Using Syntactic Knowledge for QA*

Gosse Bouma, Ismail Fahmi, Jori Mur, Gertjan van Noord, Lonneke van der Plas, and Jörg Tiedemann

Information Science, University of Groningen,
PO Box 716 9700 AS Groningen The Netherlands
g.bouma@rug.nl

Abstract. We describe the system of the University of Groningen for the monolingual Dutch and multilingual English to Dutch QA tasks. First, we give a brief outline of the architecture of our QA-system, which makes heavy use of syntactic information. Next, we describe the modules that were improved or developed especially for the CLEF tasks, among others incorporation of syntactic knowledge in IR, incorporation of lexical equivalences and coreference resolution, and a baseline multilingual (English to Dutch) QA system, which uses a combination of Systran and Wikipedia (for term recognition and translation) for question translation. For non-list questions, 31% (20%) of the highest ranked answers returned by the monolingual (multilingual) system were correct.

1 Introduction

Joost (see figure 1) is a question answering system for Dutch which performs full syntactic analysis of the question and all text in the document collection. Answers are extracted by pattern matching over dependency relations, and potential answers are ranked, among others, by computing the syntactic similarity between the question and the sentence from which the answer is extracted. Apart from the three classical components *question analysis*, *passage retrieval* and *answer extraction*, the system also contains a component (called *qatar*) for extracting answers off-line. All components in our system rely heavily on syntactic analysis, which is provided by Alpino [3], a wide-coverage dependency parser for Dutch.

Question analysis produces a set of dependency relations (i.e. the result of syntactic analysis), it determines the question type, and identifies keywords (for IR). Depending on the question type the next stage is either passage retrieval or table look-up. If the question type matches one of the table categories, this table will be searched for an answer. Tables are created off-line for facts that frequently occur in fixed patterns. We store these facts as potential answers together with the IDs of the paragraphs in which they were found. If table look-up cannot be used, we follow the other path through the QA-system to the

* This research was carried out as part of the research program for *Interactive Multimedia Information Extraction*, IMIX, financed by NWO, the Dutch Organisation for Scientific Research.

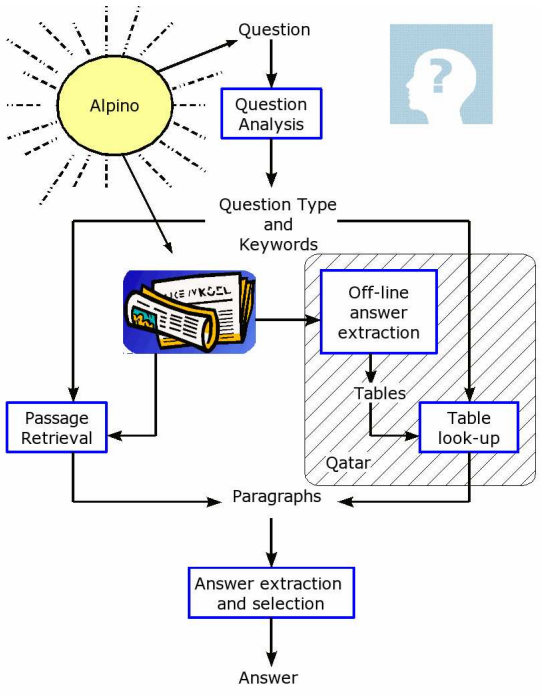

Fig. 1. System architecture of Joost

passage retrieval component. The IR-engine selects relevant paragraphs to be passed on to subsequent modules for extracting the actual answer.

The final processing stage is answer extraction and selection. The input to this component is a set of paragraph IDs, either provided by Qatar or by the IR system. For questions that are answered by means of table look-up, the tables provide an exact answer string. In this case the paragraph context is used only for ranking the answers. For other questions, answer strings have to be extracted from the paragraphs returned by IR. Answer extraction patterns are similar to those used for off-line extraction, but typically more general (and more noisy). Answers are ranked on the basis of frequency of the answer, overlap in named entities between question and answer, syntactic similary between question and answer, and (estimated) reliability of the extraction pattern used. Finally, the highest ranked answers are returned to the user. More detailed descriptions of the system can be found in [1] and [2].

