Coupling Named Entity Recognition, Vector-Space Model and Knowledge Bases for TREC-11 Question Answering Track

P. Bellot(1), E. Crestan(1,2), M. El-Bèze(1), L. Gillard(1), C. de Loupy(2)

(1) Laboratoire d'Informatique d'Avignon (LIA)
339 ch. des Meinajaries, BP 1228
F-84911 Avignon Cedex 9 (France)
{ patrice.bellot, eric.crestan, marc.elbeze, laurent.gillard }@lia.univ-avignon.fr

(2) Sinequa S.A.S.
51, rue Ledru Rollin
F-94200 Ivry-sur-Seine (France)
{ loupy, crestan }@sinequa.com

Abstract: In this paper, we present a question-answering system combining Named Entity Recognition, Vector-Space Model and Knowledge Bases to validate answers candidates. Applying this hybrid approach, for our first participation in the TREC Q&A.

Keywords: Question Answering, Named Entity Recognition, Vector-Space Model, Knowledge Bases

1. Introduction

Our approach combines a Named Entity Recognition System developed at Sinequa[1] and an answer retrieval system based on Vector Space model that uses some Knowledge Bases developed at the Laboratoire d'Informatique d'Avignon[2].

First, the Named Entity Recognition system is briefly described, including specific features (section 2). Then, a summarized description of the SIAC (Segmentation et Indexation Automatique de Corpus) information retrieval system is given (section 3). For the purpose of Question Analysis, several Question Taggings have been employed, they are exposed in section 4. The approach using Knowledge Bases is then depicted (section 5), with a summary of its coverage (section 5.3). Section 6 is devoted to the Question Ordering problem. Finally, we present several experiments in the frame of TREC-11 (section 7).

2. Named Entities

Detection of Named Entities (NE) is one of the key elements in the Question Answering task. In the past few years, there was a growing interest in NE analysis. Most current techniques for NE recognition are based on handcrafted finite state patterns [Appelt *et al.*, 1995; Weischedel, 1995], on Hidden Markov Model [Bikel *et al.*, 1999] or on Maximum entropy approach [Borthwick, 1999]

The NE analysis approach used in this task is based on a cascade of transducers. Some special features have been added to enhance the NE recognition. Among those features, a *normalization function* for normalizing proper noun occurrences in a text frame has been engineered, as well as a trivial *pronominal anaphora resolution* module. All these aspects are described further on.

For each type of NE, a transducer has been manually developed using a test corpus for validation. The transducer vocabulary is not only based on lexical information, but on semantic information too. For the purpose of NE analysis, we built several resources: list of words for entities like FIRST NAME, PROFESSION, CURRENCY and thesaurus for GEOGRAPHY for instance. Most of the expected answer types (presented in appendix 10.1) are NE recognized by our system, except for NPP entity (person names) hyponyms.

With regard to the output, XML has been used to represent the tagged documents, as shown below:

*"<NPP>**Brown**</NPP>, <PROF>**director**</PROF> of the <ORGAN><CITY>**Los Angeles**</CITY> **Centers for Alcohol and Drug Abuse**</ORGAN>."*

One can observe from the previous example that embedded entities are allowed.

[1] Sinequa S.A.S.: http://www.sinequa.com
[2] LIA: http://www.lia.univ-avignon.fr/

2.1. Normalization Function

The identification of all the occurrences of person names is a difficult task when performed by transducers only. Many reasons could be mentioned to explain this phenomenon. The most common case is when a LAST NAME is given without any FIRST NAME. We are also aware that our resource of FIRST NAME is not (and will never be) exhaustive. This prevents us from using this semantic information in order to detect person names. However, as observed by several authors, in most cases, person names are given at least once in full form (FIRST NAME followed by eventually a MIDDLE NAME and the LAST NAME). This appears to be exact when dealing with newspaper articles (their style obeying certain editorial rules).

In order to reduce the number of unrecognized person name occurrences, a straightforward algorithm was developed based on the previous observations. First, the LAST NAME parts of the detected person name are extracted. Then, the document is parsed again in order to detect all the LAST NAME occurrences that were forgotten by the transducer. This could be done thanks to the person name previously extracted. The additional person name could then be used by the other transducers in the sequence.

In the following example, a correct normalization of the person's name "*John Paloma*" is presented:

"The biggest problem is identifying where these people are," said ***John Paloma****, 36, one of the outreach workers.*

…

For that reason, ***Paloma*** *used to stash equipment around town -- for example, high atop public toilets.*

The biggest inconvenient of this technique is that an incorrect detection of a word as a last name will affect the rest of the document processing. This mainly occurs when a first name is ambiguous (e.g. *Rose*, *France*, …).

