Cross-lingual Question Answering with QED

**Kisuh Ahn, Beatrice Alex, Johan Bos, Tiphaine Dalmas,
Jochen L. Leidner, and Matthew B. Smillie**

School of Informatics, University of Edinburgh
email: trec-qa@inf.ed.ac.uk

Abstract

We present improvements and modifications of the QED open-domain question answering system developed for TREC-2003 to make it cross-lingual for participation in the Cross-Linguistic Evaluation Forum (CLEF) Question Answering Track 2004 for the source languages French and German and the target language English. We use rule-based question translation extended with surface pattern-oriented pre- and post-processing rules for question reformulation to create and English query from its French or German original. Our system uses deep processing for the question and answers, which requires efficient and radical prior search space pruning. For answering factoid questions, we report an accuracy of 16% (German to English) and 20% (French to English), respectively.

1 Introduction

This report describes QED, a question answering (Q&A) system developed at the University of Edinburgh, and its performance at CLEF-2004. QED [LBD+04] was originally developed for monolingual (English) Q&A tasks, so we needed to extend it with a machine translation (MT) component in order to be able to participate in the CLEF evaluation exercise. We concentrated on the languages French and German for the cross-language QA task and used the 200 French and German questions from CLEF-2003 [MRV+03] as developement data. As we aimed at English as target language, we only required an MT component to translate the questions.

The CLEF evaluation exercise for QA is based on that of TREC [Voo04]. The task is to give possibly exact answers for factoid and definition questions, and back these up with a document that supports the answer. Questions for which no answer can be found in the document collection have to be answered with the string "NIL". Each answer needs to be associated with a confidence value (a number between 0 and 1), in order to reward systems that are able to evaluate their own performance.

In the remainder of this paper we describe the general architecture of the cross-lingual QED question answering system as well as its individual components (Section 2). Most of the QED system is similar to that described in [LBD+04], minus the more elaborate question-typing, the use of Lemur instead of MG for Information Retrieval (IR), and several minor enhancements in the various components. We present our results obtained in the CLEF-2004 evaluation in Section 3, and conclude in Section 4.

2 The QED System

2.1 Architecture

The translation component was added as a front-end to the existing English QED open domain question answering (QA) system. We chose this system architecture in order to exploit the already available end-to-end QA system which was developed for TREC-2003 [LBD+04].

The questions are translated using our MT module, tokenized, and optionally reformulated. After stemming, POS-tagging and parsing, the question is parsed. A semantic representation is generated from the grammatical relations, which is used to construct a query. The query is posed to the document retrieval module to obtain documents. A passage segmenting and ranking tool is used to prune the search space and find document regions likely to contain answers. Its output is parsed and a semantic representation for answer candidates is created likewise. An answer extraction module attempts to match and score representations of question and answer candidates. Finally, evidence from the Web in the form of co-occurrence counts is used to check answer candidates for validity and the best answer is output.

2.2 Machine Translation

Our translation component consists of Babelfish[1], an online machine translation (MT) engine based Systran. This is a rule-based MT engine, which makes use of both bilingual dictionaries and linguistic rules designed empirically for specific language pairs. Perhaps unsurprisingly, we initially observed several errors specific to language pairs that occurred regularly for various types of questions. Using Babelfish, we translated 200 CLEF-2003 questions DE→EN automatically and let a linguist judge the results for acceptability. Only 29% were found to be acceptable by the human subject. Many of the errors were caused by foreign words and literally translated Named Entities.

We decided to develop automatic pre- and post-processing rules to improve the quality of the MT output. As the English MT output serves as input into the QA system, our aim was to produce MT output as correct as possible. We therefore invested some time in examining the types of errors that occurred in the Systran output for both language pairs, and devised sets of pre- and post-correction rules.[2]

Pre-correction After an extensive analysis of the MT output of the development data, we identified such instances and designed pre-processing rules to reformulate certain questions into simpler constructions. For example, we reformulated French questions starting with "*À quel moment*" into "*Quand*" (when) questions.

