A Cross-Lingual German-English Framework for Open-Domain Question Answering

Bogdan Sacaleanu and Günter Neumann

LT-Lab, DFKI, Saarbrücken, Germany
{bogdan, neumann}@dfki.de

Abstract. The paper describes QUANTICO, a cross-language open domain question answering system for German and English. The main features of the system are: use of preemptive off-line document annotation with syntactic information like chunk structures, apposition constructions and abbreviation-extension pairs for the passage retrieval; use of online translation services, language models and alignment methods for the cross-language scenarios; use of redundancy as an indicator of good answer candidates; selection of the best answers based on distance metrics defined over graph representations. Based on the question type two different strategies of answer extraction are triggered: for factoid questions answers are extracted from best IR-matched passages and selected by their redundancy and distance to the question keywords; for definition questions answers are considered to be the most redundant normalized linguistic structures with explanatory role (i.e., appositions, abbreviation's extensions). The results of evaluating the system's performance by CLEF were as follows: for the best German-German run we achieved an overall accuracy (ACC) of 42.33% and a mean reciprocal rank (MRR) of 0.45; for the best English-German run 32.98% (ACC) and 0.35 (MRR); for the German-English run 17.89% (ACC) and 0.17 (MRR).

1 Introduction

QUANTICO is a cross-language open domain question answering system developed for both English and German factoid and definition question. It uses a common framework for both monolingual and cross-language scenarios, with different workflow settings for each task and different configurations for each type of question. For tasks with different languages on each end of the information flow (questions and documents) we cross the language barrier rather on the question than on the document side by using free online translation services, linguistic knowledge and alignment methods. An important aspect of QUANTICO is the triggering of specific answering strategies by means of control information that has been determined by the question analysis tool, e.g. question type and expected answer type, see [3] for more details. Through the offline annotation of the document collection with several layers of linguistic information (chunks, appositions, named entities, sentence boundaries) and their use in the retrieval process, more accurate and reliable information units are being considered for answer extraction, which is based on the assumption that redundancy is a good indicator of information suitability. The answer selection component normalizes

C. Peters et al. (Eds.): CLEF 2006, LNCS 4730, pp. 328 – 338, 2007.

and represents the context of an answer candidate as a graph and computes its appropriateness in terms of the distance between the answer and question keywords.

We will begin giving a short overview of the system and presenting its working for both factoid and definition questions in monolingual and cross-language scenarios. We will then continue with a short description of each component and close the paper with the presentation of the CLEF evaluation results and the error analysis outcome.

2 System Overview

QUANTICO uses a common framework for both monolingual and cross-language scenarios, but with different configurations for each type of question (definition or factoid) and different workflow settings for each task (DE2DE, DE2EN or EN2DE).

Concerning the workflow settings, the following things are to be mentioned. For the monolingual scenario (DE2DE) the workflow is as follows (according to the architecture in the Figure 1): 1-4-5-6/7 with the last selection depending on the question type. For a cross-language scenario, the workflow depends on the language of the question: for German questions and English documents (DE2EN) the workflow is 1-2-3-4-5-6/7, that is, the question is first analyzed, then translated and aligned to its translations, so that based on the generated *QAObj* and the alignments a new English *QAObj* is being computed; for English questions and German documents (EN2DE) the workflow is 2-1-4-5-6/7, that is, the question is first translated and then the best translation – determined according to linguistic completeness – is being analyzed resulting in a *QAObj*. The difference in the system's workflow for the cross-language scenario comes with our choice of analyzing only German questions, since our analysis component, based on the SMES parser [1], is very robust and accurate. In

