P patterns on the basis of their frequency.

The multilevel algorithm for the induction of patterns is composed of two parts. The first one consists in calculating the minimal edit distance between the two sentences to generalize, which enables to calculate the number of edit operations (insertion, deletion and substitution) that are necessary to transform one sentence to the other one. The second step extracts the more specific multilevel pattern that generalizes the two sentences. We complete some of the alignments by adding two wildcard operators: (*s*) represents 0 or 1 instance of any word while (*any_word*) represents exactly 1 instance of any word. See Algorithm 1 for the precise algorithms that correspond to these two steps.

Here are some examples of the definitional patterns induced by this algorithm[5]:
<answer> (<focus>
<focus> (<answer>
<focus> , le */the/* <answer>
<focus> , un */a-an/*<answer>
<focus> être */to be/* L_DET_ARTICLE_INDEF <answer>
<focus> , (*any_word*) <answer>

3.2 Application of patterns to extract answers

The definition patterns learnt by following the procedure described in the previous section aims at extracting the short answers to questions about definitions. We integrated these definition patterns into the Œdipe question answering system by applying them after the selection of passage answers. This application consists in instantiating the definition patterns with the elements of the considered question resulting from its analysis and to align the instantiated patterns with the passage answer,

[5]The translation of lexical items in patterns are given as */translated lexical item/*.

Algorithm 1 Pattern-learning algorithm [12]

Consider a(1,n) and b(1,m) are two sentences of lengths n and m words.

Algorithm for calculating the minimal edit distance between sentences

```
D[0,0] = 0
for i = 1 to n do D[i,0] = D[i-1,0] + cost(insertion)
for j = 1 to m do D[0,j] = D[0,j-1] + cost(deletion)
for i = 1 to n do
  for j = 1 to m do
  D[i,j] = min (D[i-1,j-1] + cost(substitution),
           D[i-1,j] + cost(insertion),
           D[i,j-1] + cost(deletion))
Print (D[n,m])
```

where: D[n,m] is edit distance between these two sentences

Algorithm for optimal pattern retrieval

```
i =n, j = m
while i≠0 and j ≠0
  if D[i,j] = D[i-1,j] + cost(insertion)
    print (*s*), i = i-1
  else if D[i,j] = D[i,j-1] + cost(deletion)
    print (*s*), j = j-1
  else if a1i = b1j
    print (a1i), i = i - 1, j = j-1
  else if a2i = b2j
    print (a2i), i = i - 1, j = j-1
  else if a3i = b3j
    print (a3i), i = i - 1, j = j-1
  else
    print (*any_word*), i = i - 1, j = j-1
```

where :
- a1(1,n), a2(1,n) and a3(1,n) are the level 1 (lexical level), level 2 (lemmatized level) and level 3 (POS level) representations for the sentence a(1,n)
- b1(1,m), b2(1,m) and b3(1,m) are the level 1 (lexical level), level 2 (lemmatized level) and level 3 (POS level) representations for the sentence b(1,m)

first to determine if the pattern matches the passage answer and second, to extract the answer to the question if such a matching is detected. More precisely, the procedure is the following:

1. Instantiation of definitional patterns. This instantiation consists in replacing all the *<focus>* tags in patterns by the focus of the question as it was identified by the dedicated rules of the question analysis module (see Section 2).

2. Application of the LIMA linguistic analyzer to each passage answer selected by Œdipe for getting the three levels of linguistic information of patterns.

3. Extraction of short answers. This extraction is performed by aligning a candidate instanciated pattern with the passage answer. This alignment starts from the focus of the window and is verified word by word until the *<answer>* tag is reached in the pattern. If this verification process does not failed, the noun phrase in the passage answer that corresponds to the *<answer>* tag in the pattern is extracted and is considered as a possible short answer.

4. Selection of the top A short answers on the basis of their frequency.

The set of short answers extracted by using definition patterns are ranked according to their score, *i.e.* the number of patterns that extract them. The short answer with the highest score is returned as the answer to the considered question.

4 Results

We submitted one run of the Œdipe system for the CLEF-QA 2006 evaluation. For the 187 factoid (F), definition (D) and temporally restricted (T) questions, Œdipe returned 30 right answers, 3 unsupported answers and 6 inexact answers, which gives an overall accuracy of 16%. Moreover, the detection of a lack of answer by Œdipe was right for only one question among the four it marked.

Table 1: Comparison of the distributions of Œdipe's right answers at CLEF-QA 2005 and CLEF-QA 2006

	Factoid (F+T)		**Definition (D)**	
	# right answers	**accuracy**	**# right answers**	**accuracy**
CLEF-QA 2005	28	18.7	0	0.0
CLEF-QA 2006	15	10.3	15	36.6

At a quick glance, the results of Œdipe at CLEF-QA 2006 seem to be comparable, with a slight improvement, to its results at CLEF-QA 2005, where its overall accuracy was equal to 0.14 with 28 right answers for the F, D and T questions. However, Table 1 shows that the distributions of right answers are quite different for the two evaluations. The use of definitional patterns leads to a very significant improvement for definition questions but results for factoid questions, which were processed by exactly the same version of Œdipe as the CLEF-QA 2005 version, significantly decrease. The improvement for definition questions was expected but there is no clear explanation of the decrease of results for factoid questions, except that they were perhaps more difficult than CLEF-QA 2005 factoid questions.