Post-correction Similarly, we devised a set of post-processing rules to correct regular errors in the MT output. For example, in the case of French questions that are distinguished by the inversion of subject pronoun and verb such as "*Où X travaille-t-il?*", the English MT output is "*Where X does it work?*" instead of "*Where does X work?*". German questions such as "*Wie heißt X?*" are literally translated into "*How is X called?*" rather than "*What is X called?*". The surface pattern-oriented pre- and post-processing rules enabled us to correct such errors automatically and thus considerably improve the MT output.

These pre- and post-processing rules improved the MT component considerably, and although the results were far from perfect, we expected them to be good enough for our purposes.

2.3 Document Retrieval, Passage Extraction and Ranking

We used the Lemur toolkit[3] to realize document retrieval using the Vector-Space Model. The question was analyzed syntactically and semantically and a weighted set of phrases were constructed from the Discourse Representation Structures, which were converted into structured queries for Lemur. The most relevant 300 documents were retrieved for subsequent processing.

Our passage segmentation and ranking component qtile takes a query and a set of retrieved documents and extracts n-sentence passages (called "tiles"), and assigns a score to them. This is done by sliding an n-sentence window over the document stream at a time in a sentence-wise fashion, retaining all window tiles that contain *at least one* of the words in the query and also always must contain *all* upper-case query words. The score is based on heuristics like

[1] `http://babelfish.altavista.com/`

[2] We used the GNU recode utility to convert the CLEF test questions from UTF-8 character encoding into ISO 8859-1 (Latin-1) encoding required by Babelfi sh.