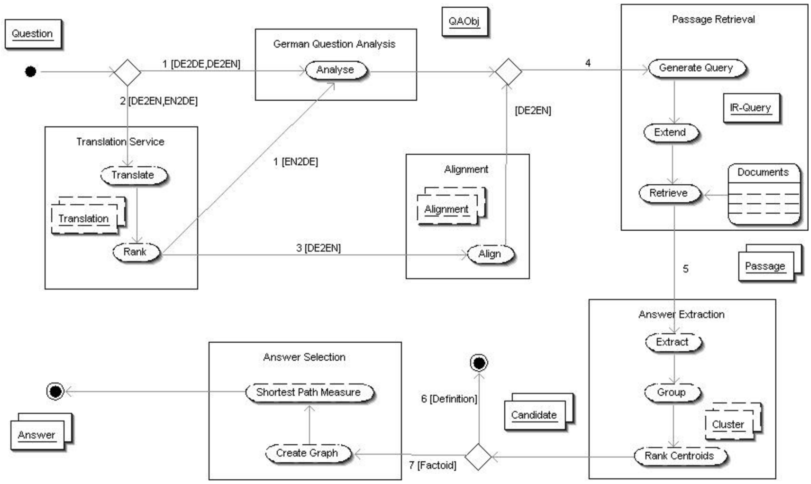

Fig. 1. System Architecture

the presence of a *Question Analysis* component with similar properties for English questions, the workflow would be the same (1-2-3-4-5-6/7) independent of the question's language.

Regarding the component configurations for each type of question (definition or factoid) the difference is to be noted only in the *Passage Retrieval* and *Answer Extraction* components. While the *Retrieve* process for the factoid questions builds on classic Information Retrieval methods, for definition questions is merely a look-up procedure in a repository of offline extracted syntactic structures as appositions, chunks and abbreviation-extension pairs. For the *Answer Extraction* component the distinction consists in different methods of computing the clusters of candidate answers: for factoid question, where the candidates are usually named entities or chunks, is based on co-reference (*John* ~ *John Doe*) and stop-word removal (*of death* ~ *death*), while for definition questions, where candidates can vary from chunks to whole sentences, is based on topic similarity (*Italian designer* ~ *the designer of a new clothes collection*).

3 Component Description

Following is a description of QUANTICO's individual components along with some examples.

3.1 Question Analysis

In the context of a QA system or information search in general, we interpret the result of a NL question analysis as a *declarative description of search strategy and control information*, see [3]. Consider, for example, the NL question result for the question "*In welcher Stadt fanden 2002 die olympischen Winterspile statt?" (The Olympic winter games took place 2002 in which town?)*, where the value of the tag *a-type* represents the expected answer type, *q-type* the answer control strategy, and *q-focus* and *q-scope* additional constraints for the search space:

```
<QOBJ msg="quest" id="qId0" lang="de" score="1">
  <NL-STRING id="qId0">
    <SOURCE id="qId0" lang="de">In welcher Stadt fanden 2002 die
        olympischen Winterspiele statt?</SOURCE>
    </NL-STRING>
  <QA-control>
    <Q-FOCUS>Stadt</Q-FOCUS>
    <Q-SCOPE>stattfind_winter#spiel</Q-SCOPE>
    <Q-TYPE restriction="TEMP">C-COMPLETION</Q-TYPE>
    <A-TYPE type="atomic">LOCATION</A-TYPE>
    </QA-control>
    <KEYWORDS>
       <KEYWORD id="kw0" type="UNIQUE">
        <TK pos="V" stem="statt#find">fanden</TK>
        </KEYWORD>
      <KEYWORD id="kw1" type="UNIQUE">
```

```
         <TK pos="N" stem="stadt">Stadt</TK>
         </KEYWORD>
        <KEYWORD id="kw2" type="UNIQUE">
         <TK pos="NUMERAL" stem="2002">2002</TK>
         </KEYWORD>
        <KEYWORD id="kw3" type="UNIQUE">
         <TK pos="A" stem="olympisch">olympischen</TK>
         </KEYWORD>
        <KEYWORD id="kw4" type="UNIQUE">
         <TK pos="N" stem="winter#spiel">Winterspiele</TK>
         </KEYWORD>
    </KEYWORDS>
    <EXPANDED-KEYWORDS />
    <NE-LIST>
      <NE id="ne0" type="DATE">2002</NE>
    </NE-LIST>
</QOBJ>
```

Parts of the information can already be determined on basis of local lexico-syntactic criteria (e.g., for the Wh-phrase *where* we can simply infer that the expected answer type is *location*). However, in most cases we have to consider larger syntactic units in combination with the information extracted from external knowledge sources. For example for a definition question like *"What is a battery?"* we have to combine the syntactic and type information from the verb and the relevant NP (e.g., combine definite/indefinite NPs together with certain auxiliary verb forms) in order to distinguish it from a description question like *"What is the name of the German Chancellor?"* In our QAS, we are doing this by following a two-step parsing schema:

- in a first step a full syntactic analysis is performed using the robust parser SMES and
- in a second step a question-specific semantic analysis.

