Cross-Lingual Question Answering

Dissertation
zur Erlangung des akademischen Grades eines
Doktors der Philosophie
der Philosophischen Fakultäten
der Universität des Saarlandes

vorgelegt von

Bogdan Eugen Sacaleanu
aus Iasi, Rumänien
Saarbrücken, 2012

Dekan:	Univ.-Prof. Dr. Erich Steiner
Berichterstatter:	Univ.-Prof. Dr. Hans Uszkoreit
	Univ.-Prof. Dr. Martin Volk

To my son Noah

Acknowledgments

I would like to thank first of all Professor Hans Uszkoreit, Professor of Computational Linguistics at the University of Saarland, for giving me the opportunity to pursue this work under his supervision and for his continuous support and guidance.

I would like to thank also Privat Dozent Dr. Günter Neumann who constantly offered interesting feedback and supportive conversations along the time.

I would like to thank to Professor Martin Volk, Professor of Computational Linguistics at the University of Zurich, who helped me with a thorough revision of my work and who was there for me when needed.

I would like to thank my wife Corina and my sister Daniela for their moral support and encouragement during the writing of this dissertation.

Short Summary in German

Innerhalb der letzten zehn Jahre hat sich Question Answering zu einem intensiv erforschten Themengebiet gewandelt, es stellt den nächsten Schritt des Information Retrieval dar, mit dem Bestreben einen präziseren Zugang zu großen Datenbeständen von verfügbaren Informationen bereitzustellen. Das Question Answering setzt auf die Information Retrieval-Technologie, um mögliche relevante Daten zu suchen, kombiniert mit weiteren Techniken zur Verarbeitung von natürlicher Sprache, um mögliche Antwortkandidaten zu identifizieren und diese anhand von Hinweisen oder Anhaltspunkten entsprechend der Frage als richtige Antwort zu akzeptieren oder als unpassend zu erklären.

Während ein Großteil der Forschung den einsprachigen Kontext voraussetzt, wobei Frage- und Antwortdokumente ein und dieselbe Sprache teilen, konzentrieren sich aktuellere Ansätze auf sprachübergreifende Szenarien, in denen die Frage- und Antwortdokumente in unterschiedlichen Sprachen vorliegen.

Im Kontext des Information Retrieval existieren drei bekannte Ansätze, die versuchen auf unterschiedliche Art und Weise die Sprachbarriere zu überwinden: durch die Übersetzung der Frage, durch die Übersetzung der Dokumente oder durch eine Angleichung von sowohl der Frage als auch der Dokumente zu einer gemeinsamen interlingualen Darstellung.

Wir präsentieren ein sprachübergreifendes Question Answering System vom Englischen ins Deutsche, das sowohl für Faktoid- als auch für Definitionsfragen funktioniert. Dazu verwenden wir ein einsprachiges deutsches System und übersetzen die Fragen vom Englischen ins Deutsche. Zwei unterschiedliche Techniken der Übersetzung werden untersucht:

- die direkte Übersetzung der englischen Fragestellung ins Deutsche und
- die Abbildungs-basierte Übersetzung, die eine Zwischendarstellung verwendet, um die „Semantik“ der ursprünglichen Frage zu erfassen und in die Zielsprache zu übersetzen.

Für beide aufgelisteten Übersetzungstechniken werden zwei Übersetzungsquellen verwendet: zweisprachige Wörterbücher und maschinelle Übersetzung. Die Zwischendarstellung erfasst die Semantik der Frage in Bezug auf die Art der Frage (*QType*), den erwarteten Antworttyp (*EAType*) und Fokus, sowie die Informationen, die den Ablauf des Frage-Antwort-Prozesses steuern.

Das deutschsprachige Question Answering System kann sowohl Faktoid- als auch Definitionsfragen beantworten und basiert auf mehreren Prämissen:

- Fakten und Definitionen werden in der Regel lokal auf Satzebene ausgedrückt;
- Die Nähe von Konzepten innerhalb eines Satzes kann auf eine semantische Verbindung hinweisen;
- Bei Faktoidfragen ist die Redundanz der Antwortkandidaten ein guter Indikator für deren Eignung;
- Definitionen von Begriffen werden mit festen sprachlichen Strukturen ausgedrückt, wie Appositionen, Modifikatoren, Abkürzungen und Erweiterungen.

Umfangreiche Auswertungen des einsprachigen Systems haben gezeigt, dass die oben genannten Hypothesen in den meisten Fällen wahr sind, wenn es um eine ziemlich große Sammlung von Dokumenten geht, wie bei der im CLEF Evaluationsforum verwendeten Version.

Abstract

Question Answering has become an intensively researched area in the last decade, being seen as the next step beyond Information Retrieval in the attempt to provide more concise and better access to large volumes of available information. Question Answering builds on Information Retrieval technology for a first touch of possible relevant data and uses further natural language processing techniques to search for candidate answers and to look for clues that accept or invalidate the candidates as right answers to the question. Though most of the research has been carried out in monolingual settings, where the question and the answer-bearing documents share the same natural language, current approaches concentrate on cross-language scenarios, where the question and the documents are in different languages. Known in this context and common with the Information Retrieval research are three methods of crossing the language barrier: by translating the question, by translating the documents or by aligning both the question and the documents to a common inter-lingual representation.

We present a cross-lingual English to German Question Answering system, for both factoid and definition questions, using a German monolingual system and translating the questions from English to German. Two different techniques of translation are evaluated:

- direct translation of the English input question into German and
- transfer-based translation, by using an intermediate representation that captures the "meaning" of the original question and is translated into the target language.

For both translation techniques two types of translation tools are used: bilingual dictionaries and machine translation. The intermediate representation captures the semantic meaning of the question in terms of Question Type (*QType*), Expected Answer Type (*EAType*) and Focus, information that steers the workflow of the question answering process.

The German monolingual Question Answering system can answer both factoid and definition questions and is based on several premises:

- facts and definitions are usually expressed locally at the level of a sentence and its surroundings;
- proximity of concepts within a sentence can be related to their semantic dependency;
- for factoid questions, redundancy of candidate answers is a good indicator of their suitability;
- definitions of concepts are expressed using fixed linguistic structures such as appositions, modifiers, and abbreviation extensions.

Extensive evaluations of the monolingual system have shown that the above mentioned hypothesis holds true in most of the cases when dealing with a fairly large collection of documents, like the one used in the CLEF evaluation forum.

Table of Contents

1 INTRODUCTION 1

1.1 RESEARCH QUESTIONS 2
1.2 CONTRIBUTIONS 5
1.3 OUTLINE OF THIS THESIS 5

2 TECHNIQUES RELATED TO QUESTION ANSWERING 7

2.1 INFORMATION RETRIEVAL 7
2.2 INFORMATION EXTRACTION 10

3 STATE OF THE ART FOR QUESTION ANSWERING 13

3.1 MONOLINGUAL QA 15
3.1.1 Question Analysis 15
3.1.2 Information Retrieval 17
3.1.3 Answer Extraction and Selection 19
3.2 CROSS-LINGUAL QA 21

4 EVALUATION METHODOLOGY 27

4.1.1 CLEF Evaluation Corpus 28
4.1.2 Evaluation Metrics 31
4.1.3 Component-wise Evaluation 32

5 QUANTICO: A CROSS-LANGUAGE QUESTION ANSWERING SYSTEM 35

5.1 MONOLINGUAL QA 39
5.2 CROSS-LANGUAGE METHODS 47
5.2.1 Direct Translation 49
5.2.2 Transfer-based Translation 50
5.3 SUMMARY 57

6 QUESTION ANALYSIS 59

6.1 GERMAN ANALYSIS 60
6.2 ENGLISH ANALYSIS 62
6.2.1 Syntactic Parser 63
6.2.2 Semantic Interpreter 64
6.3 EVALUATION 69
6.4 SUMMARY 70

7 INFORMATION UNIT RETRIEVAL 72

7.1 TEXT PROCESSING 73
7.1.1 LingPipe 73
7.1.2 Preemptive linguistic annotation 75
7.2 SEARCH ENGINE 77
7.2.1 Apache Lucene 78
7.2.2 Indexing Sentences 80
7.2.3 Scoring Schemes 81
7.3 QUERY FORMULATION 83
7.3.1 Query Generation 83
7.3.2 Query Extension 85
7.4 EVALUATION 86
7.4.1 Monolingual Experiments 87
7.4.2 Cross-lingual Experiments 90
7.5 SUMMARY 101

8 ANSWER EXTRACTION 102

8.1 ANSWERS TO FACTOID QUESTIONS 104

8.1.1 Candidates Extraction by Redundancy in Centroid Ranker *104*
8.1.2 Answer Selection by Proximity *105*
8.2 Answers to Definition Questions 106
8.2.1 Appositions *107*
8.2.2 Acronyms *108*
8.2.3 Lexical Definition *110*
8.2.4 Hypernyms *111*
8.3 Evaluation 112
8.4 Summary 114

9 CONCLUSIONS AND FUTURE WORK 115

9.1 Summary of contributions and answers to research questions 118
9.2 Future Work 119

ANNEXES 123

Annex 1 – Sound Shifts between English and German 123
Annex 2 – DROOLS Rules for English Question Analysis 127

BIBLIOGRAPHY 139

RESUME 153

List of Figures

FIGURE 1. A GENERIC MONOLINGUAL QA SYSTEM ARCHITECTURE....13
FIGURE 2. A GENERIC CROSS-LINGUAL QA SYSTEM ARCHITECTURE....14
FIGURE 3. FORMAT OF QUESTION-ANSWER PAIRS IN DISEQUA CORPUS....30
FIGURE 4. CONVERSION OF LINKED QUESTIONS. 30
FIGURE 5. IDEAL QA SYSTEM....35
FIGURE 6. SKETCH OF THE LEXICAL SOLUTION. 36
FIGURE 7. COMMON FRAMEWORK OF QUANTICO 40
FIGURE 8. RESULT OF QUESTION ANALYSIS....41
FIGURE 9. DIRECT TRANSLATION METHOD. 49
FIGURE 10. TRANSFER-BASED METHOD. 50
FIGURE 11. MRD TRANSFER-BASED METHOD. 52
FIGURE 12. MT TRANSFER-BASED TRANSLATION. 53
FIGURE 13. INITIAL WORD ALIGNMENT. 54
FIGURE 14. PART-OF-SPEECH FILTERING. 55
FIGURE 15. DIRECTFILTER DICTIONARY LOOK-UP. 55
FIGURE 16. BACKPROPAGATIONFILTER DICTIONARY LOOK-UP. 55
FIGURE 17. OVERLAPFILTER. 56
FIGURE 18. LCSR FILTER. 57
FIGURE 19. OVERLAP FILTER AND FINAL ALIGNMENT. 57
FIGURE 20. QUESTION ANALYSIS ARCHITECTURE. 59
FIGURE 21. RESULT OF GERMAN QUESTION ANALYSIS....61
FIGURE 22. COMMON STRUCTURE OF OPEN QUESTIONS. 63
FIGURE 23. OUTPUT OF STANFORD PARSER. 64
FIGURE 24. FOCUS OF FACTOID QUESTIONS. 67
FIGURE 25. POSSIBLE INSTANCES OF DIFFERENT EA_TYPES....68
FIGURE 26. UNIT RETRIEVAL ARCHITECTURE....72
FIGURE 27. DOCUMENT ANNOTATION WITH LINGPIPE....76
FIGURE 28. DOCUMENT ANNOTATION WITH LINGPIPE - REVISED. 76
FIGURE 29. RESULT OF DIFFERENT LUCENE ANALYZERS. 79
FIGURE 30. LUCENE REPRESENTATION OF A SENTENCE. 81
FIGURE 31. OUT-OF-DOCUMENT SENTENCE COHERENCE. 82
FIGURE 32. QUERY GENERATOR DATA. 84
FIGURE 33. COMPARISON OF DIFFERENT TECHNIQUES OF CROSS-LINGUALITY 93
FIGURE 34. COMPARISON OF DIFFERENT TECHNIQUES OF CROSS-LINGUALITY 94
FIGURE 35. COMPARISON OF DIFFERENT TECHNIQUES OF CROSS-LINGUALITY 95
FIGURE 36. RESULTS OF LEXICAL AND CONCEPTUAL QUERY EXTENSION....96
FIGURE 37. RESULTS OF LEXICAL AND CONCEPTUAL QUERY EXTENSION....96
FIGURE 38. RESULTS OF LEXICAL AND CONCEPTUAL QUERY EXTENSION....97
FIGURE 39. RESULTS OF LEXICAL AND CONCEPTUAL QUERY EXTENSION....98
FIGURE 40. RESULTS OF LEXICAL AND CONCEPTUAL QUERY EXTENSION....98
FIGURE 41. RESULTS OF LEXICAL AND CONCEPTUAL QUERY EXTENSION....99
FIGURE 42. RESULTS OF LEXICAL AND CONCEPTUAL QUERY EXTENSION....100
FIGURE 43. RESULTS OF LEXICAL AND CONCEPTUAL QUERY EXTENSION....100
FIGURE 44. RESULTS OF LEXICAL AND CONCEPTUAL QUERY EXTENSION....101
FIGURE 45. ANSWER EXTRACTION ARCHITECTURE. 102
FIGURE 46. COMPARISON TO CLEF-DE BEST RESULTS FOR FACTOID QUESTION. 117

List of Tables

TABLE 1. COMPARISON OF QUESTION TRANSLATION METHODS. 57
TABLE 2. QUESTION ANALYSIS ACCURACY. 70
TABLE 3. FORWARD INDEX EXAMPLE. 77
TABLE 4. INVERTED INDEX EXAMPLE. 77
TABLE 5. COMPARISON OF QUESTION TRANSLATION METHODS (REVISED). 91
TABLE 6. SPELLING CONSONANT SHIFTS. 125
TABLE 7. SPELLING VOWEL SHIFTS. 126

1 Introduction

Ever since the 1940s, with the introduction of the first digital computers, the idea of having machines take on the burden of assiduous tasks for humans started taking form. One of these tasks was directly related to storage and retrieval of data, as in the area of educational and corporate libraries. With the increasing amount of information being continuously stored, even librarians who are experts at finding and organizing information and at interpreting information needs needed assistance for mastering their work. Both hardware, for data storage, and software, for data retrieval, solutions have been considered over the years, with an unbalanced evolution for these two technologies: while hardware development was rapidly advancing and making possible the storage of huge volumes of data, software for accessing these data was still in its early stages. One effect of this was that relevant information was partly ignored since it was never uncovered, leading in turn to much duplication of effort and work. In response to this information overload, intelligent retrieval systems were developed in an attempt to render the information more accessible, which supported both decision-making and actions based on it.