2.2. Pronominal Anaphora Resolution

Pronominal Anaphora is the most widespread type of anaphora. Resolving them could lead to an improvement in the Q&A task. For example, the following question expects an answer of type DATE:

"*When did president* ***Herbert Hoover*** *die ?*"

One of the top documents found on the Internet contains the answer to that question. However, the sentence containing the answer does not contain the key element "*Hoover*", but only the anaphora "*he*":

"*After his 1932 defeat,* ***Hoover*** *returned to private business. …*

He *died in New York City on October 20, 1964*."

Resolving this particular case would greatly help finding

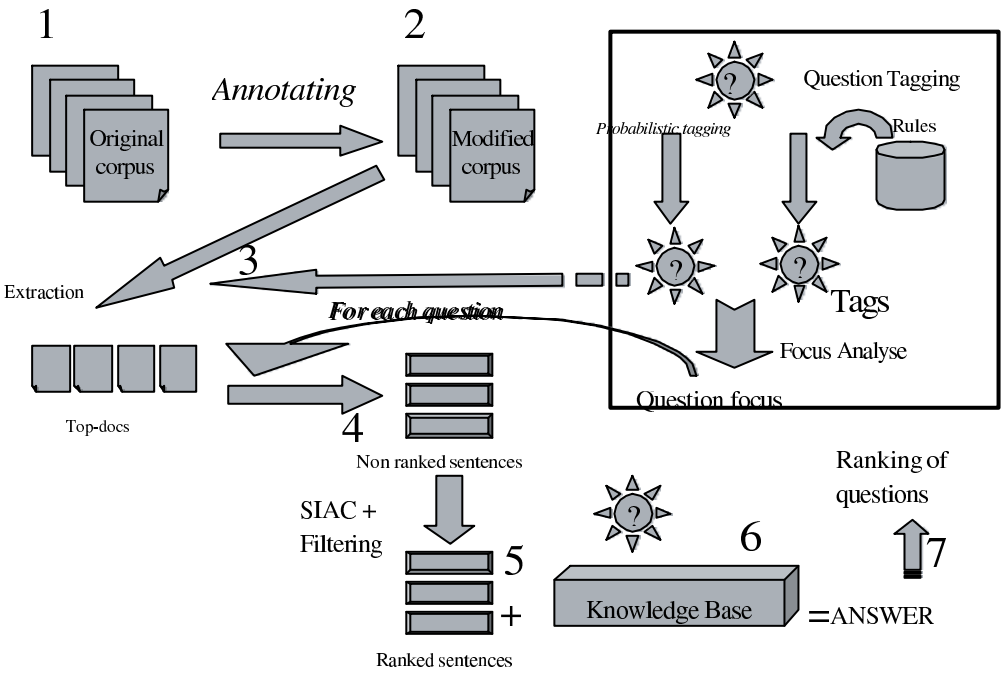

Figure 1 - From corpus to answers

the correct answer. For this reason, we have chosen to develop a Pronominal Anaphora Resolution, even though it is a "*naïve*" one. We decided to not resolve all the pronominal anaphora, but only for personal pronouns *he* and *she*, when they do not occur in quotations. The

approach is based on syntactic roles of person names. A person name used as a subject is a candidate for a future anaphora resolution (according to its sex).

Although this method is quite naïve, we achieved reasonable results on our test corpus. However, we have not yet evaluated the benefit of such a resolution in the whole QA task.

3. SIAC

The SIAC information retrieval system (Figure 1 shows the Java-based GUI of SIAC) has been designed to evaluate the classification and segmentation methods we work on [Bellot & El-Bèze, 2000]. During TREC-11 Q&A track, SIAC has been used to index and to rank sentences extracted from the top-docs documents by employing some classical methods: vector space model, cosine similarity and TFIDF weighting scheme.

Let Q be a question and S be a sentence. Let u be a lemma[3], N(u) be the number of sentences containing u in the set of top-docs related to question Q, TF(u) be the frequency of u and N be the total number of sentences extracted from top-docs. The similarity between Q and S is estimated by the cosine measure (formula 1):

$$\text{cosine}(S,Q)=\frac{\sum_{u\in S\cap Q} w_{u,S}.w_{u,Q}}{\sqrt{\sum_{u\in S} w_{u,S}^2.\sum_{u\in Q} w_{u,Q}^2.}} \quad (1)$$

with:

for document words: $$w_{u,S}=TF(u,S).\left(1-\log_2\frac{N(u)}{N}\right) \quad (2)$$

for query words: $$w_{u,Q}=TF(u,Q).\left(1-\log_2\frac{N(u)}{N}\right) \quad (3)$$

4. Question Tagging

We defined a hierarchical set of tags corresponding to the types of expected answers (see appendix 10-1). This set was built according to a manual analysis of the TREC-9 and TREC-10 Q&A questions.

For tagging TREC-11 Q&A questions, we have developed a rule-based tagger and we have employed a probabilistic tagger based on supervised decision trees [Béchet *et al.*, 2000] for the question patterns that did not correspond to any rule.

[3] We used the TreeTagger [Schmid, 1994, 1995] in order to obtain POS-tags and lemmas.

