Question Answering with Joost at CLEF 2007[*]

Gosse Bouma, Geert Kloosterman, 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 present a brief overview of our QA-system, which makes heavy use of syntactic information. Next, we describe the modules that were developed especially for CLEF 2007, i.e. preprocessing of Wikipedia, inclusion of query expansion in IR, anaphora resolution in follow-up questions, and a question classification module for the multilingual task. We achieved 25.5% accuracy for the Dutch monolingual task, and 13.5% accuracy for the multilingual task.

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, Wikipedia, Information Retrieval, Query Expansion, Anaphora Resolution

1 Introduction

The Question Answering task for CLEF 2007 contained two innovations. First, the document collection was extended with Wikipedia, the online encyclopedia that is available for many different languages. As described in section 3, we preprocessed the XML source files for this document collection so that we could index it adequately for the purposes of Information Retrieval. In addition, we extracted all relevant plain text, and parsed it.

Second, the test questions were grouped in topics. Within a topic, questions might refer to or presuppose information from previous questions or answers to these questions. We developed a simple anaphora resolution system (described in section 5) that detects anaphoric elements in a question, and tries to find a suitable antecedent in the first question of a topic, or in the answer to that question.

[*] 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.

In addition to these innovations, we also improved the Information Retrieval component of our QA system. In section 4, we show that query expansion based on (automatically acquired) synonym-lists and blind relevance feedback improves the mean reciprocal rank of the IR module.

In section 6 we describe a question classification module for the multilingual QA system, which uses both the question class assigned to the English source question and the class assigned to the automatically translated Dutch target question. This leads to a modest improvement.

The results of our system are discussed in section 7, and some suggestions for future work are given in section 8.

2 Joost: A QA system for Dutch

Joost (Bouma et al., 2005) is a question answering system for Dutch which is characterized by the fact that it relies on syntactic analysis of the question as well as the documents in which answers need to be found. The complete document collection is parsed by Alpino (Bouma, van Noord, and Malouf, 2001), a wide-coverage dependency parser for Dutch. The resulting depedency trees are stored as XML. Answers are extracted by pattern matching over syntactic 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.

The architecture of our system is depicted in figure 1. Apart from the standard components *question analysis*, *passage retrieval*, *answer extraction* and *answer ranking*, the system also contains a component called *Qatar*, which collects all answers to questions of a specific type (i.e. birthdates) off-line. Answers to questions for which a Qatar-table exists are found by means of table look-up.

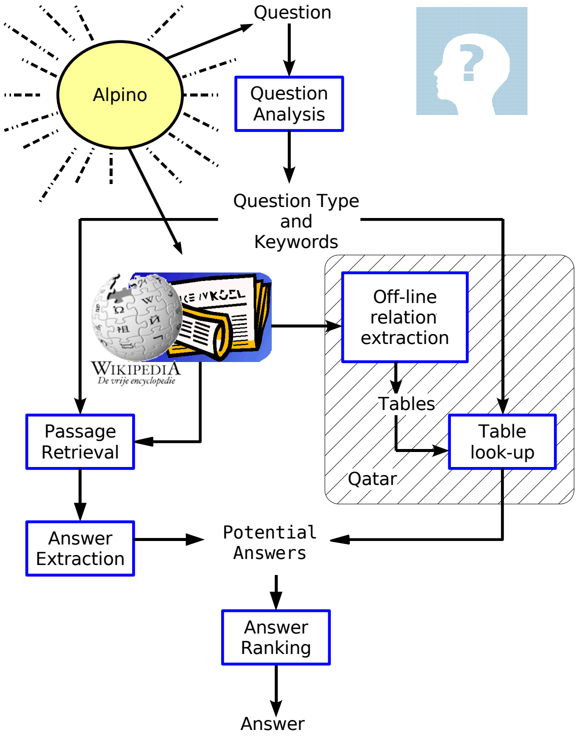

Figure 1: System architecture of Joost.

The first processing stage is question analysis. The input to the question analysis 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. Qatar consists of a number of manually written syntactic patterns for extraction of interesting relations (i.e. *creator-object* tuples such as *Heinrich Mann - Der Untertan*). Recall is improved by using a set of equivalence rules for syntactic dependency patterns (Bouma, Mur, and van Noord, 2005), and by using anaphora resolution (Mur, 2006). Using these patterns, the parsed corpus is searched exhaustively, and all extracted relation tuples are stored in tables. We defined patterns for 20 relations. Using these patterns, almost 400K relation instances were extracted from the Dutch Wikipedia.