Information Retrieval (IR) was one of the first fields of research that targeted the development of intelligent retrieval systems to search through large text corpora for documents related to a request. Unfortunately, IR systems merely provided access to the whereabouts of documents related to the information needs, and it was up to the user of such systems to identify the context and assess the relevance of the provided results.

Information Extraction (IE) systems came closer to the goal of providing information related to a given request through automatic extraction of data from documents by filling out predefined templates. In this way, information was presented in the context of the template and referenced in the document, specifying not only its whereabouts, but its context and relationship to the context as well. The information usually consisted of entities and relations between entities reflecting facts about “who

did what to whom, where, when and how". Though very successful, these retrieval systems are by their nature highly specialized and domain dependent, making them harder to port to new types of data and new domains.

Question Answering (QA) systems promise to deliver direct access to the information requested by providing focused, context-supported, concise answers to natural language questions. Question Answering is built on top of existing retrieval technologies, Information Retrieval and Information Extraction specifically, leveraging the best results of its predecessors for enhanced data access, but inheriting some of their known limitations, as well.

One important limitation relates to the cross-linguality, characterized by having the question and answer-bearing texts in different languages. Cross-linguality is particularly important for locating information on the Internet, where resources in various languages are easily accessible. An essential factor of effective cross-lingual QA (CLQA) is the translation process that enables automatic comparison of subject representations between question and documents. There are three known methods of crossing the language barrier: by translating the question, by translating the documents and by translating both the question and the documents to a common inter-lingual representation. Question translation is the most widely used matching strategy for CLQA due to its tractability; that is, the greater simplicity of translating the question than to translate a large set of documents that include the answer.

1.1 Research Questions

This research presents an open-domain, cross-lingual English to German Question Answering system that leverages the performance of a mono-lingual German system by translating the questions into the target language. We compare two different techniques of translation, by directly translating the question and by translating the result of interpreting the question. We also investigate two methods of query expansion for the document retrieval, through synonyms and through related concepts. Issues of term ambiguity during expansion are being dealt with by reducing the limits of the retrieved textual unit to those of a sentence and decreasing the probability of inappropriate meanings to co-occur with keywords from the question. We explore as well several strategies of extracting answers and combining them in a framework for factoid and definition question answering.

We intend to answer the following research questions during this work:

- *Are small sentence-based retrieval units a feasible way of locating relevant information for open-domain Question Answering?*

  Most of the factoid and definition questions are asking about properties of a given concept. Unless the requested information is out of the common sense world, facts about the target concept may already contain the answer. By regarding sentences as the most compact forms of expressing facts, it should be possible to correctly answer questions based on sentences that match the given concept.

- *Is query extension without disambiguation in the context of small-sized retrieval units effective in identifying relevant retrieval units?*

  The purpose of query extension is to bridge the difference between vocabularies of both the user and the document collection. Targeting an increased recall for information retrieval, it often falls short in maintaining precision figures because of the ambiguity of the content words considered for expansion. Automatic methods of disambiguation alleviate this issue by using features of the context in order to choose the right meaning to be extended. Unfortunately, factoid questions are too short to provide enough context and therefore are not suitable for automatic disambiguation techniques. We assume that the small sentence-based contexts provided by the unit retrieval will support the *one sense per collocation* property of human languages, according to which words tend to exhibit only one sense in a given collocation. Accordingly, inappropriate meanings of question words would be inherently filtered out during the retrieval process.

- *Is proximity a good approximation for the linguistic relationship among words in selecting the correct answer?*

  Words in a question have some explicit or implied linguistic relationship between them and a good answer is likely to be one that has the same relationship between those words. The idea of *proximity* is to provide an approximation to matching the linguistic relations between words by increasing the chance to find question words in some relationship, which in turn increases the chance of getting the words in the right relationship.

- *Are there any comparable methods for crossing the language barrier on the question side beside the widespread machine translation?*

  Using machine translation (MT) to translate the question is straightforward and leverages the existence of a mono-lingual QA system with no changes required, although the resulting translation is often syntactically ill formed and therefore inappropriate for a meaningful interpretation of the question. Analyzing the original question upfront and translating the result by way of Machine Readable Dictionaries (MRD) seems to be a feasible alternative, but it brings along issues of ambiguity that might influence the performance of later components. An alternative to using MRD is to align the MT translation back to the original question and use these alignments to translate the result of the question analysis. This new method leverages the machine translation results that inherently disambiguate translations of question words by selecting the most appropriate one based on its context.

- *Do small size retrieval units benefit translation through Machine Readable Dictionaries (MRD) by helping reduce the ambiguity of words in local contexts?*

  One of the main disadvantages of question translation by way of MRD is that the results are ambiguous and may contain translations that are not appropriate to the intended meaning as given by the context of use. Failure to determine the right translation might result in retrieving false positives when searching for relevant data and is responsible for the system's low precision. Instead of employing word sense disambiguation techniques during the translation process, we let the system filter out irrelevant meanings by reducing the size of the unit retrieval. Intuitively, we expect that by narrowing the length of the retrieval unit to that of a sentence, irrelevant meanings of question words will rarely co-occur within the local context of a sentence.

1.2 Contributions

The primary contributions of this research are as follows:

- Development of a scalable cross-lingual framework for Question Answering based on two different techniques of crossing the language barrier: direct machine translation of the question and transfer of the semantic interpretation of a question by way of term translation (machine readable dictionaries, machine translation word alignments).
- Development of a Question Analysis module based on semantic rules defined over syntactic constituents.
- Development of a Unit Retrieval module that uses Named Entities and small unit sizes during indexing in order to narrow down the search space and increase the number of relevant matches. The module also leverages the side effects of employing the small local context of a retrieval unit for implicitly disambiguating words for a query extension component.
- Development of a strategy-based Answer Extraction module with proximity as a key concept for approximating the semantic relationship between words for factoid questions and lexico-syntactic patterns as a method of extracting answers to definition questions.
- Development of a Cross-Lingual Question Answering system that outperforms state-of-the-art systems.

1.3 Outline of this Thesis

The remaining chapters of this thesis are organized as follows:

- Chapter 2. This chapter briefly presents two techniques related to Question Answering, namely Information Retrieval and Information Extraction, techniques that are part of most architectures for Question Answering.
- Chapter 3. This chapter gives a short presentation of Question Answering systems' architecture and reviews the state-of-the-art research in both mono-lingual and cross-lingual Question Answering.

- Chapter 4. This chapter outlines the evaluation methodology for Question Answering systems and presents two ways of evaluating performance: component-wise and of the whole system.
- Chapter 5. This chapter introduces an open-domain mono-lingual Question Answering system for German and ways of extending it to answer English questions by integrating two competitive techniques of question translation. The evaluation methodology is outlined by briefly reviewing the test collection and effectiveness metrics used and presenting details of evaluating each component (glass box evaluation).
- Chapter 6. This chapter describes the Question Analysis modules for both German and English, with development and implementation details for the latter. The results of the empirical evaluation are presented and a brief error-analysis is provided.
- Chapter 7. This chapter discusses the preprocessing of the document collection, which can improve unit retrieval for factoid questions. Changes to the search engine in terms of indexing unit and weighing schemes are shown and methods of query expansion are introduced. Performance of the cross-lingual system with different combinations of methods for question translation, query expansion and indexing is evaluated and the best results analyzed.
- Chapter 8. This chapter introduces different strategies of extracting candidate answers for factoid and definition questions. While for definition questions the system relies on linguistic knowledge in the form of syntactic patterns for pinpointing possible answers, for factoid questions proximity to question keywords is considered. For both types of questions, selection of correct answers is based on the hypothesis that redundancy of data is a good indicator for its suitability.
- Chapter 9. This chapter makes some concluding remarks about the proposed methods for cross-lingual Question Answering, identifies the primary contributions of this thesis, and proposes promising avenues for future investigation into this problem.

2 Techniques related to Question Answering

Question Answering is building on the outcomes of its precursors for intelligent information access, namely *Information Retrieval* and *Information Extraction*. While they have not delivered the results comparable to a human expert, these techniques have paved the way for more advanced information processing and are therefore an integral part of the Question Answering systems.

2.1 Information Retrieval

Information Retrieval (IR) is concerned with retrieving from a large document collection those parts that are in some way relevant to a given query. IR is closely related to Question Answering, as we previously mentioned, since QA systems generally make use of information retrieval engines in order to narrow down the number of documents to be searched and processed in order to find a correct answer to a question.

An IR engine takes as input a query expressed in the engine's query syntax, which can be as simple as a "bag of words" or as complicated as phrases, sets of synonyms and keywords in strict order over windows of text. As output, an IR engine provides a ranked list of documents drawn from the collection it has previously indexed, documents that are considered relevant to the information need by some matching strategies.

Set Theoretic Models

A number of different approaches to the IR problem have been developed that fall broadly into two categories: Boolean and ranked retrieval. Early IR systems were Boolean systems which allowed users to specify their information need using a complex combination of Boolean AND, OR and NOT operators. Boolean systems have several shortcomings, e.g., there is no inherent notion of document ranking, and it is very hard for a user to form a good search request. Even though Boolean systems

usually return matching documents in some order, e.g., ordered by date, or some other document feature, relevance ranking is often not critical in a Boolean system. Even though it has been shown by the research community that Boolean systems are less effective than ranked retrieval systems, many power users still use Boolean systems as they feel more in control of the retrieval process.

The *fuzzy-set model* is based on fuzzy-set theory, which allows partial membership in a set, as compared with conventional set theory, which does not. It redefines logical operators appropriately to include partial set membership, and processes user queries in a manner similar to the case of the Boolean model. Nevertheless, IR systems based on the fuzzy-set model have proved nearly as incapable of discriminating among the retrieved output as systems based on the Boolean model. The strict Boolean and fuzzy-set models are preferable to other models in terms of computational requirements, which are low in terms of both the disk space required for storing document representations and the algorithmic complexity of indexing and computing query-document similarities.

However, most everyday users of IR systems expect the systems to do ranked retrieval. IR systems rank documents by estimating the relevance of a document for a user query. Most IR systems assign a numeric score to every document and rank documents by this score. Several models have been used for QA-embedded IR systems. Some of the most applied models are the vector space model and the probabilistic models.

Vector Space Model

The *vector space model* (VSM) of information retrieval, first introduced by Gerard Salton (Salton et al., 1975), models both the documents in the collection and the query strings as vectors in a finite dimensional Euclidean vector space. The space has one dimension for each of the terms in the language, with the entry for a given term being the weight given to that term for the document considered (0 if the term is not present in the document). The similarity factor for a given document is calculated as the scalar product of the vectors representing the query and the document.

There are different weighing schemes that can be used within the vector space model. The most common term-weighing approach for a VSM is known as the TF-IDF approach, which stands for term frequency - inverse document frequency.

Term frequency refers to the number of times a term appears within a document. The inverse document frequency of a term is the degree of how rare the term is across the entire corpus. The idea is that if a term occurs frequently in a document, but not frequently in the whole corpus, then that term has a high probability of semantically characterizing that document. In TF-IDF weighting, each term is weighted by the product of its term frequency and its inverse document frequency. It is usual to normalize the term weights against document length to avoid preferentially retrieving very long documents, which contain more terms and therefore have higher term frequencies for those terms than shorter documents have.

Once the document and query vectors have been constructed, there are several ways to calculate the similarity factor. One of the best known is the cosine measure, which assumes that terms occur independently of each other. Relating user queries to similar documents in the corpus is equivalent to computing the cosine of the angle between the query vector and the projections of document vectors onto the hyperplane containing the query vector. The standard VSM can be described as follows.

Basic assumption:

- All terms (the word which can be used as a keyword) are set as $k_1,...,k_t$.
- Express the arbitrary document D as an n-dimensional vector

$\vec{d} = (w_{1d},...,w_{nd})^T$, where w_i is the weight of term k_i in document D.

- Express the question of user Q as an n-dimensional vector

$\vec{q} = (w_{1q},...,w_{nq})^T$ as well.

- How close the value of similarity between the question Q and each document D is the cosine of the angle made by the two vectors (Formula 2.1).

$$sim(Q,D) = sim(\vec{q},\vec{d}) = \frac{\vec{q} * \vec{d}}{|\vec{q}| * |\vec{d}|} \quad (2.1)$$

The keyword (term) weight w_{ij} can be computed using the TF-IDF model as follows:

$$w_{ij} = tf_{ij} * idf_i \quad (2.2)$$

$$tf_{ij} = \frac{freq(k_i, D)}{\max_{l=1,n}(freq(k_l, D))} \quad (2.3)$$

$$idf_i = \log \frac{N}{df_i} + 1 \quad (2.4)$$

where, tf_{ij} is the normalized term frequency, idf_i is the inverse document frequency, N is the number of documents, df_i is the number of documents containing term k_i. Additionally, the performance of this kind of retrieval algorithm can be improved by filtering out stop words, which are functional words such as articles and prepositions so frequent in the entire corpus that their presence in a document does not contribute to the document's relevance to the query.