4.1. Rule-based tagger

Our rule-based tagger is a set of Perl scripts. The main input consists on an XML file that contains 156 manually built regular expressions. These regular expressions are not exhaustive since they are based on TREC-9 and TREC-10 questions only. The following is an extract of this file: the <CITY> tag defines 3 question patterns for which the expected answer is a city.

```
<CITY>
        <s> ZTRM <\/s> (In IN in )?(([Ww]hat WP
[Ww]hat)|([Ww]hich WDT [Ww]hich)) (\w+ JJ\w? \w+
)?((city NN city)|(seaport NN seaport)|(capital
NN capital)|(town NN town))
        <s> ZTRM <\/s> What WP What is VBZ be the
DT the (\w+ JJ\w? \w+)? ((city NN city)|(seaport
NN seaport)|(capital NN capital)|(town NN town))
        <s> ZTRM <\/s> Name VB name the DT the
(\w+ JJS \w+ )?city NN city
</CITY>
```

Among the 500 TREC-11 questions, 277 questions were tagged with the rule-based tool and 223 using decision trees.

4.2. Probabilistic Tagger

The probabilistic tagger is based on the named-entity recognizer presented during ACL-2000 [Béchet *et al.*, 2000]. This recognizer uses a supervised learning method to select their most distinctive features automatically select from a set of noun phrases, embedding named entities of different semantic classes,. The result of the learning process is a semantic classification tree (a particular decision tree introduced by [Kuhn & De Mori, 1996] to classify new strings from a corpus of tagged strings) that tags an unknown entity relying on its context. The adaptation of this recognizer to this task was realized by Fréderic Béchet: the tags are not linked to a particular entity but to the question as a whole.

To "grow" decision trees, one needs a sample corpus (manually tagged TREC-10 questions in our case) and a set of key features to split tree nodes. The list of features is generated from the training corpus. Each feature corresponds to a sequence of words and/or POS tags. Splitting is made by asking whether a selected feature matches a certain regular expression involving words, POS and gaps occurring in the TREC-11 question.

In order to evaluate our probabilistic tagger, we have subdivided the 500 TREC-10 questions into two sets: a learning set (259 questions) and a test set (150 questions). Over this 150 questions test set, we obtained a 68.5% precision level for 127 questions (23 questions were not tagged because the probability of the chosen tag was less than a minimal threshold).

For example, CITY is the tag chosen for question 1204 whereas all other candidate tags have a zero probability.

Question 1204:

```
sample_1204 <s> ZTRM </s> What WP What is VBZ be
the DT the cap=tal NN capital of IN of <UNK> NP
<UNK> ? ZTRM ? </s> ZTRM=</s> = CITY


sample_1204 ACTOR_ACTRESS 0 BIOGRAPHY 0 BIRD 0
BODY_PART 0 CITY=1 COMMON_WORD 0 COMPANY 0
CONTINENT 0 COUNTRY 0 COUNTY 0<=R> CURRENCY 0
DATE 0 DEFINITION 0 DEPTH 0 DIAMETER 0 DISTANCE 0
DURATION 0 EVENT 0 EXPANDED_ACRONYM 0 EXPLANATION
0 EXPLORATOR_RESEARCHER 0 FAMOUS_NPP 0
FAMOUS_PLACE 0 FAMOUS_PLACES 0 F=OWER 0 FOOD 0
HEIGHT 0 HEMISPHERE 0 INVENTOR 0 LENGTH 0 <=R>
MEDIA 0 MINERAL 0 MONEY 0 MOUNTAIN 0 MUSICIAN 0
NUMBER 0 =THER_NP 0 PERCENTAGE 0 PHRASE 0 PLANET
0 POLITICIAN 0 POPULATION 0 RIVER 0 SEA 0 SEASON
0 SPEED 0 SPORTSMAN 0 STAR 0=00 STATE 0 TEAM 0
TEMPERATURE 0 UNIV 0 VEGETAL 0 WEIGHT 0.0<=R> 0
WRITER 0 YEAR 0
```

In order to tag TREC-11 questions that were not tagged by our rule-based tagger, the learning was realized over the whole set of TREC-10 questions.

4.3. Filtering and Answer Extraction

The sentences allowing to answer questions do not necessarily contain a word of the questions. At the opposite, a sentence may contain some keywords of the question without being related to it. Thus, a classical retrieval scheme such as similarity computation in the vector space model is not sufficient.

In our case, the sentences from top-docs (the list of top-docs is the one given by NIST) are ranked by SIAC according to the similarity between them and the question. We had no time to implement a specific module to detect the focus of questions or to analyze their domain-dependent semantic properties. In order to filter sentences that probably did not contain the answer, we only kept those with a proper name appearing in the question[4] and those containing an entity of the same type than the expected answer type. This strategy prevents us from answering some questions (a NIL answer is given by the Q&A system because of the lack of proper names in the ranked sentences and/or in the question) but it enables us to select some answers more easily.