For all questions that cannot be answered by Qatar, we follow the other path through the QA-system to the passage retrieval component. Instead of retrieving full documents, the IR module retrieves passages (see section 3). The 40 most relevant passages retrieved by IR are passed on to the answer extraction module. Here, we use patterns similar to those used by Qatar, but now slightly more general, to find the actual answer strings. Per sentence, at most one potential answer string is selected.

The final step is answer selection. Given a list of potential answers from Qatar or the IR-based QA module, the most promising answer is selected. Answers are ranked using various features, such as syntactic overlap between question and answer sentence, word overlap, proper name overlap, the reliability of the pattern used to extract the answer, and the frequency of the answer. The answer ranked first is returned to the user.

3 Preprocessing Wikipedia

New in this year's CLEF QA tracks was the inclusion of Wikipedia in the corpus. The Wikipedia corpus is different from the newspaper texts that were used so far in a number of ways. First of all, whereas the newspaper collection is relatively redundant (there are two newspapers covering the same period in the Dutch collection, and news stories tend to contain a fair amount of repetition), this is far less the case for the encyclopedia, which contains many facts that are mentioned only in one article. Thus, we expect redundancy-based techniques (typically using patterns that are noisy but provide high recall in combination with frequency-based ranking of results) to less effective for Wikipedia. Second, Wikipedia consists of structured web-documents, containing many lists, tables, and cross-references. In the newspaper collection, only article titles, and paragraphs are provided. By mining the structure of Wikipedia documents, it is possible to extract a large number of facts that cannot be found using syntactic patterns. Due to time constraints, we applied only the syntactic patterns that were developed for the newspaper collection.

An XML-version of the Dutch Wikipedia was provided by the University of Amsterdam.[1] For IR and parsing, we were interested in obtaining just the text in each of the articles. We developed a series of stylesheets which removes material that was irrelevant for our task (i.e. navigation and pictures), and which returns the remaining content as highly simplified XML, containing only information that is required to identify the segmentation of the text into titles, sections, and lists. The segmentation is used in the IR index. From the simplified XML, plain text can be extracted easily. The result is tokenized and split into 4.7 million sentences. The sentences were parsed with the Alpino-parser.

The Qatar relation extraction module searches the corpus exhaustively for interesting facts, and stores these facts in a database. For Wikipedia, we used the patterns as they were developed for the newspaper corpus, with only minor modifications. In particular, we did not try to extract facts from lists, or using the XML structure.

Our IR system retrieves passages rather than complete articles. For previous CLEF tasks, we used the existing paragraph markup in the newspaper data to split documents into passages to be retrieved. For Wikipedia, similar markup exists but often refers to very small units, in many cases only single sentences. The Dutch Wikipedia corpus contains about 4.7 million sentences split into

[1] http://ilps.science.uva.nl/WikiXML/

about 2 million units. Single sentence units usually correspond to headers and subsection headers which often contain important keywords that match well with given queries. Unfortunately, including these as separate passages results in a strong preference for these units when retrieving passages. To avoid this we implemented a simple solution that merges consecutive units until they are bigger than a pre-defined size of 200 characters.

Despite its simplicity this approach works sufficiently well and made it possible to easily integrate the new Wikipedia corpus into our IR index. The same approach has also been applied to the newspaper corpus in order to create and index with a uniform segmentation.

4 Passage Retrieval with and without query expansion

In CLEF 2007 we submitted two runs of our system, applying two different settings of the information retrieval (IR) component which is used to retrieve relevant passages for a given question. The main difference between these two settings is the inclusion of query expansion techniques in one of them.

Common to both settings is the approach, previously described in Tiedemann (2005), in which linguistic features have been integrated in the IR index. IR queries are constructed from questions using various features and feature combinations. Furthermore, keyword selection constraints are introduced using part-of-speech tags (*POS*) and dependency relation types (*rel*). For each keyword type a separate weight is used to optimize retrieval performance. Furthermore, we also use proximity queries requiring terms within a given text window. Keyword weights and window sizes have been trained on questions and answers from the CLEF QA tracks in 2003 and 2005 using a genetic algorithm. The mean reciprocal rank of relevant passages retrieved has been improved from 0.52 (using standard plain text keywords) to 0.62 (including linguistic features and optimized settings) for questions from the training set and from 0.49 to 0.57 on unseen evaluation data (CLEF 2004 questions). Details of the optimization procedure are discussed in Tiedemann (2005).