2.2 Information Extraction

Information Extraction (IE) is an established technology enabling relevant content to be extracted from textual information available electronically. IE essentially builds on natural language processing and computational linguistics, but it is also closely related to the well-established area of information retrieval and it is as a method of searching for information in some ways similar to Question Answering. The IE community devised its own evaluation exercise, the Message Understanding Conferences (MUC), which ran between 1987 and 1998, the last MUC-7 was held in 1998. The termination of the MUC exercises, coupled with the desire to push language understanding technology in novel directions via open evaluation exercises, were enabling conditions for the TREC question answering evaluation. Generally, the process of IE has two major parts. First, the system extracts individual "facts" from the text of a document through local text analysis. Second, it integrates these facts, producing larger facts or new facts (through inference). As a final step after the facts are integrated, the pertinent facts are translated into the required output format.

Template Matching

Previously known as message understanding, the overall goal of information extraction is to uncover information in free text that matches given templates. Templates are as diverse as representing events, references to entities, business deals, or anything else of interest to the user. Each template contains a number of slots that

the Information Extraction system wants to fill. For example, a user requirement for information about car accidents might use a template made of fields such as “Number of injured”, “Number of cars involved”, “Names of victims”, “Location”. The Information Extraction engine would then attempt to fill these fields as if entering the information in a database. When an Information Extraction system finds some text matching one of its templates, it uses as much context as it can to fill out all of the slots in the template.

There are, however, a number of limitations to template filling. Templates are usually hand-crafted by human experts to suit a particular domain and therefore cannot be easily ported to a new domain. The need to customize templates for the needs of a new domain can be considered a sub-problem of the more general issue of suiting a generic IE system for the needs of a particular user. In this sense, different approaches exist that induce patterns from positive training examples and user input. On the other hand, Question Answering improves on Information Extraction through templates and is much more in line with the idea of user-driven Information Extraction, allowing users to specify exactly what they want the extraction machine to provide.

Syntactic Structure Identification

Identifying some features of syntactic structure simplifies the task of information extraction. Often the arguments to be extracted match noun phrases in the text, and the relationships to be extracted correspond to grammatical relations. But identifying the full syntactic structure of a sentence is a challenging task. Therefore, there is a great variation in the amount of syntactic structure that is explicitly identified.

Some systems don’t have any separate phase of syntactic analysis. Others attempt to build a complete parse of a sentence. Most systems fall in between and build a series of parse fragments. In general, they only build structures about which they can be quite certain, either from syntactic or semantic evidence.

Named Entity Recognition

Named Entity (NE) Recognition is a specific form of the Information Extraction task that targets the identification of phrases in text referring to entities like people, organizations, dates and currency amounts, and extracting their semantics. Names appear frequently in many types of texts, and identifying and classifying them

simplifies further processing; names, furthermore, are important as argument values for many extraction tasks. Names are identified by a set of patterns (regular expressions) that are stated in terms of parts-of-speech, syntactic features, and orthographic features (e.g. capitalization). However, it is not enough for an NE recognizer to be able to identify that the phrase "President John Bush" refers to a person; the system must be able to fill out a template of information, such as that the person is male, his first name is "John", his last name is "Bush" and his function is "President". Examples of NE extraction systems include the LingPipe (Alias-I, 2003), GATE (Cunningham, H. at al., 2002) and the Stanford Named Entity Recognizer (Finkel, J. R. et al., 2005).

Among the limitations of IE systems is the fact that the templates have to be hand-crafted by humans, an effort that can be costly and usually not transferable across domains. However, a database can be created from a large body of text with information about different types of events or entity references, and if combined with modern natural language database interfaces can make a kind of narrow-domain QA system. Similar to natural language database front-ends, IE systems are limited in the types of questions they can answer by the structure of their database templates. Just as with IR engines, however, a good IE system can be an enormously useful resource for a high-quality QA system. IE can assist with question analysis, helping the system understand what type of entity it is looking for, and also with answer extraction, identifying entity references of the desired type among passages retrieved by upstream passage analysis and document retrieval modules.

3 State of the Art for Question Answering

Several workshops and evaluation forums - such as TREC, CLEF and NTCIR - are part of a surge in research approaches and systems that are being developed for Question Answering. These systems cover a wide scope of different methods and architectures, such as question type ontology, external databases of world knowledge, heuristics for extracting answers of certain types, reasoning through inference rules, feedback loops, generation of answers, machine translation, machine learning and even logical analysis, so that it is nearly impossible to capture all within a single architecture. However, the systems developed share a common pipeline architecture (Figure 1) that combines three essential modules in a sequential manner: *question analysis*, *information retrieval*, and *answer extraction and selection*.

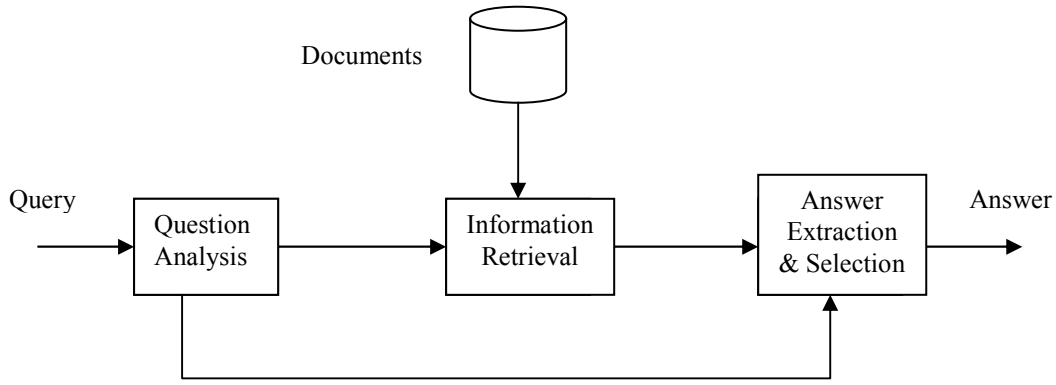


Figure 1. A generic monolingual QA System architecture.

The *Question Analysis* module processes the questions (e.g. part-of-speech tagging, named entity extraction, parsing), and both analyzes and classifies them according to different ontologies. At this stage, information related to the question's semantics and expected answer type is extracted, which triggers different strategies of retrieval and answer extraction later on.

The *Information Retrieval* module generates queries according to question types, keywords, and additional content. Based on these queries, relevant documents expected to contain correct answers are retrieved.

The *Answer Extraction and Selection* module extracts candidate answers from relevant documents and assigns them a confidence score, and then selects the most probable answer as correct based on notions of overlap and similarity.

In the field of cross-lingual Question Answering, crossing the language barrier between the question and the document collection can be done at two stages: before the Question Analysis by translating the question and before the Information Retrieval by translating the documents. Systems built for the cross-lingual scenario of Question Answering leverage existing monolingual architectures and use an additional module of translation (Figure 2).

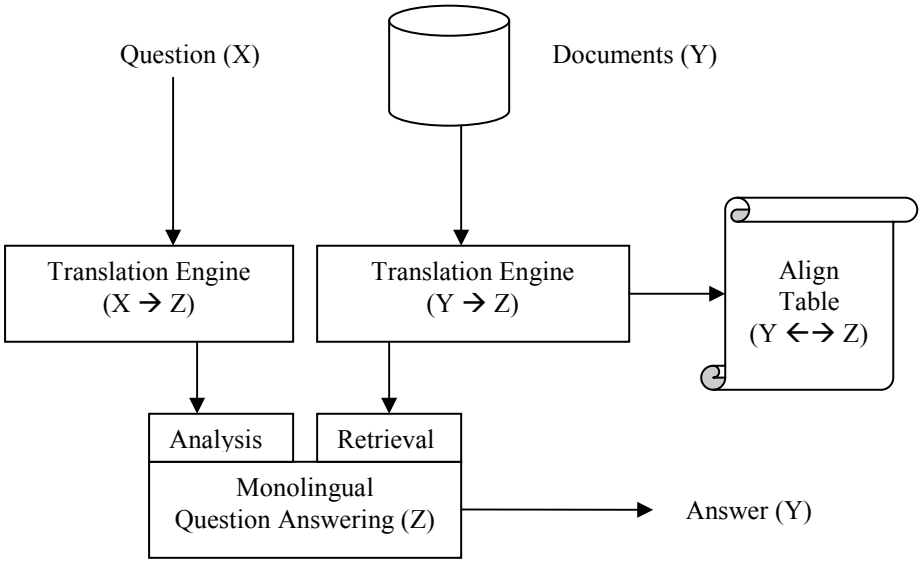


Figure 2. A generic cross-lingual QA System architecture

Depending on the target language of the translation component, we can distinguish three methods of cross-linguality:

- by *Question Translation*, where the question is translated into the language of the document collection (Z = Y),
- by *Document Translation*, where the documents are translated into the language of the information request (Z = X),

- by using an *interlingua* common subject representation, where both the question and the document collection are translated into a third language (Z != X and Z != Y), either a natural language or a formal representation.

From this point on, the cross-lingual QA system deals with a single language (Z) and can leverage the power of existing monolingual Question Answering modules. A by-product of translating the document collection is an alignment table of the original documents to the resulting translations (Y ←→ Z) that allows giving the correct answer in its original context (Y).

3.1 Monolingual QA

3.1.1 Question Analysis

In order to understand what the question asks for, an important step for extracting the exact answer is to detect the semantic type of the question. Placing the questions into several semantic categories imposes some constraints on the possible answers and suggests at the same time potential different processing strategies.

Question categorization can be approached in different ways that can be either rule-based or learned methods. Of the rule-based approaches, one of the simplest, and yet quite effective, ways is to apply pattern matching to the question to identify its type (Monz et al., 2001). Hermjakob, U. (2001) also fully parses questions and then applies a large number of rules to the parse tree to classify questions. Another method is a heuristic rule-based algorithm, which requires writing some heuristic rules manually for question classification, although it is tremendous amount of tedious work (Radev et al., 2002; Molla, D. & Gardiner, M., 2004). As an alternative to pattern matching, there are much more sophisticated means for question classification based on machine learning (Suzuki et al., 2003). Zhang, D. & Lee, W. (2003) use support vector machines, a machine learning approach to question classification. Li, W. (2002) uses language models for question classification. Li, X. & Roth, D. (2002) make use of a multi-class learning with a Sparse Network of Winnows (SNoW) and a two-layer class hierarchy. Metzler, D. & Croft, W. B. (2005) use prior knowledge about correlations between question words and types to train word-specific question classifiers. Nguyen, M. L. et al. (2007) propose a subtree mining method for question classification and use a maximum entropy and boosting model with subtree features.

Li, F. et al. (2008) formulate the classification task as a word sequence tagging problem and use Conditional Random Fields classifiers to tag features of question words to include both syntactic and semantic information. Mikhailian, A. et al. (2009) propose another model based on word tagging, using for each word features like strings and PoS-tags on a 4-word window, as well as the WH-word, the parsed subject of the question and the first nominal phrase.

Once the type of entity being sought has been identified, the remaining task of question analysis is to identify additional constraints that entities matching the type description must also meet. This process may be as simple as extracting keywords from the rest of the question to be used in matching against candidate answer-bearing sentences. This set of keywords may then be expanded using synonyms and/or morphological variants (Srihari, R. & Li,W. , 2000) or using full-blown query expansion techniques by issuing a query based on the keywords against an encyclopedia and using top ranked retrieved passages to expand the keyword set (Ittycheriah, A. et al., 2001). Harabagiu, S. & Lacatusu, F. (2004) use FAQ data to learn by way of bootstraped information extraction how to expand query terms by answer terms. Riezler, S. et al. (2007) use the same type of question-answer pairs to train an end-to-end phrase-based monolingual SMT (statistical machine translation) model that learns correlations between words and phrases in questions and answers. Tellez, A. et al. (2007) mine association rules that represent pairs of highly related concepts from the document collection and use them for extending the initial query. Bernhard, D. & Gurevych, I. (2009) propose new kinds of datasets for training monolingual SMT models by combining different lexical semantic resources such as WordNet, Wikipedia and Wiktionary.

More advanced approaches to constraint identification, like Harabagiu, S. et al. (2000), use a wide-coverage statistical parser that aims to produce full parses from which dependencies between terms of the question are captured. Scott, S. & Gaizauskas, R. (2001) use a robust partial parser to determine grammatical relations that hold between the sought after entity and terms in the question. Hartrumpf, S. (2005) uses a complete sentence parse to build a semantic network of the MultiNet formalism for the question that has to be matched by the semantic network of a document containing the correct answer. Bos, J. & Nissim, M. (2007) build a semantic representation of the question in the form of a Discourse Representation

Structure (DRS) that delivers further background knowledge for finding appropriate answers.

Our approach: For Question Analysis, we use a full parse of the question in order to classify the question into predefined categories by applying rules. At the same time, we extract the keywords and salient information as focus from the question and extend these with lexical- and conceptual-related items. No disambiguation is attempted at this point, rather we rely on the small retrieval units for naturally selecting the appropriate meanings of collocating words.

3.1.2 Information Retrieval

The function of the retrieval component is not to find actual answers to the question, but to identify textual units that are probable to contain an answer. Several aspects are to be considered at this stage: the retrieval model, the size of the retrieval unit and the ranking methodology.