[3] `http://www-2.cs.cmu.edu/~lemur/`

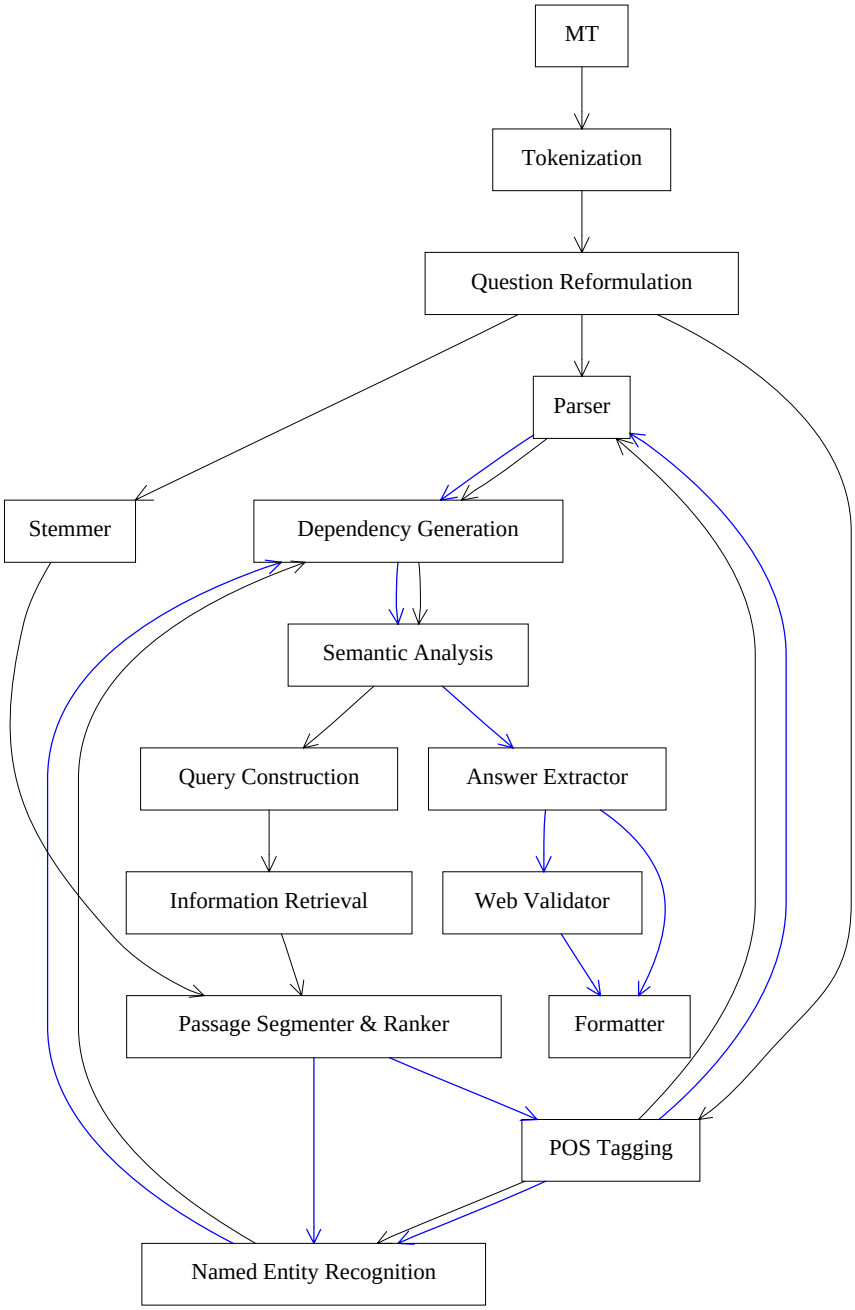

Figure 1: The QED system architecture for CLEF-2004 (dataflow graph). Normal arrows represent processing of the question, bold arrows represents processing of answers.

- number of non-stopword query word tokens (as opposed to types) found in the tile;
- a comparison of the capitalization of query occurrence and tile occurrence of a term;
- the occurrence of 2-grams and 3-grams in both question and tile.

Each tile's score s is multiplied with a slightly asymmetric triangular window function w to weights sentences in the centre of a window higher than in the periphery and to break ties (W is the number of word tokens):

$$w(s) = \begin{cases} 1.1 \times \frac{s}{|W|} & s \leq |W| \\ 1.0 \times -\frac{s}{|W|} & \text{otherwise} \end{cases}$$

The qtile component has linear asymptotic time complexity and requires constant space. For CLEF-2004 we use a window size of 3 sentences and output the top-scoring 100 tiles (duplicates are eliminated) for further processing.

2.4 Question Typing

We used a taxonomy of eleven basic question types (Figure 2), based on the strategies used for finding suitable answers within the large variety of question patterns. This division is based on answers in the form of sentences (S), adjectives (ADJ), and noun phrases (NP). Some of the question-types are further divided into subtypes, where C is a concept, R a relation, and U a unit of measurement.

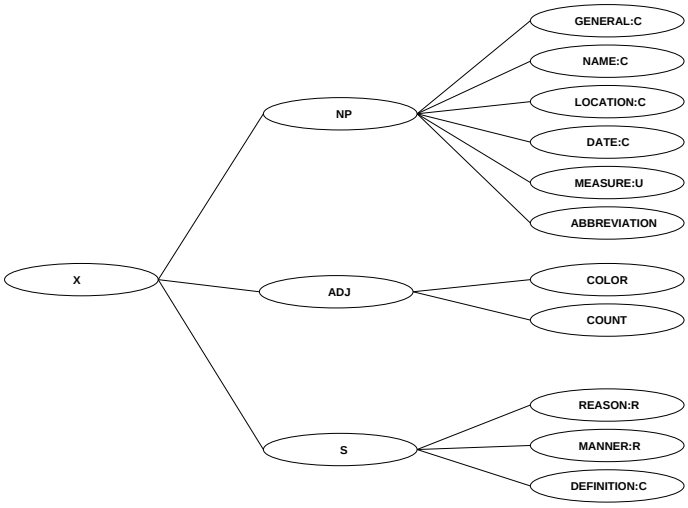

Figure 2: The Question Type Taxonomy used in QED

The question types are determined after the semantic analysis of the question using a rule-based system. For instance, "*How how is the sun?*" gets assigned the question type MEASURE:TEMPERATURE, and "*Who is Janis Joplin?*", the question type DEFINITION:PERSON. The question types are used by the answer selection component to constrain the set of potential answers.