The second run includes various forms of query expansion. The main purpose of expanding the query is to increase recall of the passage retrieval component in order to minimize the risk of missing relevant information. We experimented with two general techniques for query expansion: global methods using fixed lists and local techniques using blind relevance feedback. For the latter we applied an implementation of the Rocchio algorithm for Lucene, LucQE (Rubens, 2007; Rubens, 2006), which we adapted to our purposes. Relevance feedback is known to be most useful for increasing recall. In blind relevance feedback (also called pseudo-relevance feedback) user interaction is simulated by simply selecting the highest ranked documents as the positive examples and ignoring negative ones. Rocchio is used to re-weight existing keywords and also to add new terms from the positive examples. We restricted this type of re-weighting and keyword expansion for the plain text field only and a maximum of 10 new keywords. The top five documents were used as positive examples and the Rocchio parameters where set to common values used in the literature ($\alpha = 1$ and $\beta = 0.75$). Furthermore, we used a fixed decay value of 0.1 for decreasing the importance of documents selected for feedback.

Furthermore, we used global expansion techniques using several lists of expansion terms. Firstly, we used redirects from the Dutch Wikipedia. Redirects link search terms similar to existing Wikipedia lemmas to corresponding articles. Redirects mainly cover spelling variations but also include various synonyms. Secondly, we used synonyms of nouns, verbs, and adjectives automatically extracted from word-aligned parallel corpora (van der Plas and Tiedemann, 2006). For this, we aligned the Europarl corpus (Koehn, 2003) with its 11 languages and used aligned translations of Dutch words as features in a distributional similarity approach. Using this technique we obtained 13,896 near-synonyms for 6,968 Dutch nouns, 3,207 near-synonyms for 1,203 verbs and 3,556 near-synonyms for 1,621 adjectives. More details about the algorithm used for the extraction can be found in van der Plas and Tiedemann (2006). Thirdly, we included ISA-relations of named entities extracted from syntactically annotated monolingual corpora (van der Plas and Bouma, 2005). The parsed document collection contains over 2 million instances of an apposition relation between a noun and a named entity (i.e. the *composer Aaron Copland*), where the noun provides an ISA-label for the named entity. After filtering infrequent combinations (often caused by parsing errors), we are left with almost 400K unique tuples.

5 Anaphora Resolution for Follow-Up Questions

A new feature in the 2007 QA task are follow-up questions. Questions are grouped in topics, consisting of a number of questions. Answering non-initial questions may require information from previous questions or answers to previous questions. The TREC QA task has included follow-up questions for a number of years. As no development data was available for the CLEF task, we used English examples from previous TREC QA tasks for inspiration.[2] Note however that in TREC descriptive *topics* are explicitly provided, whereas in CLEF only an numeric topic id is given.

The most important aspect of follow-up questions is anaphora resolution, i.e. the process of detecting anaphoric phrases that depend on a previous antecedent expression for their interpretation, and assigning a correct antecedent to them.

A noun phrase was considered to be anaphoric if it was a personal (1-b) or impersonal (2-b) pronoun, a possessive pronoun (1-c), a deictic pronoun (3-b), an NP introduced by a deictic determiner (4-b), or an NP introduced by a definite determiner and not containing any modifiers (5-b).

(1) a. When was Napoleon born?
    b. Which title was introduced by *him*?
    c. Who were *his* parents?

(2) a. What is the KNMI?
    b. When was *it* founded?

(3) a. What is an ecological footprint?
    b. When was *this* introduced?

(4) a. Who lead the Russian Empire during the Russion-Turkish War of 1787-1792?
    b. Who won *this war*?

(5) a. Since when is Cuba ruled by Fidel Castro?
    b. When was the flag of *the country* designed?