In the rest of the paper, we focus on components of the system that were revised or developed for CLEF 2006, and on discussion of the results.

2 Linguistically Informed Information Retrieval

The information retrieval component identifies relevant paragraphs in the corpus to narrow down the search space for subsequent modules. Answer containing paragraphs that are missed by IR are lost for the entire system. Hence, IR performance in terms of recall is essential. Furthermore, high precision is also desirable as IR scores are used for ranking potential answers.

Given a full syntactic analysis of the CLEF text collection, it becomes feasible to exploit linguistic information as a knowledge source for IR. Using Apache's IR system Lucene [4], we can index the document collection along various linguistic dimensions, such as part of speech tags, named entity classes, and dependency relations. We defined several *layers* of linguistic features and feature combinations and included them as index fields. In our current system we use the following layers: text (stemmed text tokens), root (root forms), RootPos (root forms concatenated with wordclass labels), RootRel (root forms concatenated with the name of the dependency relation to their head words), RootHead (dependent-head bigrams using root forms), RootRelHead (dependent-head bigrams with the type of relation between them), compound (compounds identified by Alpino), ne (named entities), neLOC, nePER, neORG (only location, person, or organisation names, respectively), and neTypes (labels of named entities in the paragraph). The layers are filled with appropriate data extracted from the analysed corpus.

Each of the index fields defined above can be accessed using Lucene's query language. Complex queries combining keywords for several layers can be constructed. Queries to be used in our system are constructed from the syntactically analysed question. We extract linguistic features in the same way as done for building the index. The task now is to use this rich information appropriately. The selection of keywords is not straightforward. Keywords that are too specific might harm the retrieval performance. It is important to carefully select features and feature combinations to actually improve the results compared to standard plain text retrieval.

For the selection and weighting of keywords we applied a genetic algorithm trained on previously collected question answer pairs. For constructing a query we defined further keyword restrictions to make an even more fine-grained selection. For example, we can select RootHead keywords from the question which have been tagged as nouns. Each of these (possibly restricted) keyword selections can be weighted with a numeric value according to their importance for retrieval. They can also be marked as "required" using the '+' character in Lucene's query syntax. All keyword selections are then concatenated in a disjunctive way to form the final query. Figure 2 gives an example.

Details of the genetic optimisation process are given in [5]. As the result of the optimisation we obtain an improvement of about 19% over the baseline using standard plain text retrieval (i.e. the text layer only) on unseen evaluation data. It should be noted that this improvement is not solely an effect of using root forms or named entity labels, but that many of the features that are assigned a high weight by the genetic algorithm refer to layers that make use of dependency information.

```
text:(stelde Verenigde Naties +embargo +Irak)
ne:(Verenigde_Naties^2 Verenigde^2 Naties^2 Irak^2)
RootHead:(Irak/tegen embargo/stel_in)
neTypes:(YEAR)
```

Fig. 2. An example IR query from the question *Wanneer stelde de Verenigde Naties een embargo in tegen Irak ?* (When did the United Nations declare the embargo against Iraq?) using plain text tokens, named entities with boost factor 2, RootHead bigrams for nouns, and the named entity class for the question type.

3 Coreference Resolution for Off-line Question Answering

Off-line answer extraction has proven to be very effective and precise. The main problem with this technique is the lack of coverage. One way to increase the coverage is to apply coreference resolution. For instance, the age of a person may be extracted from snippets such as:

(1) a. de 26-jarige Steffi Graf (*the 26-year old Steffi Graf*)
b. Steffi Graf....de 26-jarige tennisster (*Steffi Graf...the 26-year old tennis player*)
c. Steffi Graf....Ze is 26 jaar. (*Steffi Graf...She is 26 years old*)

If no coreference resolution is applied, only patterns in which a named entity is present, such as (1-a) will match. Using coreference resolution, we can also extract the age of a person from snippets such as (1-b) and (1-c), where the named entity is present in a preceding sentence.

