The University of Groningen at QA@CLEF 2006 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
g.bouma@rug.nl

Abstract

We describe our system 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 esepcially for the CLEF tasks, i.e. (1) incorporation of syntactic knowledge in the IR-engine, (2) incorporation of lexical equivalences, (3) incorporation of coreference resolution for off-line answer extraction, (4) treatment of temporally restricted questions, (5) treatment of definition questions, and (6) 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% of the highest ranked answers returned by the monolingual system were correct and 20% of the answers returned by the multilingual system.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: H.3.1 Content Analysis and Indexing; H.3.3 Information Search and Retrieval; J.5 [**Arts and Humanities**]: Language translation; Linguistics

General Terms

Algorithms, Measurement, Performance, Experimentation

Keywords

Question answering, Dutch, Lexical Equivalences, Coreference Resolution

1 Introduction

Joost is a question answering system for Dutch which is characterized full syntactic analysis of the question and of all text returned by the IR engine for a given query. 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. A brief overview of the system architecture is given in section 2. More detailed descriptions of the system can be found in Bouma et al. (2005) and Bouma et al. (2006). In the rest of the paper, we focus on components of the system that were revised or developed for

*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.

CLEF 2006, and on discussion of the results. Section 3 discusses the IR system, which tries to use various linguistic features to improve precision. In section 4, we discuss the effect of incorporating coreference resolution into the module which extracts answers to frequently asked question-types off-line. Section 5 contains an overview of techniques we implemented to identify (near) synonyms, spelling variants, etc. Sections 6 and 7 present our treatment of definition and temporally restricted questions. A description of our baseline multilingual QA system (based on Systran and Wikipedia) is given in section 8. The results of the evaluation are presented in section 9.

2 Architecture

We briefly describe the general architecture of our QA system Joost. The architecture of our system is depicted in figure 1. Apart from the three classical components *question analysis*, *passage retrieval* and *answer extraction*, the system also contains a component called *Qatar*, which is based on the technique of extracting answers off-line. All components in our system rely heavily on syntactic analysis, which is provided by Alpino (Bouma, van Noord, and Malouf, 2001), a wide-coverage dependency parser for Dutch. Alpino is used to parse questions as well as the full document collection from which answers need to be extracted. A brief overview of the components of our QA system follows below.

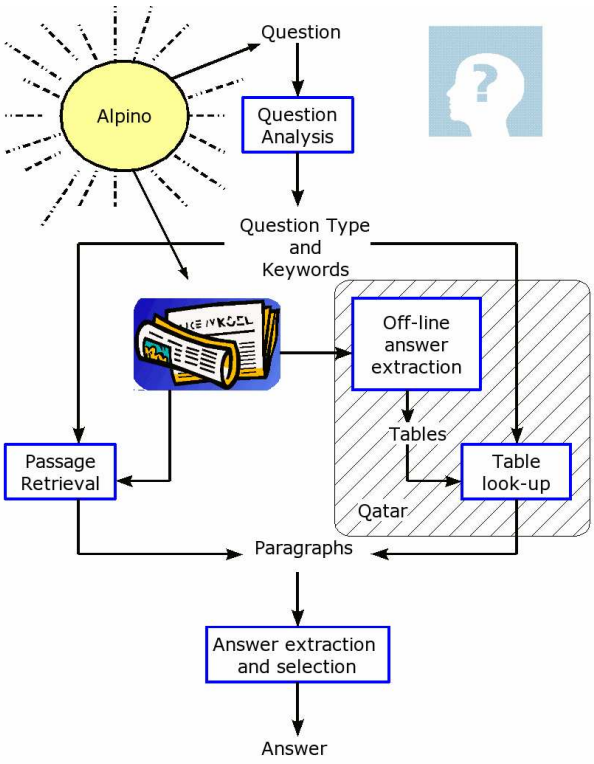

Figure 1: System architecture of Joost.

The first processing stage is question analysis. The input to this component is a natural language question in Dutch, which is parsed by Alpino. The goal of question analysis is to determine the question type and to identify keywords in the question.

Depending on the question type the next stage is either passage retrieval or table look-up (using Qatar). If the question type matches one of the table categories, it will be answered by Qatar. 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. During the question answering process the question type determines which table is selected (if any).