Antecedents were restricted to named entities from the first question/answer pair of a topic. The answer was chosen as antecedent if the initial question was one of a limited number of question types which ask for a named entity (i.e. *what is the capital of, who wrote/founded/.. , who is the chair/president/.. of*). In other cases, the first named entity from the question was chosen. We adopted this naive approach mostly because we lacked data to test and evaluate more sophisticated approaches. Note also that quite a few TREC systems limit anaphora resolution to resolving anaphoric expressions to the topic of the question (see Hickl et al. (2006) for a notable exception), apparently with reasonable success.

Our anaphora resolution system operates on the result of the syntactic dependency parse of the sentence. If anaphora resolution applies, and an antecedent is found, the set of dependency relations for the question is extended with dependency relations for the antecedent. That is, given an `Anaphor` resolved to `Antecedent`, for each dependency relation ⟨`Head, Rel, Anaphor`⟩ in the question, we add a dependency relation ⟨`Head, Rel, Antecedent`⟩. Note that, as the IR system described in the previous section constructs queries on the basis of the dependency parse of the question, this ensures that the `Antecedent` is also included in the IR query.

According to our inspection of the best monolingual run, there were 56 questions which required anaphora resolution. For 29 questions (52%), a correct antecedent for an anaphoric expression was found. In 15 cases (27%), a wrong antecedent was given. An important source of errors were cases where the answer to the initial question was correctly chosen as antecedent, but the answer

[2] i.e. `trec.nist.gov/data/qa/2006_qadata/QA2006_testset.xml`

Target	FQs	Target	FQs
EN	133	RO	78
NL	122	FR	76
DE	84	PT	50
IT	84	ES	30

Table 1: Number of follow-up questions (FQs) per target language. All tasks consist of 200 questions.

was wrong. Incorrect antecedents also occurred when the intended antecedent was not (analysed by the parser as) a named entity. In cases such as (3-b) above, the antecedent is a common noun (*ecological footprint*). In cases such as ((4-b)) the antecedent (*Russian-Turkish War*) is analysed by the parser as an NP headed by a noun (*war*), and thus not recognized as a named entity. There were only a few cases where the antecedent was not in the first question/answer pair of a topic but in a later question.

12 cases (21%) were missed altogether by the anaphora module. These are due to the fact that no attempt was made to treat temporal anaphora such as *toen, destijds* (*during that moment/period*), and *daarvoor* (*before this date*), to treat locative uses of *er* (*there*), and a number of highly implicit anaphoric relations (i.e. given a question about the theme park *de Efteling*, the question *which attraction opened in 1993?* should be interpreted as asking about an attraction in *de Efteling*). A few antecedents were missed because we resolved at most one anaphoric element per question, whereas some questions actually contain two anaphoric elements (i.e. *Who was* he *in the eighties version of* the cartoon?).

Finally, there were 4 cases where anaphora resolution was triggered by an element that was not anaphoric (*false alarms*). These were all caused by the fact that relative pronouns were misclassified as deictic pronouns by the resolution component.

An interesting feature of CLEF is the fact that similar tasks are being executed for different languages. Follow-up questions were included in all tasks, but table 1 shows that the number of such questions varies considerably per target language. This suggests that the number of anaphoric expressions is also likely to vary considerably between tasks.

6 Question Classification in Multilingual QA

Our system for multilingual QA performs English to Dutch QA, i.e. questions are in English, and answers are to be found in the Dutch document collection. English questions are translated to Dutch using Babelfish/Systran. As explained in Bouma et al. (2006), one problem with this approach is the fact that proper names and concepts are often mistranslated (i.e. they are translated whereas they should remain unchanged, or a special translation exists in Dutch, or a complex name is not recognized as a syntactic unit, and is split up in the translated string). As the presence of names and concepts directly influences the performance of the QA system, we tried to reduce the number of errors using Wikipedia. For each name or concept in the English question, we check if there is a Wikipedia lemma with that title. If so, we check if a link to a corresponding Dutch page exists. If this is the case, the title of the Dutch lemma is used in the translation. Otherwise, the English name is used in the translation.