We selected 12 answer types that we expect to benefit from coreference resolution: `age`, `date/location of birth`, `age/date/location/cause of death`, `capital`, `inhabitants`, `founder`, `function`, and `winner`. Applying the basic patterns to extract facts for these categories we extracted 64,627 fact types. We adjusted the basic patterns by replacing the slot for the named entity with a slot for a pronoun or a definite NP.

Our strategy for resolving definite NPs is based on knowledge about the categories of named entities, so-called instances (or categorised named entities). Examples are *Van Gogh* IS-A *painter*, *Seles* IS-A *tennis player*. We acquired instances by scanning the corpus for apposition relations and predicate complement relations We scan the left context of the definite NP for named entities from right to left. For each named entity we encounter, we check whether it occurs together with the definite NP as a pair on the instance list. If so, the named entity is selected as the antecedent of the NP. As long as no suitable named entity is found we select the next named entity and so on until we reach the beginning of the document. If no named entity is found that forms an instance pair with the definite NP, we select simply the first preceding named entity.

We applied a similar technique for resolving pronouns. Again we scan the left context of the anaphor (now a pronoun) for named entities from right to left. We implemented a preference for proper nouns in the subject position. For

each named entity we encounter, we check whether it has the correct NE-tag and number. If we are looking for a person name, we do another check to see if the gender is correct.[1] After having resolved the anaphor, the corresponding fact containing the antecedent named entity was added to the appropriate table.

We estimated the number of additional fact types we found using the estimated precision scores (on 200 manually evaluated facts). Coreference resolution extracted approximately 39,208 (60.7%) additional facts. 5.6% of these involve pronouns, and 55.2% definite NPs. The number of facts we extracted by the pronoun patterns is quite low. We did a corpus investigation on a subset of the corpus which consisted of sentences containing terms relevant to the 12 selected question types[2]. In only 10% of the sentences one or more pronouns appeared. This outcome indicates that the possibilities of increasing coverage by pronoun resolution are inherently limited.

4 Lexical Equivalences

One of the features that is used to rank potential answers to a question is the amount of syntactic similarity between the question and the sentence from which the answer is taken. Syntactic similarity is computed as the proportion of dependency relations from the question which have a match in the dependency relations of the answer sentence. In [6], we showed that taking syntactic equivalences into account (such as the fact that a *by*-phrase in a passive is equivalent to the subject in the active, etc.) makes the syntactic similarity score more effective.

In the current system, we also take lexical equivalences into account. That is, given two dependency relations ⟨`Head, Rel, Dependent`⟩ and ⟨`Head′, Rel, Dependent′`⟩, we assume that they are equivalent if both `Head` and `Head′` and `Dependent` and `Dependent′` are near-synonyms. Two roots are considered near synonyms if they are identical, synomyms, or spelling variants, or if one is an abbreviation, genitive form, adjectival form, or compound suffix of the other.

A list of synonyms (containing 118K root forms in total) was constructed by merging information from EuroWordNet, dictionary websites, and various encyclopedias (which often provide alternative terms for a given lemma keyword). The spelling of person and geographical names entities tends to be subject to a fair amount of variation and the spelling used in a question is not necessarily the same as the one used in a parapgraph which provides the answer. Using edit distance to detect spelling variation tends to be very noisy. To improve the precision of this method, we restricted ourselves to person names, and imposed the additional constraint that the two names must occur with the same function in our database of functions (used for off-line question answering). Thus, *Felipe Gonzalez* and *Felippe Gonzales* are considered to be variants only if they are known to have the same function (e.g. prime-minister of Spain). Currently, we

[1] We created a list of boy's names and girl's names by downloading such lists from the Internet. To be accepted as the correct antecedent, the proper name should not occur on the name list of the opposite sex of the pronoun.

[2] terms such as "geboren" (*born*), "stierf" (*died*), "hoofdstad" (*capital*) etc.

recognize 4500 pairs of spelling variants. The compound rule applies when one of the words is a compound suffix of the other. It also cover multi word terms analyzed as a single word by the parser (i.e. *colitis ulcerosa*).