For all questions that cannot be answered by Qatar, we follow the other path through the QA-system to the passage retrieval component. Previous experiments have shown that a segmentation of the corpus into paragraphs is most efficient for information retrieval (IR) performance in QA. Hence, IR passes relevant paragraphs to subsequent modules for extracting the actual answers from these text passages.

The final processing stage in our QA-system 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. We then retrieve all sentences from the text collection included in these paragraphs. For questions that are answered by means of table look-up, the tables provide an exact answer string. In this case the context is used only for ranking the answers. For other questions, answer strings have to be extracted from the paragraphs returned by IR. The features that are used to rank the extracted answers will be explained in detail below. Finally, the answer ranked first is returned to the user.

3 Linguistically Informed Information Retrieval

The information retrieval component in our system is used to identify relevant paragraphs from the CLEF corpus to narrow down the search for subsequent answer extraction modules. Accurate IR is crucial for the success of this approach. Answer containing paragraphs that have been 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 (Jakarta, 2004), 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 extracted from syntactically analysed sentences and included them as index fields. In our current system we use 12 layers containing the following features: text (stemmed plain text tokens), root (linguistic 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 (compositional compounds identified by Alpino), ne (named entities), neLOC (location names), nePER (person names), neORG (organisation names), and neTypes (labels of named entities identified 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. We can select keywords based on their wordclass, their relation to the head word and based on a combination of the two. 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. Look at the example query in figure 2 to get an impression of possible queries in the system.

Note that the question type provided by the question analysis module is used to query the neTypes layer with a corresponding named entity label.

The optimsation procedure using the genetic algorithm works essentially as follows: First we

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

Figure 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 the following keyword selections: (1) all plain text tokens (except stop words), (2) Named entities weighted with boost factor 2, (3) RootHead bigrams for all words tagged as noun, (4) the question type transformed into a named entity class, (5) plain text keywords of words in an object relation (embargo & Irak).

start with initial settings using only one type of keyword selection. These settings are applied to construct queries from our given collection of questions. The queries are then used to retrieve a fixed number of paragraphs for each question and the retrieval performance is measured in terms of mean reciprocal rank scores. We used the answer string provided in the training data to determine if a paragraph is relevant or not. After the initial step two preferable settings (according to the scores) are selected and their settings are combined to test new parameters. Additionally we apply simple mutation operations to alter parameters at random from time to time. The process of selecting and combining is then repeated until no significant improvement can be measured anymore. Details of the genetic optimisation process are given in (Tiedemann, 2005). 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.

4 Coreference Resolution for Off-line Question Answering

The system component *Qatar* extracts potential answers from the corpus off-line using dependency based patterns. Off-line answer extraction has proven to be very effective. The results typically show a high precision score. However, the main problem with this technique is the lack of coverage of the extracted answers. 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. They are shown in table 1. Applying the basic patterns to extract facts for these categories we extracted

Answer Type	Answer Type	Answer Type	Answer Type
Age	Age of Death	Cause of Death	Founder
Date of Birth	Date of Death	Capital	Function
Location of Birth	Location of Death	Inhabitants	Winner

Table 1: Answer types for which coreference resolution was applied

64,627 fact types. We adjusted the basic patterns by replacing the slot for the named entity with a slot for a pronoun. Similarly, we adjusted the patterns to match sentences with a definite noun. We considered noun phrases preceded by a definite determiner as definite noun phrases.

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[1].

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. The pronouns we tried to resolve were the nominative forms of the singular pronouns *hij* (he), *zij/ze* (she), *het* (it) and the plural pronoun *zij/ze* (they). We chose to resolve only the nominative case, as in almost all patterns the slot for the name was the slot in subject position. The number of both the anaphor and the antecedent was determined by the number of the main verb. Since we find the anaphors by matching patterns, we knew what the named entity (NE) tag of the antecedent should be.

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 so and if it concerns a non-person NE-tag, the named entity is selected as the antecedent. If we are looking for a person name, we have to do another check to see if the gender is correct. To determine the gender of the selected name we created a list of boy's names and girl's names by downloading such lists from the Internet[2]. The female list contained 12,691 names and the male list 11,854 names. To be accepted as the correct antecedent, the proper name should not occur on the name list of the opposite sex of the pronoun. After having resolved the anaphor, the fact was added to the appropriate table.