5. The Use of Knowledge Bases

We have chosen to take benefit from a set of knowledge DataBases (KDB) for several reasons, mainly: *i.)* Assess the reliability of our search engine, *ii.)* For a given relation between two NE, provide a bootstrap that may be used in the later steps of an iterative process (we plan to develop it soon). This process will be useful to extract other instances of such relations from full text collections. Therefore, it may be misleading to consider that the underlying idea of this component was to constitute a large Data Base of FAQ (Frequently Asked Questions), even though it has also been used as such.

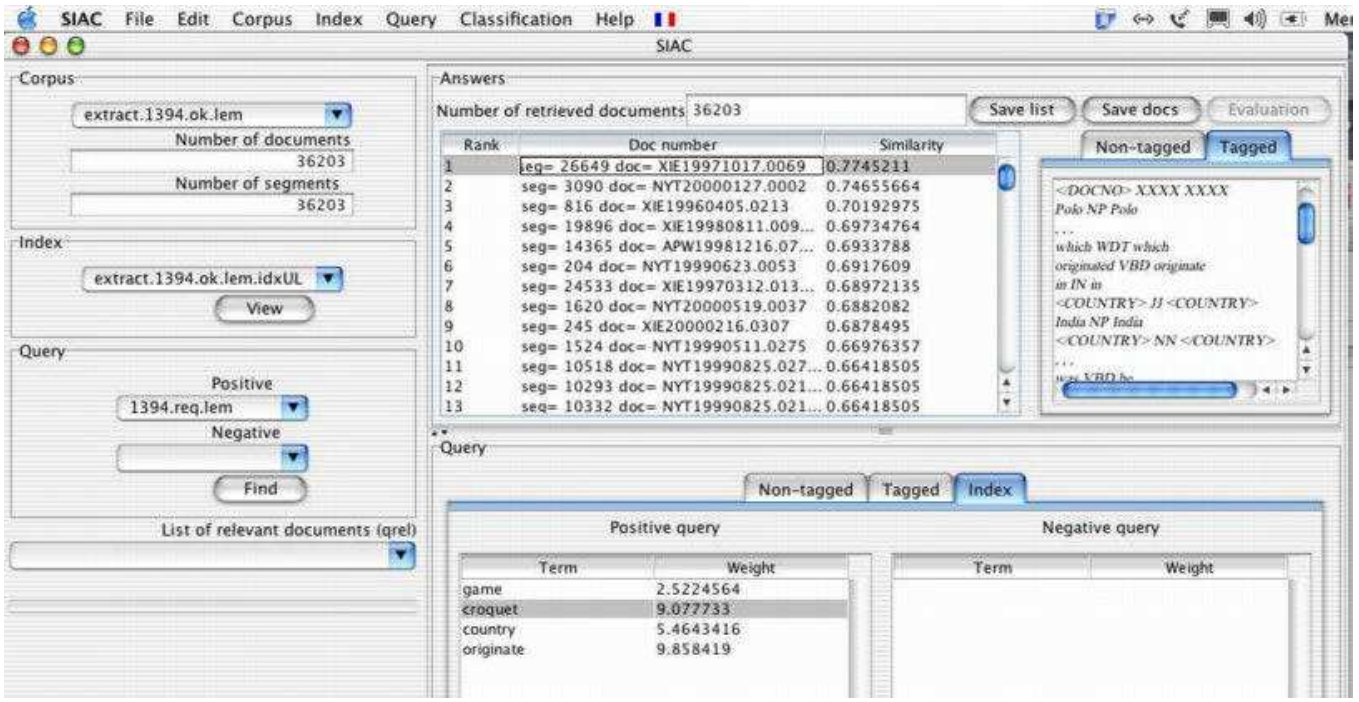

Figure 2 - The SIAC user interface

5.1. *Coupling SIAC and the use of KDB*

The link between a question and the production of the KDB component may be seen as a relation more than a function since the output may be multiple. To handle this (1-n) generation, we found it convenient to code the set of candidate answers using a regular expression. This regular expression is then applied on the sentences extracted by the search engine for 2 purposes: *i.)* Select the most likely answer *ii.)* Provide a support to the answer as required by the QA TREC protocol.

5.2. *Some characteristics of the KDB used*

5.2.1. USA topics

As it appears obviously from a quick analysis of the Q set (TREC-8 through TREC-11), several questions are focused on various attributes related to the United States of America. Thus, we have searched the net (mainly from the following url: http://www.50states.com/) in order to collect as many data related to these topics as possible. The coverage of such a "USA-centered" KDB is shown in Table 5.1.