Moreover, Table 2 shows that the question analysis module is not responsible of this decrease of Œdipe's results for factoid questions since its accuracy for CLEF-QA 2006 questions is higher than for CLEF-QA 2005 questions. It should also be noted that the focus was correctly identified for all the definition questions.

Table 2: Results of the question analysis module for CLEF-QA 2006 and comparison with CLEF-QA 2005

question type	# questions	# incorrect types	accuracy (2006)	acc. (2005)
Factoid (F+T)	146	9	93.8	86.0
Definition (D)	41	4	90.2	100

5 Conclusion

In this article, we have given an overview of the version of the Œdipe system that participated to the French monolingual track of the CLEF-QA 2006 evaluation and we have more particularly detailed its new aspects with regards to the CLEF-QA 2005 version, that is to say the learning from examples of lexico-syntactic patterns and their application for extracting short answers for definition questions. These new aspects proved to be interesting but there is still room for improvements. The first of them is certainly to integrate the syntactic analysis of LIMA in Œdipe, which would particularly enable to take each noun phrase as a whole and make extraction patterns more general. The second main improvement is to extend the use of extraction patterns to the processing of factoid questions. The results of Œdipe for that kind of questions are significantly lower than its CLEF-QA 2005 results and an explanation of this fact still have to be found. However, it is clear that using such patterns would be an interesting means to focus the search of a named entity answer in passages.

References

[1] Christelle Ayache. Campagne EVALDA/EQUER : Evaluation en question-réponse, rapport final de la campagne EVALDA/EQUER. Technical report, ELDA, 2005.

[2] Antonio Balvet, Mehdi Embarek, and Olivier Ferret. Minimalisme et question-réponse : le système Oedipe. In *12ème Conférence annuelle sur le Traitement Automatique des Langues Naturelles (TALN 2005)*, pages 77–80, Dourdan, France, June 2005.

[3] Romaric Besançon and Gaël Chalendar (de). L'analyseur syntaxique de LIMA dans la campagne d'évaluation EASY. In *12ème Conférence annuelle sur le Traitement Automatique des Langues Naturelles (TALN 2005)*, pages 21–24, Dourdan, France, June 2005.

[4] Romaric Besançon, Gaël Chalendar (de), Olivier Ferret, Christian Fluhr, Olivier Mesnard, and Hubert Naets. *4th Workshop of the Cross-Language Evaluation Forum, CLEF 2003*, chapter Concept-based Searching and Merging for Multilingual Information Retrieval: First Experiments at CLEF 2003, pages 174–184. Springer Verlag, 2004.

[5] Romaric Besançon, Mehdi Embarek, and Olivier Ferret. *Multilingual Information Access for Text, Speech and Images - 5th Workshop of the Cross-Language Evaluation Forum, CLEF 2004*, volume 3491/2005 of *Lecture Notes in Computer Science*, chapter Integrating New Languages in a Multilingual Search System Based on a Deep Linguistic Analysis, pages 83–89. Springer Berlin / Heidelberg, 2005.

[6] Romaric Besançon, Mehdi Embarek, and Olivier Ferret. *Multilingual Information Access for Text, Speech and Images - 6th Workshop of the Cross-Language Evaluation Forum, CLEF 2005*, volume 4022 of *Lecture Notes in Computer Science*, chapter The Œdipe System at CLEF-QA 2005. Springer Verlag, September 2006.

[7] Hang Cui, Min-Yen Kan, and Tat-Seng Chua. Generic soft pattern models for definitional question answering. In *28th Annual International ACM SIGIR Conference on Research and Development of Information Retrieval (SIGIR 2005)*, Salvador, Brazil, August 2005.

[8] Yongping Du, Xuanjing Huang, Xin Li, and Lide Wu. A novel pattern learning method for open domain question answering. In *International Joint Conference on Natural Language Processing (IJCNLP'04)*, pages 81–89, 2004.

[9] Mehdi Embarek and Olivier Ferret. Extraction de relations sémantiques à partir de textes dans le domaine médical. In *JOBIM 2006*, Bordeaux, France, July 2006.

[10] Ralph Grishman and Beth Sundheim. Design of the MUC6 evaluation. In *MUC-6 (Message Understanding Conferences)*, Columbia, MD, 1995. Morgan Kauffmann Publisher.

[11] Florent Jousse, Isabelle Tellier, Marc Tommasi, and Patrick Marty. Learning to extract answers in question answering: Experimental studies. In *CORIA*, pages 85–100, 2005.

[12] Patrick Pantel, Deepak Ravichandran, and Eduard Hovy. Towards terascale knowledge acquisition. In *International Conference on Computational Linguistics (COLING 2004)*, pages 771–777, Geneva, Switzerland, 2004.

[13] Deepak Ravichandran. *Terascale Knowledge Acquisition.* PhD thesis, University of Southern California, 2005.

[14] Deepak Ravichandran and Eduard Hovy. Learning surface text patterns for a question answering system. In *40^{th} Annual Meeting of the Association for Computational Linguistics (ACL'02)*, pages 41–47, Philadelphia, July 2002.

[15] M. M. Soubbotin and S. M. Soubbotin. Patterns of potential answer expressions as clues to the right answer. In NIST, editor, *TREC-10 Conference*, pages 175–182, Gathersburg, MD, 2001.