2.5 Linguistic Analysis

The C&C maximum entropy POS tagger [CC03a] is used to tag the question words and the text segments returned by the tiler. The C&C NE-tagger [CC03b] is also applied to the question and text segments, identifying named entities from the standard MUC-7 data set (locations, organisations, persons, dates, times and monetary amounts). The POS tags and NE-tags are used to construct a semantic representation from the output of the parser.

We used the RADISP system [BC02] to parse the question and the text segments returned by the tiler. The RADISP parser returns syntactic dependencies represented by grammatical relations such as `ncsubj` (non-clausal subject), `dobj` (direct object), `ncmod` (non-clausal modifier), and so on. The set of dependencies for a sentence are annotated with POS and NE information and converted into a graph in Prolog format.

To increase the quality of the parser output, we reformulated imperatives in "list questions" (e.g. *Name countries in Europe*) into proper question form (*What are countries in Europe?*). The RADISP parser was much better at returning the correct dependencies for such questions, largely because the RADISP POS tagger typically assigned the incorrect tag to *Name* in the imperative form. We applied a similar approach to other question types not handled well by the parser.

The output of the parser, a set of dependency relations (describing a graph) between syntactic categories, is used to build a semantic representation—both for the question under consideration and for the text passages that might contain an answer to the question. Categories contain the following information: the surface word-form, the lemmatized word-form, the word position in the sentence, the sentence position in the text, named-entity information, and a POS tag defining the category.

2.6 Semantic Interpretation

Our semantic formalism is based on Discourse Representation Theory [KR93], but we use an enriched form of Discourse Representation Structure (DRS), combining semantic information with syntactic and sortal information. DRSs are constructed from the dependency relations in a recursive way, starting with an empty DRS at the top node of the dependency graph, and adding semantic information to the DRS as we follow the dependency relations in the graph, using the POS information to decide on the nature of the semantic contribution of a category.

Following DRT, DRSs are defined as ordered pairs of a set of discourse referents and a set of DRS-conditions. The following types of basic DRS-conditions are considered: `pred(x,S)`, `named(x,S)`, `card(x,S)`, `event(e,S)`, and `argN(e,x)`, `rel(x,y,S)`, `mod(x,S)`, where `e`, `x`, `y` are discourse referents, `S` a constant, and `N` a number between 1 and 3. Questions introduce a special DRS-condition of the form `answer(x,T)` for a question type `T`. We call this the *answer literal*; answer literals play an important role in answer selection.

Implemented in Prolog, we reached a recall of around 80%. (By *recall* we mean the percentage of categories that contributed to semantic information in the DRS). Note that each passage or question is translated into one single DRS; hence DRSs can span several sentences. Some basic techniques for pronoun resolution are implemented as well. However, to avoid complicating the answer extraction task too much, we only considered non-recursive DRSs in our TREC-2003 implementation, i.e. DRSs without complex conditions introducing nested DRSs for dealing with negation, disjunction, or universal quantification.

Finally, a set of DRS normalisation rules are applied in a post-processing step, thereby dealing with active-passive alternations, question typing, inferred semantic information, and the disambiguating of noun-noun compounds. The resulting DRS is enriched with information about the original surface word-forms and POS tags, by co-indexing the words, POS tags, the discourse referents, and DRS-conditions.