TREC	8	9	10	11	Total
Motto	0+0	0+0	1+0	1+0	**2+ 0**
Flower	0+0	0+0	2+1	0+0	**2+1**
Song	0+0	0+0	0+0	1+0	**1+ 0**
Tree	0+0	1+0	0+0	0+0	**1+ 0**
Bird	0+0	1+0	3+0	1+0	**5+ 0**
Governor	0+0	0+0	1+0	3+0	**4+ 0**
Creation	0+0	1+0	3+0	2+0	**6+ 0**
Capital	1+5	2+1	0+5	1+6	**4+17**
Population	0+4	4+5	1+4	1+3	**6+16**
President	1+1	2+1	4+0	5+0	**12+ 2**
Total	**2+10**	**11+7**	**15+10**	**15+9**	**43+36**

Table 5.1. : Coverage of some KDB on the Q sets **#1** centered on the US **+ #2** not centered on the US

It was also an opportunity to cope with similar questions when they can be asked on other countries. In each cell of Table 5.1, the first number concerns US centered questions, the second one, other countries.

5.2.2. Book topics

In another direction, we have included in this process the relation book/author (*who wrote the book "title"?*). We have extracted from the web a list of bibliographical references. There are currently 15 800 entries in this specific KDB. Most of them come from the Pennsylvania University library and may be found at the following url: http://onlinebooks.library.upenn.edu/titles.html. We have also exploited shorter lists as the ones available at the url: http://www.state.nh.us/nhsl/bookbag/a.html.

TREC	8	9	10	11	Total
Book author	**2 / 3**	**5 / 7**	**1 / 1**	**0 / 2**	**8 / 13**

Table 5.2 : Coverage of the author KDB on the Q sets **#1** answers produced by KDB / **#2** questions on this topic

The formulation of a question is not always as precise as *who wrote the book "y"?*. Elliptic sentences as *who wrote "y"?* or *who is the author of "y"?* are more ambiguous. For instance, in Q8/196, *"Hamlet"* may be a movie or the famous play. The case is also encountered in Q11/1759: *"Fiddler on the Roof"* may be a novel or a musical. The novel was not in our KDB and it is a chance since only the musical has been considered as the correct answer by the judges. Whether we decide to enrich our resource or not, we have to take this kind of difficulty into account.

5.2.3. Archives

It was also natural to check whether questions found in TREC-11 were not already present in previous TRECs. In such case, the answer provided could be reused. Let us call an Archive A_i, a pair of two sets: (Questions, Answers) of $TREC_i$. Until we got A_{11} (the patterns of $TREC_{11}$), we have considered the following: for Q_8 use A_{9+10}, for Q_9 use A_{8+10}, for Q_{10} use A_{8+9}, for Q_{11} use A_{8+9+10} (1st line of Table 5.3). As shown in 2nd line of Table 5.3, the coverage on $Q_{8\text{-}10}$ does not increase a lot when A_{11} is also taken into account, except for Q_{10}.

TREC	8	9	10	11	Total
# Q	**0**	**4**	**5**	**5**	**14**
Including A_{11}	**0**	**4**	**9**	**5**	**18**

Table 5.3: Considering other Q sets as FAQ

Note that we did not search for a similar question but for exactly the same one. Therefore, some improvements can be made here.

5.2.4. Typos and Variants

Typos may be seen as a noise disturbing the canal between the input (Q) and the output (A). For a question such as Q10/1249/*Who wrote "The Devine Comedy"?* the relation (Dante – Divine Comedy) included in the KDB described in 5.2.2 could not be exploited. We have used the classical edit distance [Lowrance & Wagner, 1975] and the dynamic time wrapping method to find the optimal way to associate words as Divine and Devine. Penalty weights have been assigned to operations (substitution, omission, insertion), and a threshold has been empirically chosen in order to avoid confusion such as *Mexico/Monaco*. This procedure is not only useful to handle typos but also to cope with the numerous variants, which can be observed for the Proper Nouns transcription of Foreign Entities (there are, for example, more than 50 ways to write *Kahdafi*. This can be coded by a regular expression *[GK]h?ah?dd?h?ah?ff?i*).

[4] The proper name detection was realized according to the POS-tags.

As far as we want to take into account human factors, we have chosen to generate an answer where the graphemes involved are the most similar ones and not necessarily the ones used in the question. Our assumption is that the user will find more acceptable a system answering sometimes to another question than a system giving a wrong answer to his question.

5.3. KDB Summary

In the subsections 5.2.1 to 5.2.3, we have given some examples of the domains covered by the KDB we used. They correspond to about half of the answers currently supported by our KDB component. The second half concerns various topics such as rivers, mountains, Nobel's, hurricanes and so on. It is impossible to describe each of them in detail here, but it is interesting to see that the coverage is more or less the same on each TREC.

TREC	8	9	10	11	Total
Answer KDB	**22**	**73**	**64**	**61**	**220**
# Questions	**200**	**694**	**500**	**500**	**1 894**
%Q handled	**11.0**	**10.5**	**12.8**	**12.2**	**11.6**

Table 5.4: Global Coverage of 36 KDB

While for 12.2 % of the Q_1 set, the KDB are able to produce an answer, it is not possible to insert all of them in our run. As mentioned in section 5.1, we have also to match each answer with the output of SIAC. Sometimes (8 / 61 cases) the search engine is too silent, therefore the set of candidates may be empty. In more than half of the cases (35 / 61), it was possible to find a pattern matching the regular expression. For the remainder (18 / 61 cases), no match has been found in the sentences retrieved by SIAC.