First, one must decide whether one wants to use a Boolean, a ranked answer or a probabilistic search engine. Despite the higher results of ranked answer engines in standard IR evaluation, some researchers have argued that Boolean engines are more suitable for use in conjunction with a QA system (Moldovan, D. et al., 2000; Gaizauskas, R. et al., 2003). Both the Boolean and the ranked answer approaches assume that the terms being used for retrieval are independent of each other and existing term relationships need not be taken into account. To overcome this problem, probabilistic models in the form of language models have been considered for retrieval, as well (Corrada-Emmanuel, A. et al., 2003; Merkel, A. & Klakow, D., 2007; Bernhard, D. & Gurevych, I., 2009).

Second, the search engine may allow retrieval of textual units smaller than documents, and various parameters need to be set therefore (passage length, passage windowing interval). Clarke, C. et al. (2000) present and evaluate an algorithm for passage selection in the context of question answering and Hovy, E. et al. (2001) experiment with how fine-grained the process of segmentation can be. Roberts, I. (2002) compares the performance of document vs. passage retrieval for question answering and concludes that using passages of two paragraphs length is better than using the whole document. Tiedemann, J., & Mur, J. (2008) investigates several ways of dividing documents into passages considering semantically motivated approaches

using co-reference chains and discourse clues against simple fixed-size window-based techniques. The results show the somehow surprising outcome that the simple techniques outperform the semantically motivated approaches. Related research performed by Khalid, M., & Verberne, S. (2008) show the effectiveness of sliding fixed-size windows compared to disjoint windows. Lao, N. et al. (2008) experiment with three retrieval units (document, block and sentence) and report their best results with a combination of document retrieval plus sentence/clause extraction.

Once relevant documents or passages have been selected, these textual units may then be further processed: sentence split, part-of-speech tagged, and chunk parsed. In order to establish an explicit link between a phrase of the expected answer type and the question, several methods can be used: linear proximity approaches, parsing of the syntactic and semantic structure, pattern matching, or textual entailment.

Research has shown that taking into account the proximity between question terms is helpful in determining whether a document contains an answer to a question (Clarke, C. et al., 2000; Kwok, K. et al., 2000). Monz, C. (2004) proposes a novel proximity-based approach to document retrieval called minimal span weighting that leads to significant improvements when compared to state-of-the-art document retrieval approaches.

Several QA systems have attempted to use syntactic information, and especially dependency relations, for this task. One approach is to look for an exact match between dependency tuples derived from the question and those present in a potential relevant document (Harabagiu, S. et al., 2000; Katz, B. & Lin, J., 2003; Litkowski, K. 2004). Punyakanok, V. et al. (2004) compute the tree edit distance between the dependency trees of the question and answer-bearing passages, and select answers from sentences which minimize this distance. Mollá, D. & Gardiner, M. (2004) compute the match between question and answer-bearing passage using a metric, which basically computes the overlap in dependency relations between the two. Verberne, S. et al. (2008) use a paragraph retrieval extended with a re-ranking module based on structural linguistic and lexical information. Yet other QA systems use syntactic and semantic analysis to represent relevant documents as a logical form prior to answer extraction (Molla, D. et al., 1998; Zajac, R., 2001; Moldovan, D. et al., 2003; Glöckner, I. & Pelzer, B., 2008).

Otherwise, pattern matching is an intuitive and effective means to associate a passage to the question. Soubbotin, M. & Soubbotin, S. (2001) apply pattern matching

to question answering with good results, and Ravichandran, D. & Hovy, E. (2002) automatically learn answer matching patterns with only a small number of training examples, while Shima, H. & Mitamura, T. (2010) use a minimally supervised bootstrapping approach to generating lexico-syntactic patterns for answer extraction. Cui, H. et al. (2007) propose the use of probabilistic patterns, called soft patterns, for definitional question answering in the TREC contest. Soft patterns generalize over lexico-syntactic "hard" (fixed) patterns in that they allow a partial matching by calculating a generative degree of match probability between the test instance and the set of training instances.

A thorough look at the application of textual entailment to Question Answering is presented by Harabagiu, S. & Hickl, A. (2006), who filter and re-rank the text fragments containing the answer candidates based on the entailment confidence assigned by the entailment engine. Another machine learning approach to the so-called Answer Validation is proposed by Wang, R. & Neumann, G. (2007, 2008), who extract parts of the dependency structures to form a new representation, named Tree Skeleton, and then apply Subsequence Kernels to learn an entailment engine. Celikyilmaz, A. et al. (2009) present a graph-based semi-supervised learning for ranking candidate sentences by exploiting unlabeled entailment relations based on a combination of syntactic and semantic features.

Our approach: For Information Retrieval we use a search engine that integrates both a Boolean and a ranked model into its scoring scheme. We experiment with different sizes of sliding windows as retrieval units (1-sentence, 3-sentences and 5-sentences), and use an integrated approach of linear proximity with semantic structure parsing (in form of expected answer type) for matching potential relevant contexts.

3.1.3 Answer Extraction and Selection

In the component of answer selection, the subject representations of the question and of the relevant textual units are matched against each other, resulting in a set of candidate answers ranked according to likelihood of correctness. Typically, systems that analyzed both an expected answer type and some additional constraints on the input question will have also the candidate passages analyzed, at least with annotations of the answer type set.

A variety of ways to extract and select candidate answers exist, ranging from simple named entity annotation to machine learning approaches. Initial experiments focused on answer types of named entities as used in Abney et al. (2000). Ittycheriah, A. et al. (2001) factor both expected answer type matching and a range of sentence, entity and linguistic features into a single scoring function that they apply to a three sentence window sliding over relevant textual units. Light et al. (2000) provide empirical evidence of upper bounds on word-based comparison approaches. Mollá, D. & Gardiner, M. (2004) combine the use of named entities with that of logical form patterns.

Moldovan, D. et al. (2000) compute an overall score for the word overlap between the question and the answer window by means of weighted numerical heuristics. Harabagiu, S. et al. (2000) extend this approach by using a machine learning algorithm to optimize the weights in a linear scoring function that subsumes features typical to the answer windows.

Nyberg, E. et al. (2003) train support vector machines, K-nearest neighbor and decision tree classifiers to assess the likelihood of individual answers. Echihabi, A. et al. (2003) use three separate answer extraction agents and combine the output scores with a maximum entropy re-ranker.

Pinchak, C. & Lin, D. (2006) break down the question into a number of possibly overlapping contexts (dependency tree paths involving the *wh*-word) and evaluate a candidate answer as to how likely it is to appear in these contexts, in place of the *wh*-word. Pinchak, C. et al. (2009) present a flexible approach based on discriminative preference ranking to determine which of a set of candidate answers are appropriate. Surdeanu, M. et al. (2008) explore preference ranking for complex-answer questions (*how to*) in which a unique correct answer is preferred over all other candidates.

Most systems employ a large variety of specific resources such as dictionaries, encyclopedias and gazetteers, as well as online semi-structured sources (Lita et al., 2004; Jijkoun, V. & de Rijke, M., 2004; Buscaldi, D & Rosso, P., 2006; Lopez, V. et al., 2010). These external, either offline or online, resources are best suited for definition questions as shown in Lin, J. and Katz, B. (2003). Further methods for answering definition questions include heuristics as definition patterns (Joho, H. & Sanderson, M., 2000), lexico-syntactic patterns (Xu, J. et al., 2005; Cui, H. et al., 2007), and making use of Wikipedia's article structure to extract explanations of key terms (Tellez, A. et al., 2007).

Using frequencies to select an answer, also known as redundancy-based answer selection, is also used in the research of Clarke et al. (2002) or Dumais et al. (2002). Some other research even goes beyond the referenced document collection and use the World Wide Web to get these frequencies (Magnini et al., 2002; Saias, J. & Quaresma, P., 2008). Newer research of Lee, Y.-H. et al. (2008) uses the concept of entropy from the information theory, which is similar to the inverse document frequency (IDF), to narrow down the number of relevant answer candidates.

Our approach: For Answer Extraction we use a redundancy-based approach for those named entities corresponding to the expected answer type, ranked by their normalized distribution over relevant sentences and documents. The Answer Selection uses a new proximity measure that approximates the syntactic relationship between words in a local context. For definition questions we employ a set of offline resources built from instantiations of manually defined lexico-syntactic patterns.

3.2 Cross-lingual QA

Question Answering is an active field of research not only in one language, English, but also in other languages. The Cross-Language Evaluation Forum (CLEF) is a forum where cross-language question answering systems are evaluated for a variety of European languages. More recently a series of workshops known as NTCIR for Asian languages like Chinese, Japanese, and Korean have offered a test bed for cross-lingual question answering as well. These workshops have become increasingly important since they are fostering research and development of question answering systems for languages other than English and across several languages.

Most of the research for Question Answering in crossing the language barrier between the question and the document collection, when each is expressed in a different language, applies methods and results known from cross-lingual Information Retrieval (CLIR). One of the basic modules in the design of Question Answering systems, Information Retrieval deals directly with both elements of the cross-lingual problem: the question, or a subject representation of it, as the information need and the documents, or possible answer-bearing units, as the pool of available information. Therefore it is the most intuitive way of approaching the cross-linguality in Question Answering systems at this level first.

As previously mentioned, three methods of bridging the difference in language between the question and the documents are popular:

- by translating the question into the language of the document collection,
- by translating the documents into the language of the input question, or
- by translating both the question and the language into a third intermediate language, called interlingua, whether the language is a natural or an artificial (formal representation) one.

In general, the translation quality is degraded by two factors:

- translation ambiguity (multiple translations with different meanings for a single source term)
- out-of-vocabulary (OOV) problem (multi-term concepts)

While the second mentioned factor is mostly addressed by use of external resources, specially designed to deal with coverage issues, the first one can be managed by exploiting techniques known as word sense disambiguation (WSD) based on contextual information. Therefore, it is expected that translation of documents, where context of ambiguous terms is larger, yields better results than translation of questions, where context is hardly present. Nevertheless, when the document collection is very large, the cost of translating it completely into another language becomes prohibitive.

The most tractable and therefore most frequently used method for crossing the language barrier is by translating the question into the language of the document collection. This can be achieved by using either automatic machine translation (MT) or machine readable dictionaries (MRD). While each of these approaches comes with its pros and cons, MT prevails for the most part in actual research. The reason for this popularity is twofold: on one side, it inherently addresses the above-mentioned problem of ambiguity, by generating at its best one single translation, based on the context available. On the other side, it lies in its ability to preserve to a fair degree the structure (syntax) of the question being translated, which is a capital factor of success for the Question Analysis and therefore for the entire Question Answering system.

We remember at this point that the Question Analysis component computes both the question and expected answer types, both of which determine the progress downstream toward extracting the correct answer. Failure at this stage of the workflow hinders the systems in delivering the right answer, regardless of how accurate and precise the subsequent components are.

By using an MT-based question translation approach to cross-lingual Question Answering minimal changes are required in order to adapt an existing monolingual system to a new source language, changes that usually consist in integrating external machine translation services. Echihabi, A. et al. (2003) report on using different techniques for question translation, of which off-the-shelf rule-based machine translation (SysTran[1]) performed better than statistical machine translation and a bilingual table lookup. Lita, V. et al. (2003) combine an off-the-shelf MT system (SysTran) to translate the question with a statistical translation model, then they retrieve the relevant cross-lingual documents, and subsequently leverage the performance of a pattern-based monolingual system.

However, when existing machine translation services are not delivering the necessary quality to accurately extract information about the semantic type of the question and further constraints imposed on its subjects and expected answers, development of a Question Analysis component for the original question becomes mandatory. This implies availability of additional natural language tools, like part-of-speech taggers and grammatical structure parsers, for the source language, beside the effort of developing a new analysis component for every language to be considered.

Sutcliffe, R. et al. (2003) use free-available online machine translation services (Google Translation) to perform cross-lingual Question Answering from French to English by first analyzing the original French question to identify its type and then translating it into the target language. Plamondon, L. & Foster, G. (2003) apply a set of manually written rules for analyzing the original French questions, followed by an IBM1 statistical translation engine to get the keywords rendered into English. Neumann, G. & Sacaleanu, B. (2003) describe the combination of several machine translation services, both online and offline, to improve their coverage, after analyzing the original German question at an earlier stage. Lao, N. et al. (2009) have found that best performing is a combination of both question translation methods, by assembling

[1] www.systransoft.com

together key terms obtained from the analysis of the translated question with key terms translated from the original question's analysis.

In contrast to machine translation, MRD is used for those language pairs for which no MT tools are readily available or the quality of translation, either due to coverage problems or too complex questions, is not satisfactory failing to offer a real advantage over bilingual dictionaries. Since determining correct question and expected answer types is crucial for the overall performance of a Question Answering system and translating the question by use of MRD addresses issues neither of grammatical structure nor of translation ambiguity in the target language, both a source Question Analysis component and methods or heuristics of target disambiguation are required.

A question answering system developed by Negri, M. et al. (2003) employs both bilingual Italian-English dictionaries and the MultiWordNet thesaurus to translate word-by-word the result of the question analysis performed on the original inquiry, overcoming ambiguity difficulties by means of statistical techniques. Bourdil, G. et al. (2004) make use of bilingual French-English dictionaries to perform translation of both uni-terms and bi-terms resulted from parsing the original French question with no attempt for disambiguation, but selection being made during the retrieval process. Ferrández, S. et al. (2009) use several multilingual knowledge resources to reference words between languages, considering more than one translation per word to search candidate answers. The resources used are the Inter Lingual Index (ILI) module of EuroWordNet and the multilingual knowledge encoded in Wikipedia. Ren, H. et al. (2010) employ an online English-Chinese dictionary as an alternative to translation engines to obtain results close to a monolingual system, outperforming the machine translation approaches used in their experiments.