We tested the effect of incorporating lexical equivalences on questions from previous CLEF tasks. Although approximately 8% of the questions receives a different answer when lexical equivalences are incorporated, the effect on the overall score is negligible. We suspect that this is due to the fact that in the definition of synonyms, no distinction is made between various senses of a word, and the equivalences defined for compounds tend to introduce a fair amount of noise (e.g. the *Calypso-queen* of the Netherlands is not the same as the *queen* of the Netherlands). It should also be noted that most lexical equivalences are not taken into consideration by the IR-component. This probably means that some relevant documents (especially those containing spelling variants of proper names) are missed.

5 Definition Questions

Definition questions can ask either for a definition of a named entity (*What is Lusa?*) or a concept (*What is a cincinatto*). We used appositions (*the Portugese press agency Lusa*), nominal modifiers (*milk sugar (saccharum lactis)*), *or* (*ofwel*) disjunctions (*milk sugar or saccharum lactis*), predicative complements (*milk sugar is (called/known as) saccharum lactis*), and predicative modifiers (*composers such as Joonas Kookonen*) to find potential answers. As some of these patterns tend to be very noisy, we also check whether there exists an ISA-relation between the head noun of the definition, and the term to be defined. ISA-relations are collected from named entity – noun appositions (48K) and head noun – concept pairs (136K) extracted from definition sentences in an automatically parsed version of Dutch Wikipedia. Definition sentences were identified automatically (see [7]). Answers for which a corresponding ISA-relation exists in Wikipedia are given a higher score.

For the 40 definition questions in the Dutch QA test set, 18 received a correct first answer (45%), which is considerably better than the overall performance on non-list questions (31%). We consider 7 of the 40 definition questions to be concept definition questions. Of those, only 1 was answered correct.

6 Temporally Restricted Questions

Sometimes, questions contain an explicit date:

(2) a. Which Russian Tsar died in 1584?
b. Who was the chancellor of Germany from 1974 to 1982?

To provide the correct answer to such questions, it must be ensured that there is no conflict between the date mentioned in the question and temporal information present in the text from which the answer was extracted.

If a sentence contains an explicit date expression, this is used as *answer date*. A sentence is considered to contain an explicit date if it contains a temporal expression referring to a date (*2nd of August, 1991*) or a relative date (*last year*). The denotation of the latter type of expression is computed relative to the date of the newspaper article from which the sentence is taken. Sentences which do not contain an explicit date are assigned an *answer date* which corresponds to the date of the newspaper from which the sentence is extracted.

For questions which contain an explicit date, this date is used as the *question date*. For all other questions, the *question date* is nil. The *date score* of a potential answer is 0 if the *question date* is nil, 1 if answer and question date match, and -1 otherwise.

There are 31 questions in the Dutch QA test set which contain an explicit date, and which we consider to be temporally restricted questions. Our monolingual QA system returned 11 correct first answers for these questions (10 of correctly answered questions ask explicitly for a fact from 1994 or 1995). The performance of the system on temporally restricted questions is similar to the performance achieved for (non-list) questions in general (31%).

7 Multilingual QA

We have developed a baseline English to Dutch QA-sytem which is based on two freely avaiable resources: Systran and Wikipedia. For development, we used the CLEF 2004 multieight corpus. [8]

The English source questions are converted into an HTML file, which is translated automatically into Dutch by Systran.[3] These translations are used as input for the monolingual QA-system described above.[4]

This scenario has a number of obvious drawbacks: (1) translations often result in grammatically incorrect sentences, (2) even if a translation can be analyzed syntactically, it may contain words or phrases that were not anticipated by the question analysis module, and (3) named entities and (multiword) terms are not recognized. We did not spend any time on fixing the first and second potential problem. While testing the system, it seemed that the parser was relatively robust against grammatical irregularities. We did notice that question analysis could be improved, so as to take into account peculiarities of the translated questions.