6. Ordering answers

This year's QA track introduced newness in the evaluation measure in such a way that systems have to cope with the following principle: rank the answers from the most reliable to the less one. In order to take into account this requirement, our answers have been ordered according to results provided by the use (or non-use) of knowledge databases (KDB) – as a way to validate an answer – and by the question classifier output. So, for each question, the question classifier assigns one (or several) expected NE(s) and its (their) corresponding confidence(s). If it cannot be decided which of the 44 available entities should be responsive, the question is tagged as "unknown". From these points, our ordering strategy can be summarized schematically as follows: divide the Q-set in three main groups,

- Q1: questions for which answers have been found by SIAC and validated with KDB. Since there is an agreement between two independent components, it is justified to assign a highest reliability score to the group produced by such a combination and to place it at the top ranks. Thirty-five questions were in this group and were ranked from 1 to 35.
- Q2: questions for which answers have been found only by SIAC and not covered by any database. This group, the major one with 438 questions, could be divided in two parts: non NIL answers (389) and NIL answers (49). As described in section 4.3, filters are applied on SIAC output in order to keep only expected entities mapping question class(es) – it may happen that all the candidates are eliminated by this filtering – that is how NIL is produced by the system. It was decided to put these NIL at the end of this group, as they are the results of many treatments and therefore the decision process becomes too uncertain. Inside non NIL answers, order was defined first by decreasing confidences (in question classes) and second by question classes. Order among question classes (see table 6.1) has been derived from previous experiments performed for tuning purpose. For example, our classification component performs well for questions asking for YEAR and DATE, and as named entities mapping these classes are also well detected, we are more confident in answers coming from these series. On the other hand, by the time of our participation, for questions asking for frequencies, named entities finder was not able to detect these expressions – accordingly it should be risky to bet on the class mapping this entity.
- Q3: questions for which the classifier did not assign a class (and tagged "unknown"). This is clearly a flaw in our system's answering process as answer selection depends on these classes. Therefore, such questions will be answered with a NIL and put at the bottom ranks. It happened thirty times over the entire TREC-11's set but three of them were finally over-handled by KDB – and backed up in the first group Q1. The remainders (27) have been left as NIL.

This ordered list (Q1, Q2, Q3) corresponds to the way we ranked the three groups.

YEAR, DATE, COUNTRY, COUNTY, NPP, ACRONYM, CITY, MAIL, MONTH, URL, STATE, ADDRESS, TITLE, LOCATION, ORGAN
Table 6.1: Top 15 questions classes (ordered by preference)

7. Experiments and results

7.1. Official results

Table 7.1 shows the results obtained by our run LIA2002a (only one run was submitted).

Number wrong (W):	440
Number unsupported (U):	4
Number inexact (X):	4
Number right (R):	**52**
Confidence-weighted score (CWS):	**0.246**
Precision of recognizing no answer	7 / 75 = 0.093
Recall of recognizing no answer	7 / 46 = 0.152

Table 7.1 : Official results

7.2. Experiments

After the dead line, we performed some additional experiments. It was possible to evaluate them thanks to the TREC-11 answers patterns made available by Ken Litkowski. For this purpose, a home-made tool was developed to compute the confidence weighted score ("CWS"). In the following, these new experiments will be referred as LIA2002o (o standing for October) and LIA2002n (n for November).

- **Evaluation of the KDB contribution:**

The results reported in table 7.2 are useful to focus only on the behavior of questions for which a KDB was involved. For this, we assume that the other answers (from rank+1 to 500) were wrong:

	#	R	U	CWS
Lia2002a	30	24	4	0.051
Errors at rank	6, 11U, 15, 16U, 21U, and 29			
Lia2002o	33	26	5	0.056
Errors at rank	6, 11U, 16, 17U, 22U, 30, and 33U			
Lia2002n	35	28	5	0.061
Errors at rank	7, 15, 17U, 19U, 24U, 32, and 35U			

Table 7.2 : KDB Contribution

Where:

- "R", "U", "CWS" stand respectively for "right", "unsupported", and "confidence weighted score".
- "Lia2002a" is the run submitted in August for TREC-11,
- "Lia2002o" is a run with few additions in KDB. Also, minor bug corrections inside our whole system and specially in ordering strategy were done (ordering for SIAC answers was broken in our TREC submission)
- "Lia2002n" is our last run. It includes two more entries (a tiny extension of the KDB). The main difference with the previous ones is that answers powered by KDB are ranked by applying the same ordering strategy as answers from SIAC.