Though associated with high computational effort required to translate the entire collection of documents, crossing the language barrier from the target language of the documents to the source language of the question has been considered in development of some Question Answering systems, as well. This approach is appropriate when adapting an existing monolingual QA system, e.g. for English, to a new document collection of another language, e.g. Spanish. In this scenario all Spanish documents are translated into English, indexed by the Retrieval component, and passed over to the English monolingual QA system that can handle them. An alternative to translating the whole collection of documents is to only translate those documents that might be relevant to the question asked. Therefore the source question can be roughly

translated with some simple techniques like MRD and consequently used to pre-fetch a set of documents into the target language, e.g. German. These possible relevant documents, relatively small in number compared to the entire collection, are then translated into the source language, e.g. English, and dynamically indexed and searched for answers by the existing monolingual system. The result is afterwards extracted from the original documents, e.g. German, by using the word alignment by-product of the performed translation.

Shimizu, K. & Akiba, T. (2005) use statistical machine translation, trained on a bilingual English-Japanese corpus, and an existing Japanese QA system to answer Japanese questions from an English document collection by translating only pre-fetched question relevant documents. Bowden et al. (2006) report comparable results with no significant difference in recall for both approaches of translating the entire document collection and only pre-fetched documents with a phrase-based statistical machine translation engine. Min, J. et al. (2010) compare query translation by using Google's online service with whole document collection translation by using a proprietary statistical machine translation tool and report better results for the first method.

The use of an interlingua representation for bridging the language difference between the question and the document collection in Question Answering has not been approached up to now. Beside the advantage that it could deliver, that of reducing the amount of work required to traverse the gap between any two languages, this approach assumes high costs through the amount of analysis required to map natural language utterances into a common representation without losing the semantic, style and emphasis of the original. A rather similar but more tractable approach is that of translating the index of the underlying IR component from the source into the target language of the question. Akiba, T. et al. (2008) present an English-Japanese QA system that uses the word translation probability from a statistical machine translation to index the Japanese documents with the corresponding English terms without losing the consistency. The passage similarity calculation subsystem computes the match between an English question and a Japanese passage in terms of the probability that the Japanese passage is translated into the English question.

Our approach: For crossing the language barrier on the question side, we provide two approaches: direct question translation and a so-called transfer-based translation. The first approach analyzes the question after first translating it by way of automatic machine translation. The second approach interprets the question in a first step and then it uses two different techniques of translating the resulting interpretation: by using machine readable dictionaries on one side and translation alignment lists from the direct translation on the other side. The first technique has no attempt for disambiguation, but builds again on the assumption that small retrieval units naturally select the appropriate meanings of collocating words.

4 Evaluation Methodology

Evaluation is a key element to making progress in developing better Question Answering systems, by serving two goals:

- to obtain information that can inform the ongoing design and development process (often referred to as descriptive evaluation);
- to decide whether an innovation is worth retaining (often referred to as analytic evaluation).

Systems often distinguish between so-called *glass box* and *black box* evaluations, which differentiate between component-wise versus whole-system evaluation. The *glass box* evaluation is a descriptive approach answering the question "***How*** does it do what it does?" while the *black box* evaluation is an analytic approach answering the question, "How ***well*** does it do what it does?" In terms of data being processed by the Question Answering system the *black box* evaluation considers only system input-output relations without regard to the specific mechanisms by which the outputs were obtained, while the *glass box* evaluation examines the mechanisms linking input and output.

Component-wise experiments can offer a better idea of the Question Answering process, can uncover what has happened and why – what is and what is not working - and may provide feedback for better design and development choices.

Whole-system experiments, on the other side, evaluate performance of the summative Question Answering process, based on the choices made for the best component design.

In the following subsections we will describe the evaluation methodology for individual QA components and also for the whole system. Before we explore in detail the QA system, in order to understand how each component is evaluated, we first provide the necessary background into evaluation corpus, effectiveness metrics, and

component-wise (question analysis, document retrieval, answer extraction and selection) criteria of success.

4.1.1 CLEF Evaluation Corpus

One of the basic requirements for evaluation is for the results to be comparable across different approaches. Therefore both experimental settings and data must be fixed in order to make sure that experiments are repeatable. To measure effectiveness of a Question Answering system in a consistent way, a test collection consisting of the following things is needed:

- a document collection,
- a test suite of natural language questions, and
- a set of question-answer pairs along with the answer-bearing ID or snippet of a document.

An evaluation corpus for both monolingual and cross-lingual Question Answering has been assembled as part of the CLEF (Cross Language Evaluation Forum) initiative, whose goal is to promote Research and Development in multilingual information access:

- by developing an infrastructure for the evaluation of both monolingual and cross-lingual information retrieval systems for European languages, and
- by creating test collections of reusable data that can be employed by system developers for benchmarking purposes.

Initiated for information retrieval systems in year 2000 and based on the "Cranfield" IR evaluation methodology (Cleverdon, C., 1991), whose main focus is on experiment comparability and performance evaluation, the coverage of CLEF has been extended to question answering systems in year 2003, motivated

- by the interest in languages other than English for this research area, and
- in order to test the portability of the existent technology developed for English in the context of the TREC workshops.

The languages involved were Dutch, Italian and Spanish in the monolingual tasks and Dutch, French, German, Italian and Spanish source language queries to an English target document collection in the bilingual task.

The document collection addressed by the questions for the monolingual tasks were three collections of newspaper and news agency documents released in 1994 and 1995, and written in Dutch, Italian, and Spanish respectively. For the cross-lingual task the Los Angeles Times newspaper collection from the same time period was used. A test suite of 200 questions was compiled for each monolingual and cross-lingual scenario along a set of relevance judgments for each question-answer pair (Figure 3) and the test collection was released under the DISEQuA corpus (Magnini, B. et al., 2003).

Over the years both the question type and the document collection have evolved from factoid to definitions, to list, to linked questions and from news to Wikipedia documents (dump of the 2006 version). With the increase in number of participants each year, the proposed evaluation tasks have become more challenging and culminated in 43 activated language combinations for 11 different languages in 2008, of which not less than 33 were set in a cross-lingual scenario.

For the experimental part of this thesis, the test collection of CLEF for two consecutive years, 2007 and 2008, will be used focused on factoid and definition questions only. We have slightly modified these test collections such that no temporal restrictions on factoid questions are allowed and linked questions, implicitly referring to a common topic, were changed to a set of self-contained questions (Figure 4).

Also, the NIL questions, asking for facts whose answers could not be found in the document collection, were removed, resulting in a test collection of 346 factoid and definition questions with answers in the CLEF document collection or Wikipedia.

```
<qa cnt="20" type="MEASURE">
      <language val="ITA" original="TRUE">
            <question assessor="ALE">
                  Quanti abitanti ha Berlino?
            </question>
            <answer n="1" idx="SDA19940804.00147">
                  3,5 milioni
            </answer>
      </language>
      <language val="SPA" original="FALSE">
            <question assessor="Víctor">
                  ¿Cuántos habitantes tiene Berlín?
            </question>
            <answer n="1" idx="EFE19940107-02622">
                  Casi cuatro millones
            </answer>
      </language>
      <language val="DUT" original="FALSE">
            <question assessor="LIT">
                  Hoeveel inwoners heeft Berlijn?
            </question>
            <answer n="1" idx="NH19950601-0163">
                  3,5 miljoen
            </answer>
      </language>
                  <language val="ENG" original="FALSE">
            <question assessor="">
                  How many inhabitants are there in Berlin?
            </question>
            <answer n="1" idx="LA010194-0094">
                  SEARCH[3,500,000]
            </answer>
      </language>
</qa>
```

Figure 3. Format of question-answer pairs in DISEQuA corpus.

TOPIC: George W. Bush
Q1: Who is George W. Bush?
Q2: When was he born?
Q3: Who is his wife?

Self-contained Questions
Q1: Who is George W. Bush?
Q2: When was George W. Bush born?
Q3: Who is George W. Bush's wife?

Figure 4. Conversion of linked questions.

Evaluation Metrics

Question Answering builds on the Information Retrieval experience accumulated over decades of research as far as the performance measures are regarded. Basic measures like precision and recall or factorizations thereof (i.e. F-measure), as known from the IR area, either lose their significance by the nature of the Question Answering problem, where only one single correct answer may exist making the recall inappropriate, or have to be reconsidered to focus on a limited number of top documents for QA-embedded IR systems. Following are some of the most popular metrics that are used in QA-embedded IR and factoid question answering evaluations.

For the clarity of definitions we will consider R to be a rank-ordered vector of results $<r_1, r_2, \ldots, r_n>$ to the information need q and $rel(r_i)$ be 1 if result r_i is relevant to q and 0 otherwise.

- **Precision at rank k** – is the number of relevant units within the top k results for a given information need.

$$P_k(R) = \frac{1}{k}\sum_{i=1}^{k} rel(r_i)$$

This measure is typically used to compare results at the top of the ranking, since that is what many users care about.

- **Mean Average Precision (MAP) at rank k** – is the mean of the average precision (AP) at rank k for a set of information needs.

$$AP_k(R) = \frac{\sum_{i=1}^{k} P_i(R) \times rel(r_i)}{\sum_{i=1}^{k} rel(r_i)}$$

The average precision (AP) measure summarizes the ranking by averaging the precision values from the rank positions where a relevant unit occurred.

- **Mean Reciprocal Rank (MRR) at rank k** – is the average of the reciprocal rank (RR) at rank k for a set of information needs.

$$RR_k(R) = \begin{cases} \frac{1}{i}, & if \quad \exists\, i < k : rel(r_i) = 1 \wedge \forall\, j < i : rel(r_j) = 0 \\ 0, & otherwise \end{cases}$$

This measure has been used for application where there is typically a single relevant unit.

4.1.2 Component-wise Evaluation

Evaluation initiatives like CLEF are typically designed with the goal of testing the performance of whole Question Answering systems and target rather the outcome of the systems as a way of comparing their effectiveness. By doing this they are more action-oriented, considering only system input-output relations without regard to the specific mechanisms by which the outputs were obtained. A more research-oriented evaluation can be done by assessing the ongoing process of the Question Answering systems for the purpose of improving it through immediate feedback by using a component-based evaluation. This kind of evaluation is, at its most basic, an assessment of efforts prior to their completion for the purpose of understanding the mechanisms behind the systems and improve them on-the-go.

Given the modular architecture of our system we perform a component-wise evaluation in order to get insight into its functioning and show the relationship between the performance of individual components and the result of the system as a whole. We experiment with different methods and resources and evaluate performance for Question Analysis, Information Unit Retrieval and Answer Extraction and Selection.

Question Analysis

The Question Analysis component plays the critical role of extracting part of the question's semantic by identifying its type and the characteristics of the expected answer, of which its type is the most important. This information determines the workflow of components downstream and focuses them on finding a correct answer of that particular type.

We perform local experiments against the CLEF test collection (Gold Standard) to assess performance of both German and English question analysis components in finding out the right question and expected answer type. It is important in a cross-lingual setting to make sure that both components are comparable in their performance in order to evaluate the effects of different question translation techniques.

Information Unit Retrieval

Most of the Question Answering systems are built upon a search engine for the retrieval of information units. By doing this, the systems leverage well-known IR techniques to narrow the search space to a limited number of relevant documents. A relevant document in the context of automatically evaluating QA-embedded search engines is a textual unit containing the correct answer to a question, regardless of the answer being supported by its context or not. Evaluations based on this assumption are called lenient, in contrast to strict evaluations where the correct answer must be supported by its context. In our experiments we are going to assume a lenient evaluation against the CLEF test collection.

Three important aspects have been shown to have an effect on the retrieval performance of search engines:

- the index unit,
- the retrieval unit, and
- the query expansion.

Index units are structural units that represent the content of a document, and they are used for searching and consequently individually retrievable from queries. Non-functional words (i.e. nouns, verbs, adjectives and adverbs) are considered as basic index units for a typical search engine, but more complex structures can be used to retrieve more focused results. We will experiment with named entities as additional index units.

Retrieval unit is the type of object returned by a search engine as the response to a query and can range from whole documents to passages, sentences, and phrases. For some applications, like Question Answering, it can be useful to shrink the retrieval unit to the extent that it can still deliver correct answer without losing its

expressiveness. In our experiments we will compare two retrieval units: sentences and passages (as a window of adjacent sentences).

Query expansion is a technique used for enhancing performance of information retrieval that expands the set of search terms in a query by adding terms automatically selected from external knowledge resources. We compare synonym expansion with conceptual related words expansion in section 7.4.1.

Improved retrieval in question answering is critical so that further modules in the QA pipeline, especially answer extraction, have sufficient (redundant) textual units that contain correct answers appearing in various contexts. Therefore the more relevant units are retrieved by the IR component, the higher the answer recall should be – i.e. the more likely it is for the correct answer to be extracted and supported by different contexts.

Answer Extraction

The central component in a Question Answering system is the answer extraction. The goal of the extraction stage is to identify potential answers in running text and score them according to how likely they are to be correct. The running text consists of documents or passages that have been retrieved by the previous stage in the pipeline. The assumption is that at least part of the documents given to the extraction component is relevant – i.e. contain a correct answer.

We experiment with two different extractors: proximity extractor for factoid questions, and a pattern-based extractor for definition questions. We evaluate the extractors using the Mean Reciprocal Rank and Precision at rank k metrics. While both metrics offer an aggregate numeric score based on the several top answers, the Top K metric is more relevant for the extraction task.

5 Quantico: A Cross-language Question Answering System

In an ideal setting, a Question Answering system would understand the question being asked and retrieve the answer from its knowledge base within seconds. Having interpreted the question it would need only a look-up for the correct answer into the knowledge it has already acquired (Figure 5). This would presume existence of a language understanding module that could automatically extract concepts and their relationships from both the question and the documents and use them for matching the set of conceptual knowledge (patterned rectangle) to deliver the correct answer.