This year, we improved the system by using newer (and much expanded) versions of Wikipedia, inclusion of redirect pages, and the online geographical database *geonames*[3] for translation of geographical locations. Inspection of the translation results suggests that the coverage of these resources is quite good, although some problems remain. The use of redirects, for instance, causes *Google* to be mapped to the less frequent term *Google Inc.* Also, abbreviations tend to be replaced

[3] www.geonames.org

		Joost		union	
Testset	Qs	MRR	1st	MRR	1st
2003	377	0.329	0.292	0.347	0.310
2004	200	0.406	0.350	0.429	0.375
2006	200	0.225	0.195	0.213	0.185

Table 2: Mean Reciprocal Rank (for the first 5 answers) and CLEF-score (1st answer correct) for English to Dutch QA, using Joost and a combination of Joost and Quest (union).

by their expanded meanings. Although both IR and the linguistic QA modules recognize many abbreviations and expanded terms as synonyms, this may still cause problems. An obvious case is a question asking for the meaning of an abbreviation. Another problem is the fact that many common words, which are not concepts occur as lemmas in Wikipedia. If no corresponding Dutch page exists, this causes some terms to show up untranslated in the Dutch question (i.e. for the adjectives *French* and *Eastern*, an English Wikipedia page exists, but no Dutch counterpart).

A second important aspect of QA-systems is question classification. As many automatically generated translations are grammatically poor, parsing may lead to unexpected results, and, as a consequence, question classification is often incorrect or impossible. To remedy this problem, we also included a question classifier for English, which we ran on the English source questions. We manually constructed a mapping from the question types used for English to the question types used in Joost. We expected that such a mapping might give more accurate results than classification of the automatically translated questions. Both the (mapped) English question type and the Joost type assigned to the translated are used to find an answer to the question. Note that question classification of the source language question is used in many MLQA systems (see Ligozat et al. (2006) for an overview), but usually the classification used for the source question is the same as that used by the answer extraction components.

There are various question classifiers for English which use the question classes of Li and Roth (2002). They propose a classification consisting of 6 coarse question types and 50 fine-grained types. Each question is assigned a label consisting of both a coarse and a fine question type. We used the automatically trained classifier described by Hacioglu and Ward (2003)[4], which uses the Li and Roth classification.

Joost uses over 40 question types, some of which correspond quite well to those of Li and Roth. Mismatches are problematic especially in those cases where Joost expects a more fine-grained class than the class produced by Li and Roth. For instance, Li and Roth classify *what is the capital of Togo* as `loc:city` whereas Joost has the class `capital`. Furthermore, the question classes assigned by Joost are not just labels, but typically consist of a label combined with one or more phrases from the question that are crucial for answering the question. I.e. the question *what does NASA stand for?* is assigned the type `abbr:exp` by Quest, whereas it is assigned the label `abbreviation(NASA)` by Joost. The mapping therefore tries to fill in missing arguments (usually names) on the basis of the syntactic parse of the Dutch translated question.

In many cases, the question class assigned by Joost is more helpful than the class assigned after mapping the English question class. An important exception, however, are questions that were assigned no class by Joost itself (usually, because the translated question contained syntactic errors that made the parser fail). In those cases, using a mapped question class is preferable over using no class at all. We tested our approach on data from previous years. The effect of including question classification based on the original question turned out to be small, however, as can be seen in figure 2. For the 2006 dataset, the effect was even negative.

[4] Available until recently at `sds.colorado.edu/QUEST`

Run	Accuracy (%)	Right	ineXact	Unsupported	Wrong
Dutch-mono	24.5	49	11	4	136
Dutch-mono + QE	25.5	51	10	4	135
En-Du	13.0	26	8	7	159
En-Du + QE	13.5	27	7	5	161

Table 3: Official CLEF scores for the monolingual Dutch task and the bilingual English to Dutch task (200 questions), with and without Query Expansion (QE) .

Q type	# q's	Accuracy (%)	Right	ineXact	Unsupported	Wrong
Factoids	156	25.6	40	5	4	107
List	16	6.3	1	0	5	10
Definition	28	35.7	10	0	0	18
Temp. Restricted	41	19.5	8	3	3	27
NIL	20	0.0	0	0	0	20

Table 4: Results per question type for the best Dutch monolingual run.

7 Evaluation

The results from the CLEF evaluation are given in table 3. Table 4 gives results per question type for the best Dutch monolingual run. For 20 questions no answer was given (i.e. NIL was returned by the system as answer).[5] There are two main reasons for this: mistakes in anaphora resolution, which made it impossible to find documents or answers matching the question and lack of coverage of the question analysis component. Although there were 28 definition questions, only 18 were classified as such by Joost. List questions were an important source of errors.