During the second step, the values for the question tags *a-type*, *q-type, q-focus* and *q-scope* are determined on basis of syntactic constraints applied on the dependency analysis of relevant NP and VP phrases (e.g., considering agreement and functional roles), and by taking into account information from two small knowledge bases. They basically perform a mapping from linguistic entities to values of the questions tags, e.g., trigger phrases like *name_of*, *type_of*, *abbreviation_of* or a mapping from lexical elements to expected answer types, like *town*, *person*, *and president.* For German, we additionally perform a *soft retrieval match* to the knowledge bases taking into account online compound analysis and string-similarity tests. For example, assuming the lexical mapping *Stadt → LOCATION* for the lexeme *town*, then automatically we will also map the nominal compounds *Hauptstadt* (capital) and *Großstadt* (large city) to *LOCATION.*

A main aspect in the adaptation and extension of the question analysis component for the Clef-2006 task concerned the recognition of the question type, i.e., simple factoid and list factoid questions, definition questions and the different types of the temporally restricted questions. Because of its high degree of modularity of the question analysis component, the extension only concerns the semantic analysis subcomponent. Here, additional syntactic-semantic mapping constraints have been implemented that enriched the coverage of the question grammar, where we used the question set of the previous Clef campaigns as our development set.

3.2 Translation Services and Alignment

We have used two different methods for responding questions asked in a language different from the one of the answer-bearing documents. Both employ online translation services (Altavista, FreeTranslation, etc.) for crossing the language barrier, but at different processing steps, i.e. before and after formalizing the user information need into a *QAObj*.

The *a priori–method* translates the question string in an earlier step, resulting in several automatic translated strings, of which the best one is analyzed by the *Question Analysis* component and passed on to the *Passage Retrieval* component. This is the strategy we use in an English–German cross-lingual setting. To be more precise: the English source question is translated into several alternative German questions using online MT services. Each German question is then parsed with SMES, our German parser. The resulting query object is then weighted according to its linguistic well–formedness and its completeness with respect to the query information (question type, question focus, answer–type).

The assumption behind this weighting scheme is that "a translated string is of greater utility for subsequent processes than another one, if its linguistic analysis is more complete or appropriate."

The *a posteriori–method* translates the formalized result of the *Query Analysis* component by using the question translations, a language modeling tool and a word alignment tool for creating a mapping of the formal information need from the source language into the target language. We illustrate this strategy in a German–English setting along two lines (using the following German question as example: "*In welchem Jahrzehnt investierten japanische Autohersteller sehr stark?*"):

- translations as returned by the on-line MT systems are being ranked according to a language model
  - *In which decade did Japanese automakers invest very strongly? (0.7)*
  - *In which decade did Japanese car manufacturers invest very strongly? (0.8)*
- translations with a satisfactory degree of resemblance to a natural language utterance (i.e. linguistically well-formedness), given by a threshold on the language model ranking, are aligned based on several filters: dictionary filter - based on MRD (machine readable dictionaries), PoS filter - based on statistical part-of-speech taggers, and cognates filter - based on string similarity measures (dice coefficient and LCSR (lowest common substring ratio)).

In: [in:1.0] 1.0
welchem: [which:0.5] 0.5
Jahrzehnt: [decade:1.0] 1.0
investierten: [invest:1.0] 1.0
japanische: [Japanese:0.5] 0.5
Autohersteller: [car manufacturers:0.8, auto makers:0.1] 0.8
sehr: [very:1.0] 1.0
stark: [strongly:0.5] 0.5

3.3 Passage Retrieval

The preemptive offline document annotation refers to the process of annotating the document collections with information that might be valuable during the retrieval process by increasing the accuracy of the hit list. Since the expected answer type for factoid questions is usually a named entity type, annotating the documents with named entities provides for an additional indexation unit that might help to scale down the range of retrieved passages only to those containing the searched answer type. The same practice applies for definition questions given the known fact that some structural linguistic patterns (appositions, abbreviation-extension pairs) are used with explanatory and descriptive purpose. Extracting these kinds of patterns in advance and looking up the definition term among them might return more accurate results than those of a search engine.