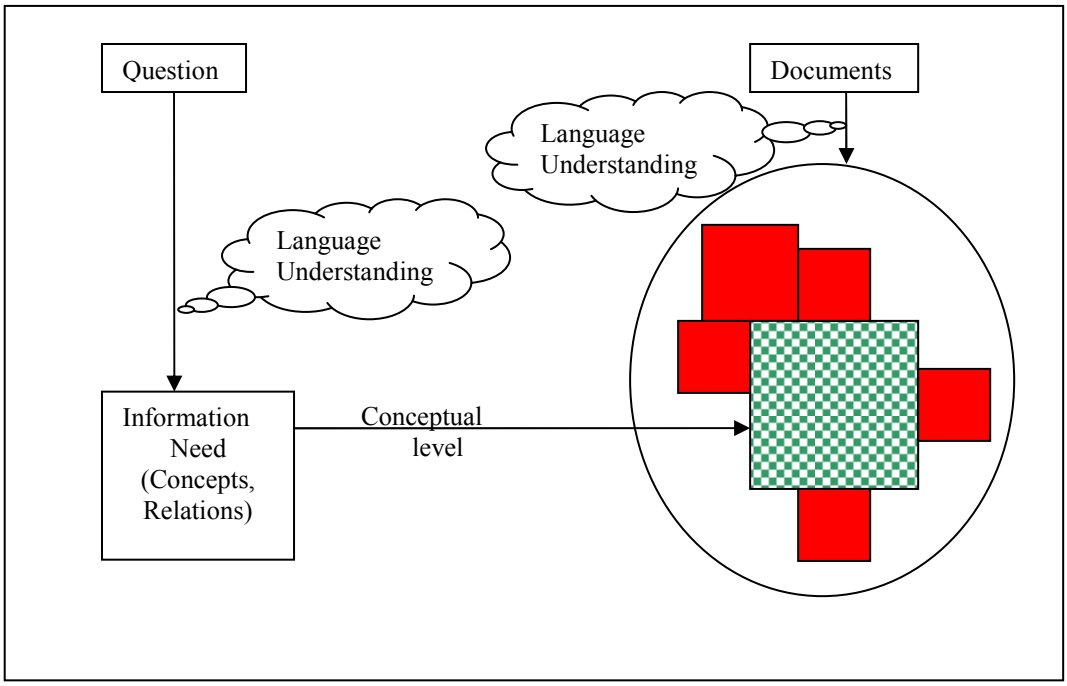


Figure 5. Ideal QA System.

Though this endeavor of understanding natural language might be worthwhile, it is also fraught with many difficulties, of which language ambiguity, both lexical and syntactic, is the biggest hindrance. Lexical ambiguity

is the characteristic of many words to have more than one meaning, of which the one that makes the most sense in a given context has to be selected. Syntactic ambiguity is a property of sentences that may be reasonably interpreted to mean more than one thing and arises from the relationship between the words and clauses of a sentence, and the sentence structure implied thereby. Failure to correctly disambiguate the natural language prevents automatic extraction of the intended meaning of an information request both in terms of concepts and their relationships, resulting in a match of the non-relevant knowledge (un-patterned rectangles) and therefore in poor performing information access technologies such as Question Answering. The lack of a language understanding module with good performance figures calls for alternatives that strive to offer good approximations for this functionality.

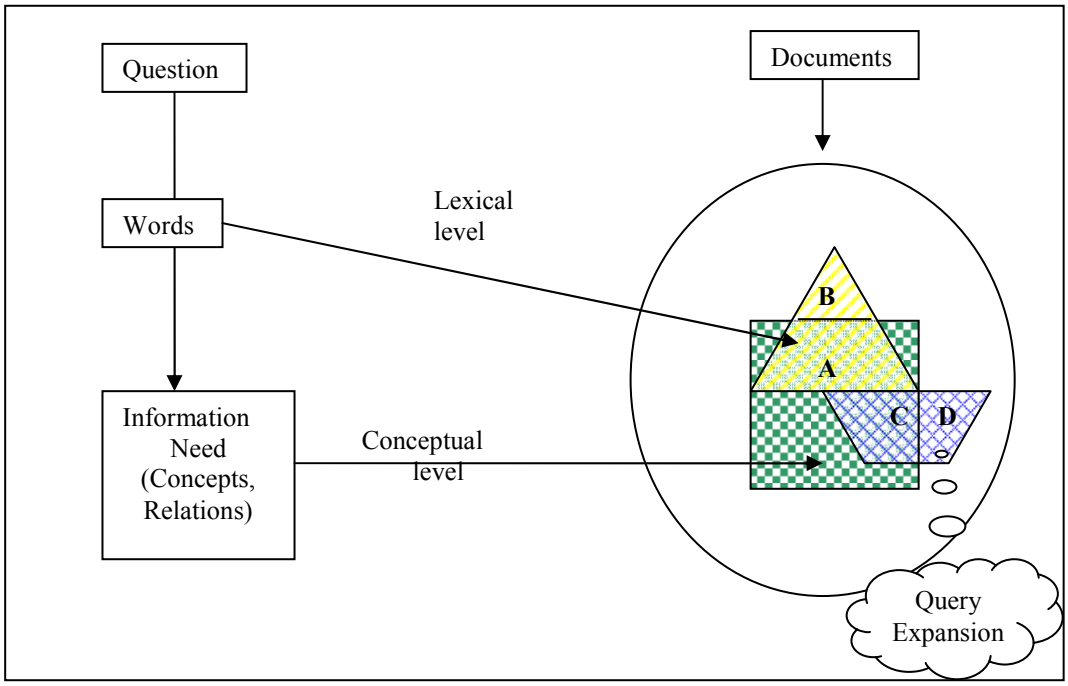


Figure 6. Sketch of the lexical solution.

For the goal of presenting a solution based onto an approximation for language understanding, we will consider the latter as being a factorization of semantics and syntax. While semantics is meant to deal with the task of extracting the right meaning or concept for a word, syntax is responsible for finding the relations that hold between such concepts in the context given by a question. The solution presented in this thesis (a system called Quantico) works at the lexical level of the question, with words instead of concepts, and provides

several methods to approximate the process of both semantics and syntax as previously defined. We will roughly present the idea of our solution by way of an example. Let us consider the following question being asked to a Question Answering system:

Who filed the suit with the federal court against O. J. Simpson?

Both words *suit* and *court* are ambiguous in their meaning according to the listing below:

suit

- <*suit*, lawsuit, case>
- <*suit*, suit of clothes>

court

- <*court*, courtroom, tribunal>
- <*court*, courtyard>

If the QA system could understand the question, it would recognize the correct meanings of the words *suit* and *court* as being <*suit*, lawsuit, case> and <*court*, courtroom, tribunal>, and identify the relationship of *charging* to hold between these. Assuming the same process on the documents side, the system will have to look-up and retrieve at the conceptual level only those segments referring to this specific meaning of the request and extract the correct answer referencing a person. These relevant segments would correspond to the patterned rectangle of Figure 6. Failing to disambiguate the lexical items would result in retrieving segments that contain the words *suit* and *court* (triangle), both relevant (part A) by addressing the right meanings (semantics) and relationships (syntax) and irrelevant by addressing either correct meanings but wrong relationships or other meanings (part B). Moreover, relevant segments mentioning synonyms of the right concepts (i.e. lawsuit, tribunal) would still be missed. To tackle this last issue, expansion of the information need with synonyms can be done in order to increase the number of relevant segments (part C). Since no way of

disambiguating the words is available, synonyms for all known meanings will be added, resulting therefore in some noise being introduced as well (part D).

We want to address the issue of irrelevant data being retrieved by devising a method to approximate the process of having both a semantic and a syntactic component in place. The goal of this method is to minimize the number of segments that are not relevant to a question, by filtering out those segments corresponding to the parts B and D in Figure 6. For that reason we will use *proximity* as a way of approximating syntactic relationships and we will narrow the length of the retrieved textual segments to only a few *sentences* (1, 3 or 5) as a way of approximating semantic disambiguation. The latter approach leverages the *one sense per collocation* property of human languages, according to which words tend to exhibit only one sense in a given collocation.

Proximity matters because words that are close to each other in the text are more likely to be closely connected in the meaning structure of the text. It is true that words in a question have some explicit or implied linguistic relationship between them, and that a good match for such questions is likely to be one that has the same relationship between those words. This is why we use proximity in our context as a crude irrelevance filter. Proximity increases the chance to find question words in some relationship, which in turn increases the chance of getting the words in the right relationship. But it's common in linguistics that structural connections are not that obviously connected to distances in the surface string. In examples like these:

- *Chapman shot Lennon.*
- *Lennon, member of the most famous rock band Beatles, has been shot by a fan named Chapman.*

Lennon is just as related to Chapman when separated by one word as by fifteen words: in both examples Chapman plays the role of the agent in a thematic relation with the verb, while Lennon is the patient that undergoes the action of the verb. While proximity will clearly put at a disadvantage this kind of structure, it is a trade-off that we will accept when dealing with large open-domain

collections of documents, where redundancy of data will rather favor shorter, more direct relations.

The *One sense per collocation* approach to approximate semantic disambiguation builds upon work done by Yarowsky, D. (1995), according to which nearby words provide strong and consistent clues to the sense of a target word, conditional on relative distance, order and syntactic relationship. We apply this idea such that nearby words are represented by other words of the question in the context of a *sentence*, whereby relative distance and syntactic relationship will be covered by the *proximity* concept described above. Intuitively, we expect that by narrowing the length of the retrieval unit to that of a sentence, irrelevant meanings of question words will rarely co-occur within the local context of a sentence. In other words we expect <*lawsuit, tribunal, O. J. Simpson*> to co-occur more often than <*lawsuit, courtyard, O. J. Simpson*> or <*suit of clothes, tribunal, O. J. Simpson*>.

Moreover, we can cast the task of finding the correct answer to a question to that of semantic disambiguation in line with the *One sense per collocation* approach. If we consider the expected answer type of a question (i.e. asking for a PERSON) as a possible ambiguous word and candidate answers as possible senses of it, then finding the most frequent answer co-occurring within sentence context with the question words and being constrained by its proximity to them, will result in providing the right answer. This way, *proximity* and *redundancy* are two strong clues for assessing the correctness of an answer to a question.

In the following subsections we will present the architecture of the monolingual German system and follow describing two different techniques of extending it to the English-German scenario of use. We finally conclude this chapter giving an overview of the evaluation methodology pursued to assess the performance of the system both at component-level and as a whole.

5.1 Monolingual QA

Quantico is a Question Answering system designed from the ground up to support both English and German as working languages. The first version was deployed as a monolingual system to cover English and German questions posed to document collections in the same language. For each of the languages an

instance of the system is up and running based on a framework (Figure 7) shared across them.

This common framework consists of three sub-systems as known from the traditional QA pipeline:

- **Question Analysis** – whose role is to interpret the *anatomy* of a question in terms of:
  - question type (definition or factoid),
  - expected answer type (i.e. PERSON, ORGANIZATION, OTHER, etc.),
  - focus, and
  - topic.

- **Unit Retrieval** – whose role is to narrow down the search space of answer-bearing textual segments to a ranked list of relevant ones.

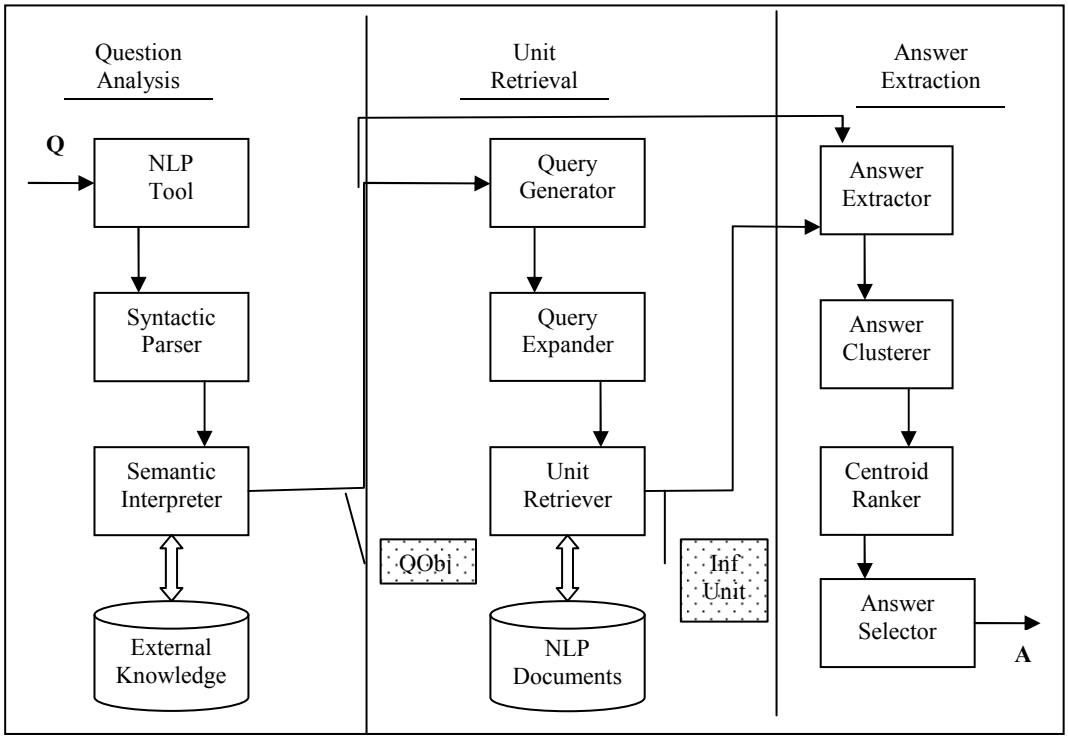


Figure 7. Common Framework of Quantico

- **Answer Extraction** – whose role is to extract possible answer candidates according to the expected answer type and select the best answer based on some *fitness* criteria.