For both extraction modules we randomly selected a sample of around 200 extracted facts and we manually evaluated these facts on the following two criteria: (1) correctness of the fact and (2) in the case of coreference resolution, correctness of the selected antecedent.

We estimated the number of additional fact types we found using the estimated precision scores. If we had only used the pronoun patterns we would have found 3,627 (5.6%) new facts. On the other hand, if we had only used the definite noun patterns we would have found 35,687 (55.2%) new facts. Using both we extracted 39,208 (60.7%) additional facts.

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[3]. 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.

5 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 Bouma, Mur, and van Noord (2005), we showed that taking syntactic equivalences into account (such as the fact that a *by*-phrase in a

[1] We limited our search to the predicate complement relation between named entities and a noun and excluded examples with negation

[2] http://www.namen.info, http://www.voornamenboek.nl, http://www.babynames.com and http://prenoms.free.fr

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

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 R and R′ are considered near synonyms in the following cases:

- R = R′,
- R and R′ are synonyms,
- R and R′ are spelling variants,
- R is an abbreviation of R′, or vice versa,
- R is the genitive form of R′, or vice versa,
- R is the adjectival form of the country name R′, or vice versa,
- R matches with a part of the compound R′, or vice versa

A list of synonyms (containing 118K root forms in total) was constructed by merging information from EuroWordNet, the dictionary website `mijnwoordenboek.nl`, 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. For instance, the 1994 Spanish prime minister is referred to as either *Felipe Gonzalez, Felippe Gonzales, Felipe Gonzales* or *Felipe González*. The spelling used in a question is not necessarily the same as the one used in a parapgraph which provides the answer:

(2) a. Hoe heet de dochter van **Deng Xiaopeng** (*What is the name of the daughter of Deng Xiaopeng?*)

(2) Deng Rong, de dochter van de Chinese leider **Deng Xiaoping** (*Deng Rong, the daughter of the Chinese leader Deng Xiaoping*).

One might consider two named entities spelling variants if the edit distance between the two is less than a certain threshold, or if one is a word suffix of the other (i.e. *Maradona* and *Diego Armando Maradona*). However, this method tends to be very noisy. To improve the precision of the 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 recognize 4500 pairs of spelling variants.

The compound rule applies when one of the words contains a hyphen (*Fiat-topman*) or a space (i.e. Latin phrases like *colitis ulcerosa* are analyzed as a single word by our parser) and the other word matches with either part of it, or when the lexical analyzer of the parser analyzes a word as a compound (i.e. *chromosoomafwijking* (*chromosome deficit*)), and the other word matches with the suffix (*afwijking*).

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.

6 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 the following answer patterns to find potential answers:

- 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*)
- Predicative modifiers (*composers such as Joonas Kookonen*)

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:

- All Named Entity – Noun appositions (48K) extracted from an automatically parsed version of the Dutch Wikipedia
- All head noun – concept pairs (136K) extracted from definition sentences found in Dutch Wikipedia .

Definition sentences were identified automatically (see Fahmi and Bouma (2006)). Answers for which a corresponding ISA-relation exists in Wikipedia are given a higher score.

For the 40 definition questions in the 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. Thus, answering concept definitions correctly remains a challenge.

7 Temporally Restricted Questions

Sometimes, questions contain an explicit date:

(3) 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.

To answer temporally restricted questions, we try to assign a date to sentences containing a potential answer to the question. 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 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,
- -1 otherwise.

There are 31 questions in the CLEF 2006 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%).

8 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. (Magnini et al., 2005)

The English source questions are converted into an HTML file, which is translated automatically into Dutch by Systran.[4] These translations are used as input for the monolingual QA-system described above.[5]

This scenario has a number of obvious drawbacks:

- Translations often result in grammatically incorrect sentences, for which no (correct) grammatical analysis can be given.
- Even if a translation can be analyzed syntactically, it may contain words or phrases that were not anticipated by the question analysis module.
- 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 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. The following examples compare the effect of direct translation (b-examples) and translation combined with Wikipedia look-up (c-examples).

(4) a. Who is Jan Tinbergen
b. Wie is Januari Tinbergen?
c. Wie is Jan Tinbergen?