2.7 Answer Selection

The answer extraction component takes as input a DRS for the question, and a set of DRSs for selected passages. The task of this component is to extract answer candidates from the passages. This is realised by performing a match between the question-DRS and a passage-DRS, by using a relaxed unification method and a scoring mechanism indicating how well the DRSs match each other.

Taking advantage of Prolog unification, we use Prolog variables for all discourse referents in the question-DRSs, and Prolog atoms in passage-DRSs. We then attempt to unify all terms of the question DRSs with terms in a passage-DRS, using an A^* search algorithm. Each potential answer is associated with a score, which we call the DRS-score. High scores are obtained for perfect matches (i.e., standard unification) between terms of the question and passage, low scores for less perfect matches (i.e., obtained by "relaxed" unification). Less perfect matches are granted for different semantic types, predicates with different argument order, or terms with symbols that are semantically familiar according to WordNet [Fel98].

After a successful match the answer literal is identified with a particular discourse referent in the passage-DRS. Recall that the DRS-conditions and discourse referents are co-indexed with the surface word-forms of the source passage text. This information is used to generate an answer string, simply by collecting the words that belong to DRS-conditions with discourse referents denoting the answer. Finally, all answer candidates are output in an ordered list. Duplicate answers are eliminated, but answer frequency information is added to each answer in this final list.

3 Evaluation and Results

We submitted two runs for each language pair (edin041deen, edin042deen, edin041fren, edin042fren), differing in the way reranking of answers was executed. The answers of the first runs for each language pair were ranked using the formula $Rank = 0.2 * S + 0.8 * F$, the answers of the second runs were ranked using the formula $Rank = 0.8 * S + 0.2 * F$ for location and measure question types, and on $Rank = 1.0 * S$ for all other question types. (Here S is the normalised DRS-score and F the normalised frequency.) The weights were estimated on the basis of running QED on TREC-2003 data. The second runs were expected to perform better.

For both languages, the second runs preformed the best, with an overal accuracy of 17.00% for German and 20.00% for French. The better scores for French are due to the differences in accuracy of the machine translation components (more time was invested in the French to English MT). Separate results for the factoid and definition questions are listed in Table 1 and Table 2.

Table 1: CLEF-2003 Performance of QED on Factoid Questions

Run	Right	Inexact	Unsupported	Accuracy
edin041deen	24	4	1	13.33%
edin042deen	29	5	0	16.11%
edin041fren	32	4	0	17.78%
edin042fren	37	6	0	20.56%

Table 2: CLEF-2003 Performance of QED on Definition Questions

Run	Right	Inexact	Unsupported	Accuracy
edin041deen	4	1	0	20.00%
edin042deen	5	2	0	25.00%
edin041fren	1	2	0	5.78%
edin042fren	3	1	0	15.00%

For the German edin041deen and edin042deen runs, the answer-string “NIL” was returned 47 times, and correctly returned 7 times (14.89%). For the French edin041fren and edin042fren, the answer-string ”NIL” was returned 70 times, and correctly returned 11 times (15.71%). The confidence-weighted score for the four runs varied between 0.04922 and 0.05889, which is probably low compared to other systems.

4 Conclusion and Future Work

We have presented our extensions to QED to enable it for cross-lingual Q&A. Our approach consisted of composing existing software (with minor enhancements) for machine translation and question answering in a sequential pipeline. The translation was enhanced using pattern replacements to correct systematic mistakes. We obtained an accuracy of 16% (German to English) and 20% (French to English), respectively,

for answering factoid questions. For definition questions, obtained an accuracy of 25% (German to English) and 15% (French to English), respectively. Definition questions constituted a minor portion of the test set.

For future work, we consider using several competing MT systems in a parallel architecture. Automatic MT evaluation scores like Bleu [PRWZ01] could also be considered to select the best translation from a set of candidate machine translation if multiple engines are available. Questions translated by multiple MT systems could also be used together as query expansions. Another proposed extension is recognition (and alignment) of Named Entities in source and target questions to avoid literal translations of proper nouns (for instance, *Spielberg→play mountain* and *Neufeld→new field*).