- **Answers allocation** (table 7.3): System succeeded in finding a non nil, right and supported answer in about 10% of the cases (column reported as "R-nil"). It provided document containing a correct answer in 15% of the cases (column reported as "D") but failed to extract it in about 5% of the cases (column reported as "D-(R-nil)"). SIAC was able to find 5% of the non nil correct answers but ten of them were overlapped by the KDB.

#	KDB Size	R	U	R-nil	D	D-(R-nil)
Lia2002a	30	53	5	**45**	72	27
Lia2002o	33	55	7	**47**	74	27
Lia2002n	35	55	7	**47**	74	27
Without any KDB	0	34	37	**26**	59	33

Table 7.3 : Answers Allocation

- **Evaluating ordering strategy:** Table 7.4 presents CWS results obtained only by correcting our ordering strategy as we intended it to be. It provided a gain of about 7% just by re-ordering 55 answers.

	R	CWS (1)	CWS (2)	Gain %
Lia2002a	53	**0,246**	**0,268**	+ 9%
Lia2002o	55	**0,258**	**0,278**	+ 8%
Lia2002n	55	**0,266**	**0,285**	+ 7%
Without any KDB	34	**0,084**	**0,139**	+ 65%

Table 7.4 : Ordering Strategy Gain

(1) Answers ordered as for August submission,
(2) Answers ordered by using planned ordering strategy.

8. Conclusion

For our first participation in TREC - question answering, we focused on a small number of questions, that is questions for which an answer can be produced with a sufficient level of confidence. The goal was to reach 30% of accuracy which is honorable as a first trial.

A lot of work remains. Firstly, we could have gone into entity recognition in greater depth, using more statistics. Secondly, because two different tools have been used in order to tag (NE) the documents and the questions, we experienced some problems making a mapping from one to the other. The lack of compatibility should be solved by using the same set of tag. Also, anaphora resolution is too simple and could be applied on many other anaphora

phenomena. Another important point: question tagging is quite weak. For example, for many questions, it assigns the same confidence to different tags. The selection of the tag to be considered could be easily improved. Moreover, answer extraction is too much simple. Because no syntactic tagging is done, it is impossible to choose precisely a phrase in which the answer is supposed to be. So, the only thing we did was to extract the searched entity wherever it was in the candidate sentence. Consequently, many wrong answers were retrieved.

Relying on some knowledge bases clearly improves the results of our system. Typo correction is quite efficient and allows us to answer correctly several questions. We can improve the cases where an answer is provided by the KDB and SIAC fails to retrieve any expected NE, by enriching the question with this answer in order to retrieve supporting documents. Moreover, we could increase the coverage of this KDB in two directions: i.) Find other knowledge sources on more and more subjects, ii.) Use each KDB as a bootstrap in order to enlarge it thru text extraction. We consider that the second item is a key point to make the first one feasible.

Finally, let us consider the graph plotted in figure 3. It represents the growth of correct answers. We can see that the curve grows in stages. Important improvements are followed by long flat lines.

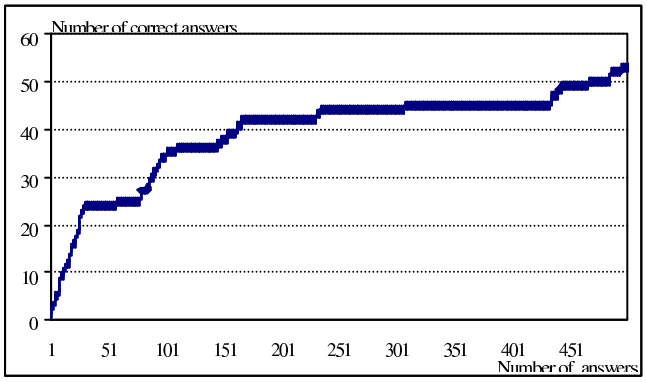

Figure 3: Number of correct answers by number of answers

In fact, there are 5 big stages:

Stage	Correct answers	Accuracy	Length of stage
1-30	24	0.8	30
78-110	11	0.33	33
145-166	6	0.27	22
433-443	4	0.36	11
484-495	3	0.25	12

Table 8.1: Location of correct answers in the list

If we do not consider the first stage (due to the KDB), we have 4 stages (which length is between 11 and 33) where accuracy is quite good (between 0.25 and 0.36). This concentration is sufficiently significant to conclude that some questions have the same behavior and the system performs quite well on these types of questions. Since they are grouped, it should be possible to detect and locate them higher in the list. It could be possible to improve the results by detecting these types of questions.

This concerns 48 questions, that is more than 90% of our correct answers. If they were located at the beginning of the list, the CWS would be 0.32 instead of 0.246.

9. References

[Appelt *et al.*, 1995] D. E. Appelt, J. R. Hobbs, J. Bear, D. Israel, M. Kameyama, A. Kehler, D. Martin, K. Myers, & M. Tyson, "*SRI International FASTUS System MUC-6 Test Results and Analysis*", Proceedings of the Sixth Message Understanding Conference, Columbia, Maryland: Morgan Kaufmann Publishers, pp. 237–248, 1995.