The impact of adding Wikipedia to the document collection was significant. Although the text version of the Dutch Wikipedia is smaller than the newspaper text collection (approximately 50M and 80M words respectively), 150 of the 180 questions (i.e. over 80%) that received an answer were answered using Wikipedia.

Definition questions are answered using a relation-table that was created of-line. In addition to these, 24 questions were assigned a question type for which a relation-table existed. This number is lower than for previous CLEF tasks.

The system normally checks that answers suggested by the system do not occur in the question. It turned out that, in the context of follow-up questions, this filter needs to take into account anaphora resolution as well. That is, if a question contained an anaphor that was resolved to antecedent A, in some cases the system would still suggest A as an answer to the question, and such answers were not filtered (as A did not occur in the question string).

8 Conclusions and Future Work

The inclusion of Wikipedia in the CLEF QA-task has made the task more realistic and attractive. We believe that performance on this task can be improved by taking the structure of Wikipedia more seriously, and by developing methods for relation and answer extraction that combine NLP with XML-based extraction.

Follow-up questions required the incorporation of a anaphora resolution component for questions. The current version of this module performs reasonably well, but its coverage should be extended (to cover locative anaphors and multiple anaphors). The proper treatment of lexical

[5] At the moment of writing, it is not clear to us whether there actually were questions for which NIL was the correct answer.

knowledge within the system remains an issue that requires more attention. The performance of the IR-module was improved using automatically acquired synonyms, but this knowledge has not been integrated yet in the relation and answer extraction modules.

References

Bouma, Gosse, Ismail Fahmi, Jori Mur, Gertjan van Noord, Lonneke van der Plas, and Jörg Tiedeman. 2005. Linguistic knowledge and question answering. *Traitement Automatique des Langues*, 2(46):15–39.

Bouma, Gosse, Ismail Fahmi, Jori Mur, Gertjan van Noord, Lonneke van der Plas, and Jörg Tiedemann. 2006. The University of Groningen at QA@CLEF 2006: Using syntactic knowledge for QA. In *Working Notes for the CLEF 2006 Workshop*, Alicante.

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

Hacioglu, Kadri and Wayne Ward. 2003. Question classification with support vector machines and error correcting codes. In *Proceedings of HLT-NACCL 2003*, pages 28–30, Edmonton, Alberta, Canada.

Hickl, Andrew, John Williams, Jeremy Bensley, Kirk Roberts, Ying Shi, and Bryan Rink. 2006. Question answering with LCC's Chaucer at TREC 2006. In E. M. Voorhees and Lori P. Buckland, editors, *TREC 2006 Proceedings*, Gaithersburg, Maryland.

Koehn, Philipp. 2003. Europarl: A multilingual corpus for evaluation of machine translation. unpublished draft, available from `http://people.csail.mit.edu/koehn/publications/europarl/`.

Li, Xin and Dan Roth. 2002. Learning question classifiers. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING)*, pages 556–562.

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.

Mur, Jori. 2006. Increasing the coverage of answer extraction by applying anaphora resolution. In *Fifth Slovenian and First International Language Technologies Conference (IS-LTC '06)*.

Rubens, Neil. 2006. The application of fuzzy logic to the construction of the ranking function of information retrieval systems. *Computer Modelling and New Technologies*, 10(1):20–27.

Rubens, Neil. 2007. Lucqe - lucene query expansion. http://lucene-qe.sourceforge.net/.

Tiedemann, Jörg. 2005. Improving passage retrieval in question answering using NLP. In C. Bento, A. Cardoso, and G. Dias, editors, *Progress in Artificial Intelligence – Selected papers from the 12th Portuguese Conference on Artificial Intelligence (EPIA)*, volume 3808 of *LNAI Series*. Springer; Berlin, Covilhã, Portugal, pages 634 – 646.

van der Plas, Lonneke and Gosse Bouma. 2005. Automatic acquisition of lexico-semantic knowledge for question answering. In *Proceedings of Ontolex 2005 – Ontologies and Lexical Resources*, Jeju Island, South Korea.

van der Plas, Lonneke and Jörg Tiedemann. 2006. Finding synonyms using automatic word alignment and measures of distributional similarity. In *Proceedings of ACL/Coling*.