With regards to the IR component of QED, answer recall after information retrieval and tiling was found low (about 30% of correct answers were not contained after these phases). This is most likely due to impedance mismatch between retrieval and tiling components and the current lack of question-type specific query expansion, and the absense of query relaxation in case no appropriate answers can be found.

The ability to process a large number of highly ranked passages is bound by the time taken by the parser. We are planning to accelerate parsing using a supertagging-based statistical parser [BCS$^+$04] in the next version of QED. This parser, based on CCG, will not only give us a gain in speed, but is also expected to increase the coverage and accuracy of the parser.

Acknowledgements

We are grateful to Steve Clark, James Curran, Malvina Nissim, and Bonnie Webber for assistance and helpful discussions, and would like to thank the system administrators Bill Hewitt and Andrew Woods for their computing support. Special thanks to John Carroll for his help with the RADISP parser, and in general to all authors of all external progams we utilized for making them available.

Alex is supported by Scottish Enterprise Edinburgh-Stanford Link (R36759), the Economic and Social Research Council, UK and the School of Informatics, University of Edinburgh. Dalmas is supported by the School of Informatics, University of Edinburgh. Leidner is supported by the German Academic Exchange Service (DAAD) under scholarship D/02/01831 and by Linguit GmbH (research contract UK-2002/2).

References

[BC02] Ted Briscoe and John Carroll. Robust accurate statistical annotation of general text. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation*, pages 1499–1504, Las Palmas, Gran Canaria, 2002.

[BCS$^+$04] Johan Bos, Stephen Clark, Mark Steedman, James R. Curran, and Julia Hockenmaier. Wide-coverage semantic representations from a CCG parser. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING '04)*, Geneva, Switzerland, 2004.

[CC03a] James R. Curran and Stephen Clark. Investigating GIS and smoothing for maximum entropy taggers. In *Proceedings of the 11th Annual Meeting of the European Chapter of the Association for Computational Linguistics (EACL'03)*, pages 91–98, Budapest, Hungary, 2003.

[CC03b] James R. Curran and Stephen Clark. Language independent NER using a maximum entropy tagger. In *Proceedings of the Seventh Conference on Natural Language Learning (CoNLL-03)*, pages 164–167, Edmonton, Canada, 2003.

[Fel98] Christiane Fellbaum, editor. *WordNet. An Electronic Lexical Database*. The MIT Press, 1998.

[KR93] Hans Kamp and Uwe Reyle. *From Discourse to Logic; An Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and DRT*. Kluwer, Dordrecht, 1993.

[LBD$^+$04] Jochen L. Leidner, Johan Bos, Tiphaine Dalmas, James R. Curran, Stephen Clark, Colin J. Bannard, Mark Steedman, and Bonnie Webber. The QED open-domain answer retrieval system for TREC 2003. In *Proceedings of the Twelfth Text Retrieval Conference (TREC 2003)*, NIST Special Publication 500-255, pages 595–599, Gaithersburg, MD, 2004.

[MRV+03] Bernardo Magnini, Simone Romagnoli, Alessandro Vallin, Jesús Herrera, Anselmo Peñas, Víctor Peinado, Felisa Verdejo, and Maarten de Rijke. Creating the DISEQuA corpus: a test set for multilingual question answering. In Carol Peters, editor, *Working Notes for the CLEF 2003 Workshop*, Trondheim, Norway, 2003.

[PRWZ01] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. Technical Report RC22176 (W0109-022), IBM Thomas J. Watson Research Center, 2001.

[Voo04] Ellen M. Voorhees. Overview of TREC 2003. In *Proceedings of the Twelfth Text Retrieval Conference (TREC 2003)*, NIST Special Publication 500-255, pages 1–13, Gaithersburg, MD, 2004.