Question Analysis – *What it does?*

The Question Analysis sub-systems reads in the user's information need as a natural-language question (i.e. Wieviele Bundesländer hat Österreich?) and generates a formal representation of its meaning, a QObject, as presented in Figure 8.

The *question type* (Q-TYPE) is a categorization of questions for purposes of distinguishing between different processing strategies and answer formats. We distinguish between FACTOID and DEFINITION questions with different weighing schemes for their unit retrieval and diverse answer size ranging from word to phrase and even full-length sentence for definitions.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<QOBJ score="1" msg="quest" lang="DE" id="qId0">
  <NL-STRING id="qId0">
    <SOURCE lang="DE" id="qId0">Wieviele Bundesländer hat Österreich ?</SOURCE>
    <TARGETS/>
  </NL-STRING>
  <QA-control>
    <Q-FOCUS>Bundesländer</Q-FOCUS>
    <Q-TOPIC>Österreich</Q-TOPIC>
    <Q-TYPE restriction="NONE">FACTOID</Q-TYPE>
    <A-TYPE type="atomic">NUMBER</A-TYPE>
  </QA-control>
  <KEYWORDS>
    <KEYWORD type="UNIQUE" id="kw1">
      <TK stem="bundesland" pos="N">Bundesländer</TK>
    </KEYWORD>
    <KEYWORD type="UNIQUE" id="kw2">
      <TK stem="Österreich" pos="N">Österreich</TK>
    </KEYWORD>
  </KEYWORDS>
  <EXPANDED-KEYWORDS/>
  <NE-LIST>
    <NE type="LOCATION" id="ne0">Österreich</NE>
  </NE-LIST>
</QOBJ>
```

Figure 8. Result of Question Analysis.

The expected *answer type* (A-TYPE) represents the class of object sought by the question. Its semantic category drives both the retrieval of segments that contain answer candidates and their actual extraction. We consider the following 8 answer types for FACTOID questions, as defined by the CLEF test collection:

- PERSON, e.g.
  Q: Who was called the “Iron-Chancellor”?
  A: Otto von Bismarck.
- TIME, e.g.
  Q: What year was Martin Luther King murdered?
  A: 1968.
- LOCATION, e.g.
  Q: Which town was Wolfgang Amadeus Mozart born in?
  A: Salzburg.
- ORGANIZATION, e.g.
  Q: What party does Tony Blair belong to?
  A: Labor Party.
- MEASURE, e.g.
  Q: How high is Kanchenjunga?
  A: 8598m.
- COUNT, e.g.
  Q: How many people died during the Terror of Pol Pot?
  A: 1 million.
- OBJECT, e.g.
  Q: What does magma consist of?
  A: Molten rock.
- OTHER, i.e. everything that does not fit into the other categories above.
  Q: Which treaty was signed in 1979?
  A: Israel-Egyptian peace treaty.

and the following four answer types for DEFINITION questions:

- PERSON, i.e. questions asking for the role/job/important information from a biographical point of view about someone,
  Q: Who is Robert Altmann?
  A: Film maker.

- ORGANIZATION, i.e. questions asking for the mission/full name/important information from a biographical point of view about an organization, e.g.
  Q: What is the Knesset?
  A: Parliament of Israel.
- OBJECT, i.e. questions asking for the description/function of objects, e.g.
  Q: What is Atlantis?
  A: Space Shuttle.
- OTHER, i.e. question asking for the description of natural phenomena, technologies, legal procedures, etc., e.g.
  Q: What is Eurovision?
  A: Song contest.

Both the question type and the expected answer type are salient information for a good performance of downstream components and failure to correctly determine them will deem the system unusable in most of the cases.

The question *focus* (Q-FOCUS) represents the property or entity that is being sought by the question and may or may not appear in the context of the correct answer, which in most of the cases is implied by it (e.g., country, city, name, age, date).

The question *topic* (Q-TOPIC) is the object (person, organization, …) or event that the question is about, whose meaning must appear in the context of the right answer.

The Question Analysis is also responsible for extracting additional constraints that the correct answer has to satisfy. Such constraints can take different forms like keywords and named entities. The keywords of the question might contain, beside the focus and the topic, lexicalizations of the relation between the two, usually in the form of a verb, and dependents or modifiers of them, which put further constraints on their meaning. Named entities recognition is also an integral part of the Question Analysis due to their special treatment during retrieval of relevant information and extraction of candidate answers.

Question Analysis – *How it works?*

The Question Analysis sub-system consists of three components: a NLP Tool, a Syntactic Parser and a Semantic Interpreter. The NLP Tool is mainly responsible for recognizing named entities and annotating them with their semantic type, according to the classification imposed by the test collection.

The Syntactic Parser's role is that of providing a list of lexical dependencies that hold between the words of the question; these dependencies form the basis for the next component. The Semantic Interpreter builds upon both these dependencies and a set of hand-crafted lexico-syntactic rules to determine the control information of the *QObject*. In this process it makes use of an external knowledge base of entities that provide hints for the expected answer type based on the focus of the question (e.g. *In which city* …. → Q-FOCUS: *city* → A-TYPE: *LOCATION*).

Unit Retrieval – *What it does?*

In line with our goal of approximating sense disambiguation by reducing the length of the retrieval unit, the document collection has been anticipatory annotated with sentence boundaries. The preemptive offline annotation additionally processed the document collection 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 focused results than those of a search engine based solely on words.

Unit Retrieval – *How it works?*

The Query Generator process mediates between the question analysis result *QObj* (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 Query Generator process builds on an abstract description of the processing method for every type of question to accordingly generate the IR query 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 IR query 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*), like 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 Query Extension component accounts for bridging this gap at the lexical level, either through look-up of hand-crafted unambiguous resources (e.g. *French ~ France ~ Frenchman*) or searching external resources like wordnets and thesauri for synonyms and conceptually related terms (e.g. *prime-minister ~ government leader*).

In the context of our experiments three different settings have been considered for the retrieval of relevant textual segments for factoid questions: one in which a passage consists of only a sentence as retrieval unit, a second one with a window of three adjoining sentences for a passage, and a third one with a window of five adjoining sentences for a passage. Concerning the query generation, only keywords with following part-of-speeches have been used for retrieval: nouns, adjectives, adverbs and verbs, whereby only nouns and adjectives are mandatory to occur in the matching relevant segments, with nouns corresponding to the question's topic higher weighed (*^weight*). In case of empty hit list retrieval, the query undergoes a relaxation process maintaining only the topic of the question, its modifiers and the expected answer type (as computed by the Question Analysis sub-system) as mandatory items:

Question: How many provinces does Austria have?
IR-Query: +neTypes:LOCATION +text: province +text:Austria^4 text:have
Relaxed IR-Query: +neTypes:LOCATION text: province +text:Austria^4
text:have

Answer Extraction - *What it does?*

The Answer Extraction & Selection sub-system 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. Using the most representative sample (centroid) of the answer candidates' best-weighed clusters, the Answer Selector sorts out a list of top answers based on a *proximity* metric defined over a graph representation of the answer's context.

Answer Extraction - *How it works?*

Based on the control information supplied by the Question Analysis sub-system (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 Answer Extractor 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 Answer Clusterer (normalization) process consists in resolving cases of co-reference, 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).

Within the Answer Selector 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 our experiments, 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.

5.2 Cross-language Methods

For the use case of answering questions asked in a language different than that of the document collection (e.g. English question and German documents) we have considered question translation as the most tractable strategy for crossing the language barrier.

It is widely recognized that there are three main approaches to translation in cross-lingual information access technologies:

- Machine Translation (MT),
- Translation by bilingual machine readable dictionaries (MRD), and
- Parallel or comparable corpora based methods.

Machine Translation Techniques

Intuitively, the MT system seems to be a good approach for cross-lingual QA and availability of high-quality MT software, able to give the user as good an idea as possible of the meaning of what is translated, makes the task much easier. Yet, for question translation, the MT has not always provided better performance than

that of a dictionary based approach. One of the reasons is that questions are often too short and do not provide sufficient contextual information for appropriately dealing with ambiguous words. Moreover, by selecting only one translation from the many candidates that the source words may have, MT prevents the system from expanding the original question by synonyms or related words.

Dictionary-based Methods

Using a bilingual MRD is the preferred approach when no high-quality MT system is available. In general, most Question Answering systems are based on "bag-of-words" architectures, in which both questions and documents are decomposed into a set of words through a process of indexing. Thus we can translate a question easily by replacing each question term with its translation equivalents from a bilingual dictionary. However, there are some problems to be noted:

- Dictionary translations are inherently ambiguous and add extraneous information.
- Failure to translate multiterm concepts such as phrases and named entities reduces effectiveness.
- Different languages have different syntax to govern the sentential structure and simply chaining up the translations in the order given by the source language won't work in most of the cases.

Parallel Corpora-based Methods

Parallel or comparable corpora are useful resources for extracting translation equivalents in the form of bilingual term lists. One disadvantage of methods based on the use of parallel and comparable corpora is lack of resources: parallel corpora are not always readily available and those that are available tend to be relatively small or to cover only a small number of subjects.

Of the above-mentioned approaches, the first two have been considered in this thesis and experimentally tested to compare their suitability to extend the actual mono-lingual design for a cross-lingual scenario.

5.2.1 Direct Translation

Under this notion we mean translation of the original question by means of online free-available MT services (Figure 9). This seems to be the most intuitive method when these kinds of tools, with general good performance, are on hand. For the purpose of our experiments we have used Google's translation service[2] (as of December 2009), powered by a statistical machine translation engine. Giving an English question, it gets translated into German through Google Translate and the result is passed to the German monolingual QA system. Before translation, the question is marked up with named entities and those with a type different from LOCATION are substituted by a placeholder. After translation the place holders are substituted back with their initial values, such that everything but LOCATION names remains unchanged.

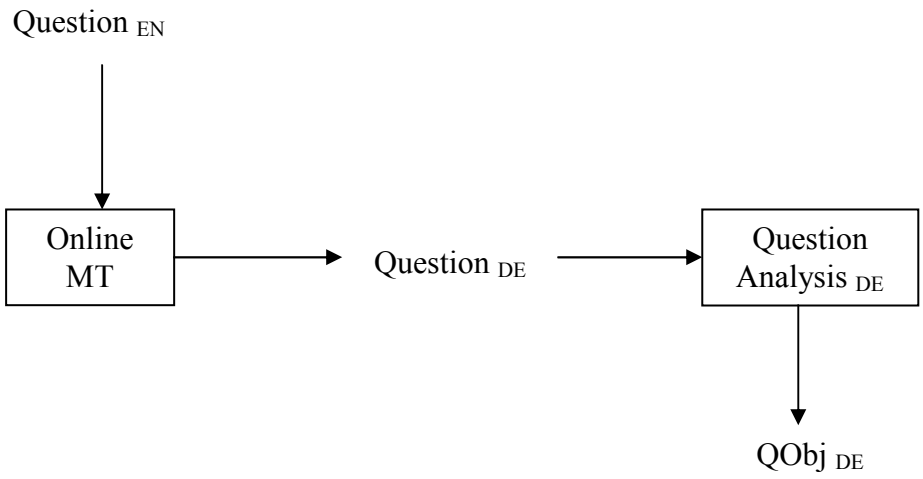


Figure 9. Direct Translation Method.

Beside the above-mentioned issues for this kind of translation we expect to face another problem due to the current implementation of our monolingual system: syntactically ill-formed translations will affect the performance of the Question Analysis sub-system since it relies on grammatically correct input to determine salient information like question type (Q-TYPE) and expected answer type (A-TYPE).

[2] http:// translate.google.com/

5.2.2 Transfer-based Translation

The transfer-based translation attempts to cover this sensitivity problem of the previous method by analyzing the question to begin with and then *transferring* the result of the Question Analysis sub-system, the *QObj*, into the target language (Figure 10). This approach assumes though the existence of a Question Analysis sub-system for the source language, as well.

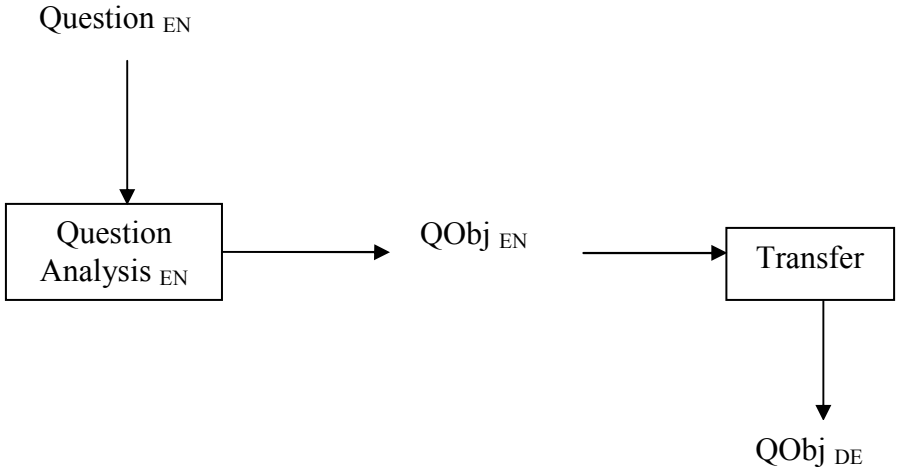


Figure 10. Transfer-based Method.

Since the *QObj* is a template structure representing part of the question's semantic through its fields, we can therefore make sure that the essence of the user's information need has been accurately captured by the analysis of the syntactically well-formed source input. The values of *QObj*'s fields are words and phrases that are best suited for word-by-word translation techniques like machine readable dictionaries and term lists generated from parallel data.

Method 1: Machine Readable Dictionaries

However, MRD come with their shortcomings as well. Trying to overcome the issues brought in by a direct translation, we have to make sure that we are not running into potential bigger problems with this new approach. What use would we have from a properly synthesized question type or focus if we would not be able to correctly translate them? Here is how we tackle the issues of MRD:

Dictionary translations are inherently ambiguous and add extraneous information.

This problem is partly covered by using part-of-speech (POS) tags for translation disambiguation and partly through the actual design of the monolingual system by working with sentences as retrieval units. Irrelevant meanings of translated question words will rarely co-occur within the local context of a sentence such that word sense disambiguation techniques for translation equivalents are not employed in first place.

Failure to translate multi-term concepts such as phrases and named entities reduces effectiveness.

We address this issue by recognizing named entities during the Question Analysis, even before the *QObj* template is generated, and considering them as immutable units during translation. We only make an exception for named entities of type LOCATION that are usually translated. As for the multi-term concepts, we treat them as such when they appear as template slot values in the *QObj* and only when no translation is available we split them into words.

Different languages have different syntax to govern the sentential structure.

This problem is already tackled by analyzing the source question in the first place before translating the result of its analysis. The Question Analysis sub-system interprets the information need based on the syntax of the source question such that when translating the values for the slots of the *QObj* the syntax of the target language is not important anymore (Figure 11).

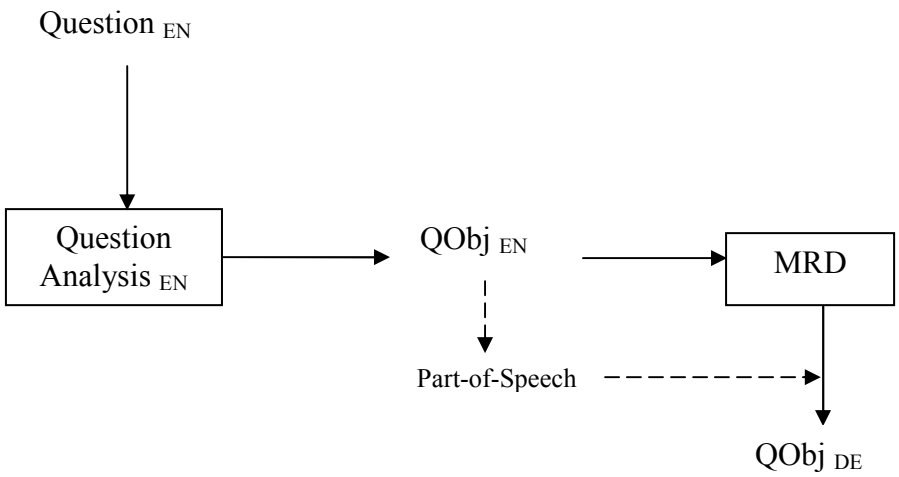


Figure 11. MRD Transfer-based Method.

The coverage of machine readable dictionaries, while not deep, is broad enough to be used for translation of words covering a wide variety of topics. For our experiments we have used an online bilingual dictionary, LEO[3], which provides more than 550,000 entries and has been considered mainly due to its wide coverage for both single and multiple-word terms. Part-of-Speech (POS) information, as generated by the Question Analysis sub-system, has been used to select only translations having the same POS with that of the source term.

Method 2: Automatically Generated Term Lists

The second method used to transfer the result of the Question Analysis, the *QObj*, goes along the word-by-word translation idea, but instead of using bilingual dictionaries it generates translation equivalents through word alignment of the MT result to the original question (Figure 12). This method is to be preferred when MT tools are readily available, but they fail to reconstruct the correct syntactic structure in the target language. The advantage of this approach over using MRD comes from the fact that machine translation software has to pick up at some stage the best translation of a word given the question's context and indirectly achieve the goal of word sense disambiguation. If several different MT tools are considered they possibly generate alternative formulations of the same meaning, which can be used to extract pairs of semantically equivalent words and phrases.

[3] http://dict.leo.org/

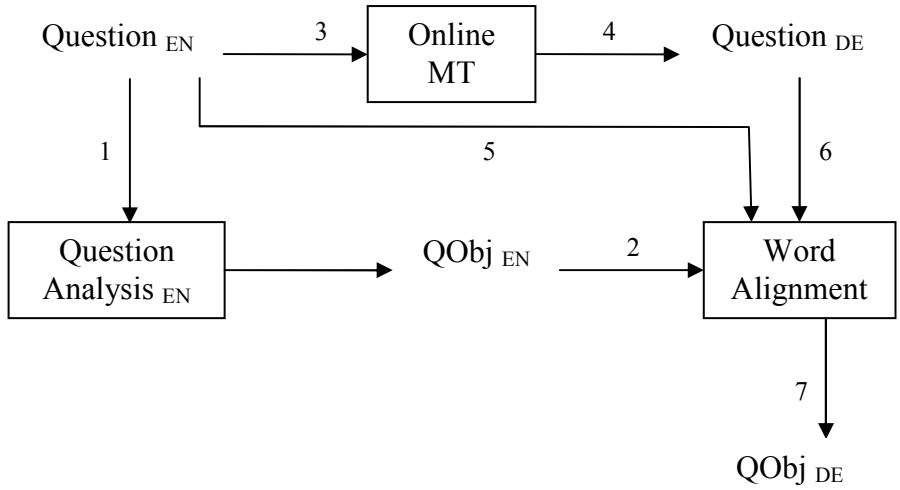


Figure 12. MT Transfer-based Translation.

There are generally two approaches to word alignment: the association approach, using some measures of correspondence, and the estimation approach, using probabilistic models. Since our parallel data consists only in one sentence (the original question vs. its translation), the latter method has to be excluded. Association-based word alignment generally undergoes three steps:

- lexical segmentation, when boundaries of lexical items are identified;
- correspondence, when possible translations are suggested in line with some correspondence measures;
- alignment, when the most likely translation is chosen.

In a first step we tokenize the sentence and its translations into a list of words. Next we employ several alignment techniques based on string similarity measures, bilingual dictionaries and part-of-speech (POS) tags. They all act like filters on a full alignment, where each source word is associated with all target words, and let through only those alignments that pass their internal selection criteria or threshold. The following filters have been considered in the development:

- Part-of-speech (based on TnT - Brants Thorsten, 2000)
- Bilingual dictionary (LEO)
    - Direct translation
    - Back propagation

- String similarity for cognates
  - Dice coefficient
  - Lowest common subsequence ratio (LCSR)
- Overlap

We describe the working of the alignment component along the following fabricated example:

Question: *What is the name of the Russian governmental news agency?*

Translation: *Wie heißt Russlands staatliche Nachrichtenagentur?*

To begin with, full alignments for every source word are generated (Figure 13) that are the target of a filtering process as described below. Every alignment has a Boolean value of *true* if already aligned and a weight associated with it.

what:	{[Wie, heißt, Russlands, staatliche, Nachrichtenagentur]} false
is:	{[Wie, heißt, Russlands, staatliche, Nachrichtenagentur]} false
the:	{[Wie, heißt, Russlands, staatliche, Nachrichtenagentur]} false
name:	{[Wie, heißt, Russlands, staatliche, Nachrichtenagentur]} false
of:	{[Wie, heißt, Russlands, staatliche, Nachrichtenagentur]} false
russian:	{[Wie, heißt, Russlands, staatliche, Nachrichtenagentur]} false
governmental:	{[Wie, heißt, Russlands, staatliche, Nachrichtenagentur]} false
news:	{[Wie, heißt, Russlands, staatliche, Nachrichtenagentur]} false
agency:	{[Wie, heißt, Russlands, staatliche, Nachrichtenagentur]} false

Figure 13. Initial word alignment.

We first use the POS filter in order to exclude unlikely alignments based on the part-of-speech tags of the words being considered (Figure 14). Beside one-to-one alignment of words with similar POS tags we allow following additional mappings (DE to EN):

- noun to adjective (i.e. *undercover agent* vs. *Geheimagent*)
- verb to prepositional or adverbial particle (i.e. *shut up* vs. *verschließen*)
- verb to noun (i.e. *take place* vs. *geschehen*)

in order to account for the most of the structural changes during translation between English and German, as well as for German composite nouns.

The dictionary-based filters are next, with the DirectFilter looking up