The *Generate Query* process mediates between the question analysis result *QAObj* (answer type, focus, keywords) and the search engine (factoid questions) or the repository of syntactic structures (definition questions) serving the retrieval component with information units (passages). The *Generate Query* process builds on an abstract description of the processing method for every type of question to accordingly generate the *IRQuery* to make use of the advanced indexation units. For example given the question "*What is the capital of Germany?*", since named entities were annotated during the offline annotation and used as indexing units, the *Query Generator* adapts the *IRQuery* so as to restrict the search only to those passages having at least two locations: one as the possible answer (*Berlin*) and the other as the question's keyword (*Germany*), as the following example shows:

+text:capital+text:Germany+neTypes:LOCATION +LOCATION:2.

It is often the case that the question has a semantic similarity with the passages containing the answer, but no lexical overlap. For example, for a question like "*Who is the French prime-minister?*", passages containing "*prime-minister X of France*", "*prime-minister X … the Frenchman*" and "*the French leader of the government*" might be relevant for extracting the right answer. The *Extend* process accounts for bridging this gap at the lexical level, either through look-up of unambiguous resources or as a side-effect of the translation and alignment process (see [4]).

In the context of the participation to CLEF two different settings have been considered for the retrieval of relevant passages for factoid questions: one in which a passage consists of only a sentence as retrieval unit from a document, and a second one with a window of three adjoining sentences for a passage. Concerning the query generation, only keywords with following part-of-speeches have been considered for retrieval: nouns, adjective and verbs, whereby only nouns are mandatory to occur in the matching relevant passages. In case of empty hit list retrieval, the query undergoes a relaxation process maintaining only the focus of the question and the expected answer type (as computed by the *Analyse* component) as mandatory items:

Question: Which country did Joe visit?
IR-Query: +neTypes:LOCATION +text:country^4 +text:Joe text:visit
Relaxed IR-Query: +neTypes:LOCATION +text:country^4 text:Joe text:visit

3.4 Answer Extraction

The *Answer Extraction* component is based on the assumption that the redundancy of information is a good indicator for its suitability. The different configurations of this component for factoid and definition questions reflect the distinction of the answers being extracted for these two question types: simple chunks (i.e. named entities and basic noun phrases) and complex structures (from phrases through sentences) and their normalization. Based on the control information supplied by the *Analyse* component (*q-type*), different extraction strategies are being triggered (noun phrases, named entities, definitions) and even refined according to the *a-type* (definition as sentence in case of an OBJECT, definition as complex noun phrase in case of a PERSON).

Whereas the *Extract* process for definition questions is straightforward for cases in which the offline annotation repository lookup was successful, in other cases it implies an online extraction of those passage-units only that might bear a resemblance to a definition. The extraction of these passages is attained by matching them against a lexico-syntactic pattern of the form:

<*Searched Concept*> <*definition verb*> .+

whereby <*definition verb*> is being defined as a closed list of verbs like “is”, “means”, ”signify”, “stand for” and so on.

For factoid questions having named entities or simple noun phrases as expected answer type the *Group* (normalization) process consists in resolving cases of coreference, while for definition questions with complex phrases and sentences as possible answers more advanced methods are being involved. The current procedure for clustering definitions consists in finding out the focus of the explanatory sentence or the head of the considered phrase. Each cluster gets a weight assigned based solely on its size (definition questions) or using additional information like the average of the IR-scores and the document distribution for each of its members (factoid questions).

3.5 Answer Selection

Using the most representative sample (centroid) of the answer candidates’ best-weighed clusters, the *Answer Selection* component sorts out a list of top answers based on a distance metric defined over graph representations of the answer’s context. The context is first normalized by removing all functional words and then represented as a graph structure. The score of an answer is defined in terms of its distance to the question concepts occurring in its context and the distance among these.

In the context of the participation to CLEF a threshold of five best-weighed clusters has been chosen and all their instances, not only their centroids, have been considered for a thorough selection of the best candidate.

4 Evaluation Results

We participated in three tasks: DE2DE (German to German), EN2DE (English to German) and DE2EN (German to English), with two runs submitted for each of the first two tasks. The second runs submitted (*dfki062*) were distinct in that the context of the retrieved passages was consisting of three sentences compared to the other runs (*dfki061*) with only one sentence per passage. A detailed description of the achieved results can be seen in Table 1.