[Béchet *et al.*, 2000] F. Béchet, A. Nasr, F. Genet, "*Tagging Unknown Proper Names Using Decision Trees*", in Proc. of ACL'2000, Hong-Kong, China, pp.77-84, 2000.

[Bellot & El-Bèze, 2000] P. Bellot, M. El-Bèze, "*Clustering by means of decision trees without learning or hierarchical and K-Means like algorithms*", in Proceedings of RIAO'2000, Paris, France, pp. 344-363, 2000.

[Bikel *et al.*, 1999] D. M. Bikel, R. Schwartz and R. M. Weischedel, "*An Algorithm that Learns What's in a Name*", Machine Learning, Special Issue on NLP, 1999.

[Borthwick, 1999] A. Borthwick, "A Maximum Entropy Approach to Named Entity Recognition", Ph.D. (1999) New York University. Department of Computer Science, Courant Institute. Specialized in artificial intelligence and computational linguistics.

[Kuhn & De Mori, 1996] R. Kuhn, Rde Mori, "*The application of semantic decision trees to natural language understanding*", IEEE Transactions on Pattern Analysis and Machine Intelligence, 17(5), pp. 449-460, 1996.

[Lowrance & Wagner, 1975] R. Lowrance & R. A. Wagner, "*An extension of string-to-string correction proble.*", Journal of the ACM, 22(2), 177-183, 1975.

[Schmid, 1994] H. Schmid, "*Probabilistic Part-of-Speech Tagging Using Decision Trees*", in Proc. of the First International Conference on New Methods in Natural Language Processing (NemLap-94), Manchester, England, pp. 44-49, 1994.

[Schmid, 1995] H. Schmid, "*Improvements In Part-of-Speech Tagging With an Application to German*", EACL SIGDAT Workshop, Dublin, Ireland, In: Feldweg and Hinrichs, eds., Lexikon und Text, pp. 47-50, 1995.

[Weischedel, 1995] R. Weischedel, "*BBN: Description of the PLUM System as Used for MUC-6*", Proceedings of the

Sixth Message Understanding Conference, Columbia, Maryland: Morgan Kaufmann Publishers, pp. 55-69, 1995.

10. Appendix

10.1. Hierarchical List of Expected Answer Types

- MISC
  - ACRONYM
  - EXPANDED_ACRONYM
  - ADDRESS
  - ZIP
  - PHONE
  - URL
  - EMAIL
  - AGE
  - DIMENSION
    - ELEVATION
    - WIDTH
    - DIAMETER
    - HEIGHT
    - DEPTH
    - AREA
    - VOLUME
  - PHRASE
    - DEFINITION
    - EXPLANATION
    - EVENT
    - TITLE
      - BOOK
      - FILM
      - MUSIC
      - PAINTING
    - BIOGRAPHY
    - COMMON_WORD
      - CLOTHES
      - VERB
      - TOY
  - PROFESION
  - OTHER_NUMERAL
    - DISTANCE
    - MONEY
    - NUMBER
    - ORDINAL
    - PERCENTAGE
      - CONVERSION_RATE
    - POPULATION
    - QUANTITY
    - SPEED
    - TEMPERATURE
    - WEIGHT
  - TIME
    - DURATION
    - FREQUENCY
    - DATE
      - BIRTHDAY
      - WDAY
      - DAY
      - MONTH
      - YEAR
      - HOUR
- NP
  - ANIMAL
    - BIRD
    - INSECT
  - BODY_PART
  - COLOR
  - CURRENCY
  - DISEASE
  - FIRSTNAME
  - FOOD
  - LANGUAGE
  - MINERAL
  - MUSICAL_INSTRUMENT
  - NICKNAME
  - SPORT
  - VEGETAL
    - FLOWER
    - FRUIT
    - VEGETABLE
  - PROPER_NOUN
    - ACTOR_ACTRESS
    - CHAIRMAN
    - FAMOUS_PERSON
      - NOBEL_PRIZE
    - INVENTOR
    - MUSICIAN
    - PAINTER
    - POLITICIAN
      - PRESIDENT_POLITICIAN
    - SPORTSMAN
    - EXPLORATOR_RESEARCHER
      - EXPLORATOR
      - RESEARCHER
    - WRITER
  - LOCATION
    - CITY
      - CAPITAL
    - CONTINENT
    - COUNTY
    - COUNTRY
    - FAMOUS_PLACE
    - HEMISPHERE
    - LAKE
    - MOUNTAIN
    - PLANET
    - RIVER
    - SEA
    - STARS
    - STATE
    - LOCATION
  - ORGANIZATION
    - COMPANY
      - STORE
    - UNIVERSITY
    - MEDIA
      - JOURNAL
      - TV
      - RADIO
    - TEAM
  - OTHER_NP
  - SEASON
- UNKNOWN
- YES_NO