The third problem seemed most serious to us. It seems Systran fails to recognize many named entities and multiword terms. The result is that these are translated on a word by word basis, which typically leads to errors that are almost certainly fatal for any component (starting with IR) which takes the

[3] Actually, we used the Babelfish interface to Systran, `http://babelfish.altavista.digital.com/`

[4] For English to Dutch, the only alternative on-line translation service seems to be Freetranslation (`www.freetranslation.com`). When testing the system on questions from the multieight corpus, the results from Systran seemed slightly better, so we decided to use Systran only.

translated string as starting point. To improve on the treatment of named entities and terms, we extracted from English Wikipedia all pairs of lemma titles and their cross-links to the corresponding link in Dutch Wikipedia. Terms in the English input which are found in the Wikipedia list are escaped from automatic translation and replaced by their Dutch counterparts directly. This potentially avoids three types of errors: the term should not be translated, but it is by Systran (*Jan Tinbergen* → *Januari Tinbergen*), (2) the term is not translated by Systran, but it should (*Pippi Longstocking*), (3) the term should be translated, but it is translated wrongly by Systran (*Pacific Ocean* → *Vreedzame Oceaan*).

Of the 200 input questions, 48 contained terms that matched an entry in the bilingual term database extracted from Wikipedia. 4 of the marked terms are incorrect (*Martin Luther* instead of *Martin Luther King* is marked as a term, *nuclear power* instead of *nuclear power plants* is marked as a term, *prime-minister* is translated as *minister-voorzitter* rather than as *minister-president* or *premier*, and *the game* is incorrectly recognized as a term (it matches the name of a movie in Wikipedia) and not translated).

Although the precision of recognizing terms is high, it should be noted that recall could be much better. Terms such as *Olympic Winter Games*, *World Heritage Sites*, and proper names such as *Jack Soden* and *Chad Rowan* are not recognized, leading to word by word translations (*Olympische Spelen van de Winter, De Plaatsen van de Erfenis van de Wereld*) that sometimes are highly cryptical (*Hefboom Soden, de Lijsterbes van Tsjaad*). In addition, many unrecognized proper names show up as discontinuous strings in the translation (i.e. *What did Yogi Bear steal* is translated as *Wat Yogi stal de Beer*).

Although the performance of the multilingual system is a good deal less than that of the monolingual system, there actually are a few questions which are answered correctly by the bilingual system, but not by the monolingual system. This is due to the fact that the (more or less) automatic word by word translations in these cases match more easily with the answer sentences than the manually constructed Dutch sentences (which paraphrase the English sentence).

8 Evaluation and Error Analysis

The results from the CLEF evaluation are given in figure 3.

The monolingual system assigned only 13 questions a question type for which a table with potential answers was extracted off-line. For only 5 of those, an answer is found off-line. This suggests that the effect of off-line techniques on the overall result is relatively small. As off-line answer extraction tends to be more accurate than IR-based answer extraction, it may also explain why the results for the CLEF 2006 task are relatively modest.[6]

[6] For development, we used almost 800 questions from previous CLEF tasks. For those questions, almost 30% of the questions are answered by answers that were found off-line. 75% of the first answers for those questions is correct. Overall, the development system finds almost 60% correct first answers.

Q type	#	# correct	% correct	MRR
Factoid Questions	146	40	27.4	
Definition Questions	40	18	45	
Temporally Restricted[5]	1	0	0	
Non-list questions	187	58	31	0.346
List Questions	13	15/65 answers correct (P@5 = 0.23)		

Q type	#	# correct	% correct	MRR
Factoid Questions	147	27	18.4	
Definition Questions	39	11	28.2	
Temporally Restricted	1	0	0	
Non-list questions	187	38	20.3	0.223
List Questions	13	4/37 answers correct (P@5 = 0.06)		

Fig. 3. Official CLEF scores for the monolingual Dutch task (top) and bilingual English to Dutch task (bottom).

If we look at the scores per question type for the most frequent question types (as they were assigned by the question analysis component) , we see that definition questions are answered relatively well (18 out of 40 of the first answers correct), that the scores for general WH-questions and location questions are in line with the overall score (16 out of 52 and 8 out of 25 correct), but that measure and date questions are answered poorly (3 out of 20 and 3 out of 15 correct). On the development-set (of 800 questions from previous CLEF tasks), all of these question types perform considerably better (the worst scoring question type are measure questions, which still finds a correct first answer in 44% of the cases).