Compared to the results from last year [3], we were able to keep our performance for the monolingual German task DE2DE (2005 edition: 43.50%). For the task English to German we were able to improve our result (2005 edition: 25.50%) and for the task German to English we observed a decrease (2005 edition: 23.50%).

Table 1. System Performance - Details

Run ID	Right		W	X	U	F	D	T	P@N L	NIL [20]		
	#	%	#	#	#	%	%	%	%	F	P	R
dfki061dede$_{M}$	80	42.32	95	6	8	38.81	56.75	29.54	25.93	0.35	0.28	0.45
dfki062dede$_{M}$	63	33.33	114	4	8	30.92	43.24	22.72	33.33	0.32	0.27	0.4
dfki061endec$_{C}$	62	32.8	117	3	6	28.94	48.64	22.72	10	0.31	0.21	0.6
dfki062endec$_{C}$	50	26.45	130	5	3	22.36	43.24	20.45	10	0.33	0.22	0.65
dfki061deen$_{C}$	34	17.89	147	9	0	17.33	20	22.5	20	0.25	0.17	0.44

Table 2 resumes the distribution of the right, inexact and unsupported answers over the first three ranked positions as delivered by our system, as well as the accuracy and MRR for each of the runs. The figures refer only to the runs submitted for the DE2DE and EN2DE tasks, since the DE2EN task evaluated just the first answer presented by the systems.

Two things can be concluded from the answer distribution of Table 2: first, there are a fair number of inexact and unsupported answers that show performance could be

Table 2. Distribution of Answers

Run ID	# Right			# inexact			# Unsupported			Accuracy	MRR
	1st	2nd	3rd	1st	2nd	3rd	1st	2nd	3rd		
dfki061dede$_{M}$	80	8	7	6	6	1	8	4	1	42.32	45.67
dfki062dede$_{M}$	63	15	3	4	5	3	8	0	2	33.33	37.83
dfki061endec$_{C}$	62	5	7	3	4	2	6	3	0	32.8	35.36
dfki062endec$_{C}$	50	10	2	5	4	2	3	2	1	26.45	29.45
dfki061deen$_{C}$	34	-	-	9	-	-	0	-	-	17.89	17.89

improved with a better answer extraction; second, the fair number of right answers among the second and third ranked positions indicate that there is still place for improvements with a more focused answer selection.

5 Error Analysis

Since the actual edition of the question-answer Gold Standard has not yet been released at the time of writing this paper, the error analysis was performed only to the runs submitted for the tasks having German as target language – for which we had access to the question-answer pairs.

The results of the analysis can be grouped along two lines: conceptual and functional. The functional errors relate to the following components used:

- named entity annotation,
- on-line translation services,

while the conceptual ones refer to decisions and assumptions we made during the development of the system:

- answer and supporting evidence are to be found within a sentence
- Answer selection for instances of top five clusters might suffice
- questions and answer contexts share a fair amount of lexical items
- definition extraction strategies are exclusive

Following we will shortly explain the above-mentioned issues and provide some examples for clarity where needed.

Functional – Named Entity

The named entity tool used (LingPipe [5]), being a statistical based entity extractor, has a very good coverage and precision on annotating the document collection, where lots of context data are available, but its performance drops when using the same model for annotating short questions. Since our Query Generator component builds on using named entities as mandatory items to restrain the amount of relevant passages retrieved, failure to consistently annotate entities on both sides (question and document) results in most cases in unusable units of information and therefore wrong answers.

Functional – Translation Services

Failure of correctly translating the question from a source language to the target language can have critical results when the information being erred on represents the focus or belongs to the scope of the question. Following are several examples of miss-translations that resulted in incorrect IR-queries generation and therefore wrong answers.