(5) a. In which country do people sleep with their feet on the pillow, according to Pippi Longstocking?
b. In welk land slapen de mensen met hun voeten op het hoofdkussen, volgens Pippi Longstocking?
c. In welk land slapen de mensen met hun voeten op het hoofdkussen, volgens Pippi Langkous?

(6) a. How large is the Pacific Ocean?
b. Hoe groot is de Vreedzame Oceaan?
c. Hoe groot is Grote Oceaan?

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

[5] 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.

Three cases can arise: the term should not be translated, but it is by Systran (*Jan 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*)

48 of the 200 input questions 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.

(7) a. What are the three elementary particles of physics according to the Standard Model?
b. Wat zijn de drie elementaire deeltjes van fysica volgens Standaardmodel? (translated)
c. Wat zijn de drie fundamentele deeltjes in het Standaardmodel uit de deeltjesfysica? (monolingual)

(8) a. Who is the author of the book Jurassic Park?
b. Wie is de auteur van het boek Jurassic Park ? (translated)
c. Wie schreef het boek Jurassic Park ? (monolingual)

In (7), the translated sentence uses *elementaire deeltjes*, which also occurs in the answer sentence. The monolingual question, however, uses the equivalent phrase *fundamentele deeltjes*, but this equivalence is not detected by the QA system. In (8) the translated question uses the noun *auteur*, which also occurs in the sentence providing the answer, whereas the monolingual version uses the verb *schrijven* (*to write*).

9 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.[7]

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).

[7]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 system finds well-over 50% correct first answers.

Q type	#	# correct	% correct	MRR
Factoid Questions	146	40	27.4	
Definition Questions	40	18	45	
Temporally Restricted[6]	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)		

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

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?*.

Attachment errors of the parser are the source of some 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 Ligozat et al. (2006) 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

Bouma, Gosse, Ismail Fahmi, Jori Mur, Gertjan van Noord, Lonneke van der Plas, and Jörg Tiedeman. 2006. Linguistic knowledge and question answering. *Traitement Automatique des Langues*. to appear.

Bouma, Gosse, Jori Mur, and Gertjan van Noord. 2005. Reasoning over dependency relations for QA. In Farah Benamara and Patrick Saint-Dizier, editors, *Proceedings of the IJCAI workshop on Knowledge and Reasoning for Answering Questions (KRAQ)*, pages 15–21, Edinburgh.

Bouma, Gosse, Jori Mur, Gertjan van Noord, Lonneke van der Plas, and Jörg Tiedemann. 2005. Question answering for Dutch using dependency relations. In *Working Notes for the CLEF 2005 Workshop*, Vienna.

Bouma, Gosse, Gertjan van Noord, and Robert Malouf. 2001. Alpino: Wide-coverage computational analysis of Dutch. In *Computational Linguistics in The Netherlands 2000*. Rodopi, Amsterdam.

Fahmi, Ismail and Gosse Bouma. 2006. Learning to identify definitions using syntactic features. In Roberto Basili and Alessandro Moschitti, editors, *Proceedings of the EACL workshop on Learning Structured Information in Natural Language Applications*, Trento, Italy.

Jakarta, Apache. 2004. Apache Lucene - a high-performance, full-featured text search engine library. http://lucene.apache.org/java/docs/index.html.

Ligozat, Anne-Laure, Brigitte Grau, Isabella Robba, and Anne Vilat. 2006. Evaluation and improvement of cross-lingual question answering strategies. In Anselmo Peñas and Richard Sutcliffe, editors, *EACL workshop on Multilingual Question Answering*. Trento, Italy.

Magnini, B., A. Vallin, C. Ayache, G. Erbach, A. Peas, M. de Rijke, P. Rocha, K Simov, and R. Sutcliffe. 2005. Overview of the clef 2004 multilingual question answering track. In C. Peters, P. D. Clough, G. J. F. Jones, J. Gonzalo, M. Kluck, and B. Magnini, editors, *Multilingual Information Access for Text, Speech and Images: Results of the Fifth CLEF Evaluation Campaign*, Lecture Notes in Computer Science Vol. 3491. Springer Verlag.

Tiedemann, Jörg. 2005. Improving passage retrieval in question answering using NLP. In *Proceedings of the 12th Portuguese Conference on Artificial Intelligence (EPIA)*, Covilhã, Portugal. LNAI Series, Springer.