A few questions are not answered correctly because the question type was unexpected. This is true in particular for the (3) questions of the type *When did Gottlob Frege live?*.

Somewhat suprisingly, question analysis also appears to have been an important source of errors. We estimate that 23 questions were assigned a question type that was either wrong or dubious. Dubious assignments arise when question analysis assigns a general question type (i.e. *person*) where a more specific question type was available (i.e. *founder*).

Attachment errors of the parser are the source of small number of mistakes. For instance, Joost replies that O.J. Simpson was accused of *murder on his ex-wife*, where this should have been *murder on his ex-wife and a friend*. As the conjunction is misparsed, the system fails to find this constituent. Different attachments also cause problems for the question *Who was the German chancellor between 1974 and 1982?*. It has an almost verbatim answer in the corpus (*the social-democrat Helmut Schmidt, chancellor between 1974 and 1982*), but since the temporal restriction is attached to the verb in the question, and the noun *social-democrat* in the answer, this answer is not found.

The performance loss between the bilingual and the monolingual system is approximately 33%. This is somewhat more than the differences between multilingual and monolingual QA reported for many other systems (see [9] for an overview). However, we do believe that it demonstrates that the syntactic analysis module is relatively robust against the grammatical anomalies present in automatically translated input. It should be noted, however, that 19 out of 200 questions cannot be assigned a question type, whereas this is the case for only 4 questions in the monolingual system. Adapting the question analysis module to typical output produced by automatic translation, and improvement of the term recognition module (by incorporating a named entity recognizer and/or more term lists) seems relatively straightforward, and might lead to somewhat better results.

References

1. Bouma, G., Mur, J., van Noord, G., van der Plas, L., Tiedemann, J.: Question answering for Dutch using dependency relations. In: Working Notes for the CLEF 2005 Workshop, Vienna (2005)
2. Bouma, G., Fahmi, I., Mur, J., van Noord, G., van der Plas, L., Tiedeman, J.: Linguistic knowledge and question answering. Traitement Automatique des Langues (2006) to appear.
3. Bouma, G., van Noord, G., Malouf, R.: Alpino: Wide-coverage computational analysis of Dutch. In: Computational Linguistics in The Netherlands 2000. Rodopi, Amsterdam (2001)
4. Jakarta, A.: Apache Lucene - a high-performance, full-featured text search engine library. http://lucene.apache.org/java/docs/index.html (2004)
5. Tiedemann, J.: Improving passage retrieval in question answering using NLP. In: Proceedings of the 12th Portuguese Conference on Artificial Intelligence (EPIA), Covilhã, Portugal, LNAI Series, Springer (2005)
6. Bouma, G., Mur, J., van Noord, G.: Reasoning over dependency relations for QA. In Benamara, F., Saint-Dizier, P., eds.: Proceedings of the IJCAI workshop on Knowledge and Reasoning for Answering Questions (KRAQ), Edinburgh (2005) 15–21
7. Fahmi, I., Bouma, G.: Learning to identify definitions using syntactic features. In Basili, R., Moschitti, A., eds.: Proceedings of the EACL workshop on Learning Structured Information in Natural Language Applications, Trento, Italy (2006)
8. Magnini, B., Vallin, A., Ayache, C., Erbach, G., Peas, A., de Rijke, M., Rocha, P., Simov, K., Sutcliffe, R.: Overview of the clef 2004 multilingual question answering track. In Peters, C., Clough, P.D., Jones, G.J.F., Gonzalo, J., Kluck, M., Magnini, B., eds.: Multilingual Information Access for Text, Speech and Images: Results of the Fifth CLEF Evaluation Campaign. Lecture Notes in Computer Science Vol. 3491. Springer Verlag (2005)
9. Ligozat, A.L., Grau, B., Robba, I., Vilat, A.: Evaluation and improvement of cross-lingual question answering strategies. In Peñas, A., Sutcliffe, R., eds.: EACL workshop on Multilingual Question Answering, Trento, Italy (2006)