“Lord of the Rings” → “Lord der Ringe” vs. “Herr der Ringe”
“states” → “Zustände, Staate” vs. „Bundesländer“
„high“ → „hoh, stark“ vs. „hoch“
„Pointer Stick“ → „Zeigerstock“ vs. „Pointer Stick“
„Mt.“ (Mount) → „Millitorr“ vs. „Mt.“

Conceptual – Answer and Supporting Evidence within a Sentence

Considering a sentence as the primary information and retrieval unit together with using the named entities as index tokens and querying terms, produced very good results in case of relatively short factoid questions where the answer and the supporting evidence (as question keywords) are to be found within the same sentence. Nevertheless, a fair amount of longer questions can only be answered by either looking at immediately adjoining sentences or using anaphora and co-reference resolution methods between noun phrases. Although LingPipe has a named entity co-reference module, it does not cover non-NE cases, which account for correctly answering some questions.

Conceptual – Answer Selection on Top Five Clusters

Looking to cover the scenario described in the previous issue, a run using three adjacent sentences as retrieval unit has been evaluated (*dfki062*). Correctly identifying answers to most of the questions by assuming scattered supporting evidence over adjoining sentences, this method invalidated some of the correctly answered factoid questions in the previous setting. The reason for that was that increasing the size of the retrieval unit produced more clusters of possible candidates and in several cases the clusters containing the correct answer were not ranked among top five and were not considered for a final selection.

Conceptual – Lexical Items Sharing between Question and Answer Context

The assumption that the question and the context of the correct answer share a fair amount of lexical items is being reflected both in the IR-query generation, although the *Expand* component might lessen it, and the answer selection. This assumption impedes the selection of correct answers that have a high semantic but little lexical overlap with the question. Some examples of semantic related concepts with no lexical overlap are as follow:

birthplace <> born
homeland <> born
monarch <> king
profession <> designer

Conceptual – Definition Extraction Strategies are Exclusive

Four extraction strategies are employed to find the best correct answer for definition questions: looking for abbreviation-extension pairs, extracting named entities with their appositions, looking for the immediately left-adjoining noun phrase and extracting definitions according to some lexico-syntactic patterns. Although the methods are quite accurate, there are cases in which either they extract false positives or the definition is inexact. Since all four strategies are exclusive, when one of them has been triggered it returns on finding a possible definition without giving a chance to any other strategy to complete. Because of this competing nature of the actual implementation, some wrong or inexact definitions are preferred over more accurate explanations.

6 Conclusions

We have presented a framework for both monolingual and cross-lingual question answering for German/English factoid and definition questions. Based on a thorough analysis of the question, different strategies are considered and alternative work-flows and components are triggered depending on the question type. Through the preemptive off-line annotation we placed some domain knowledge (i.e. named entities, appositions, abbreviations) on the document collection, so that a more effective passage retrieval approach can be used.

Intuitive assumptions regarding the unit of retrieval granularity (i.e. at sentence level) and the overlap of lexical information between the question and the relevant units have lead to promising results in the CLEF evaluation campaign, though the error analysis revealed some cases for which these premises do not hold. These are the entry points for further research to be pursued, both at the functional and conceptual level.

Acknowledgments

The work presented in this paper has been funded by the BMBF projects HyLaP (FKZ: 01 IW F02) and SmartWeb (FKZ: 01 IMD 01A).

References

1 Neumann, G., Piskorski, J.: A shallow text processing core engine. Computational Intelligence 18(3), 451–476 (2002)

2 Neumann, G., Sacaleanu, B.: Experiments on Robust NL Question Interpretation and Multi-layered Document Annotation for a Cross-Language Question/Answering System. In: Peters, C., Clough, P.D., Gonzalo, J., Jones, G.J.F., Kluck, M., Magnini, B. (eds.) CLEF 2004. LNCS, vol. 3491, pp. 411–422. Springer, Heidelberg (2005)

3 Neumann, G., Sacaleanu, B.: Experiments on Cross-Linguality and Question-Type Driven Strategy Selection for Open-Domain QA. In: Peters, C., Gey, F.C., Gonzalo, J., Müller, H., Jones, G.J.F., Kluck, M., Magnini, B., de Rijke, M., Giampiccolo, D. (eds.) CLEF 2005. LNCS, vol. 4022, pp. 429–438. Springer, Heidelberg (2006)

4 Sacaleanu, B., Neumann, G.: Cross-Cutting Aspects of Cross-Language Question Answering Systems. In: Proceedings of the EACL workshop on Multilingual Question Answering - MLQA'06, Trento, Italy (2006)

5 Alias-i.LingPipe1.7 (2006), http://www.alias-i.com/lingpipe