```
{name=NN, governmental=JJ, is=VBZ, the=DT, agency=NN, what=WDT, news=NN,
of=IN, Russian=JJ}
{Wie=PWAV, Russlands=NN, staatliche=ADJA, Nachrichtenagentur=NN,
heißt=VVFIN}


what:          [Wie] false
is:            [heißt] false
the:           [] false
name:          [Russlands, Nachrichtenagentur, heißt] false
of:            [heißt] false
russian:       [Russlands, staatliche, Nachrichtenagentur] false
governmental:  [Russlands, staatliche, Nachrichtenagentur] false
news:          [Russlands, Nachrichtenagentur, heißt] false
agency:        [Russlands, Nachrichtenagentur, heißt] false
```

Figure 14. Part-of-Speech filtering.

translations of the English words and matching them against those in the actual alignment (Figure 15) and the BackPropagationFilter looking up words in the opposite direction (Figure 16). The latter filter is covering alignment of English complex terms and phrasal verbs that are translated into one single German correspondent.

```
what:          [Wie] false
is:            [heißt] false
the:           [] false
name:          [Russlands, Nachrichtenagentur, heißt] false
of:            [heißt] false
russian:       [Russlands, staatliche, Nachrichtenagentur] false
governmental:  [staatliche] TRUE
news:          [Russlands, Nachrichtenagentur, heißt] false
agency:        [Russlands, Nachrichtenagentur, heißt] false
```

Figure 15. DirectFilter dictionary look-up.

```
what:          [Wie] false
is:            [heißt] false
the:           [] false
name:          [Russlands, Nachrichtenagentur, heißt] false
of:            [heißt] false
russian:       [Russlands, staatliche, Nachrichtenagentur] false
governmental:  [staatliche] TRUE
news:          [Nachrichtenagentur] TRUE
agency:        [Nachrichtenagentur] TRUE
```

Figure 16. BackPropagationFilter dictionary look-up.

Between filters that are able to mark an alignment as *true* we employ an OverlapFilter that excludes already aligned words from the rest of the open alignments (Figure 17).

what:	[Wie] false
is:	[heißt] false
the:	[] false
name:	[Russlands, ~~Nachrichtenagentur~~, heißt] false
of:	[heißt] false
russian:	[Russlands, ~~staatliche~~ , ~~Nachrichtenagentur~~] false
governmental:	[staatliche] TRUE
news:	[Nachrichtenagentur] TRUE
agency:	[Nachrichtenagentur] TRUE

Figure 17. OverlapFilter.

Finally, the alignment methods based on string similarity measures are used that are best suited for discovering *cognates*, etymologically related words across languages, by way of their spelling. We use therefore a variant of the *Dice coefficient* for character bigrams formulated as follows:

$$DiceCoefficient = \frac{2 * \left| bigrams(x) \cap bigrams(y) \right|}{\left| bigrams(x) \right| + \left| bigrams(y) \right|} \qquad (3.1)$$

and another measure called *longest common subsequence ratio* (LCSR). The LCSR is another measure of string similarity that takes advantage of the observation that parts of a string may be similar while the prefixes and suffixes are not (or any other part of the string). The LCSR is computed by finding the longest substring in common between the two strings and returning the ratio of the length of that string to the length of the longer of the two words in the pair. Both measures have been adapted to address the property of sound shifting for German and English, covering both consonants and vocals according to the tables Table 6 and Table 7 in Annex 1.

what:	[Wie] false
is:	[heißt] false
the:	[] false
name:	[Russlands, heißt] false
of:	[heißt] false
russian:	[Russlands] TRUE
governmental:	[staatliche] TRUE
news:	[Nachrichtenagentur] TRUE
agency:	[Nachrichtenagentur] TRUE

Figure 18. LCSR Filter.

For our example only the LCSR filter triggered a change (Figure 18) in the final alignment of the translation (Figure 19). It is this alignment that provides a list of terms with their most likely translations to be considered for transferring the result of the Question Analysis from the source into the target language.

what:	[Wie] false
is:	[heißt] false
the:	[] false
name:	[~~Russlands~~, heißt] false
of:	[heißt] false
russian:	[Russlands] TRUE
governmental:	[staatliche] TRUE
news:	[Nachrichtenagentur] TRUE
agency:	[Nachrichtenagentur] TRUE

Figure 19. Overlap Filter and Final Alignment.

5.3 Summary

If we were to consider in the context of Question Answering a comparison of the above mentioned methods along their stated weak/strong features, we could draw the following table:

	Syntactic Structure	Word Translation
Direct Translation	o	o
Method 2	+	o
Method 1	+	+/-

Table 1. Comparison of question translation methods.

According to it, we expect the second transfer-based translation method to outperform the direct translation one, given that the word alignment process does a reasonable job on aligning the word translations back to their source. Regarding the MRD method, we also expect it to outperform the direct

translation one, given that the assumption of co-occurring relevant meanings in local context holds true. Since we are using variable sizes of context as retrieval units, for some of them this assumption might be invalidated.

6 Question Analysis

Question Analysis is the key component of a Question Answering system, since it interprets the user request and transfers it in a system-internal representation, based on which downstream components do their work. Failure to correctly *understand* the question at this stage may result either in further components not being triggered or wrong answers being provided. The main purpose of the Question Analysis is to find a question's type and focus, and the expected answer type, first of all, and to identify further constraints, like contextual keywords, that the correct answer has to fulfill.

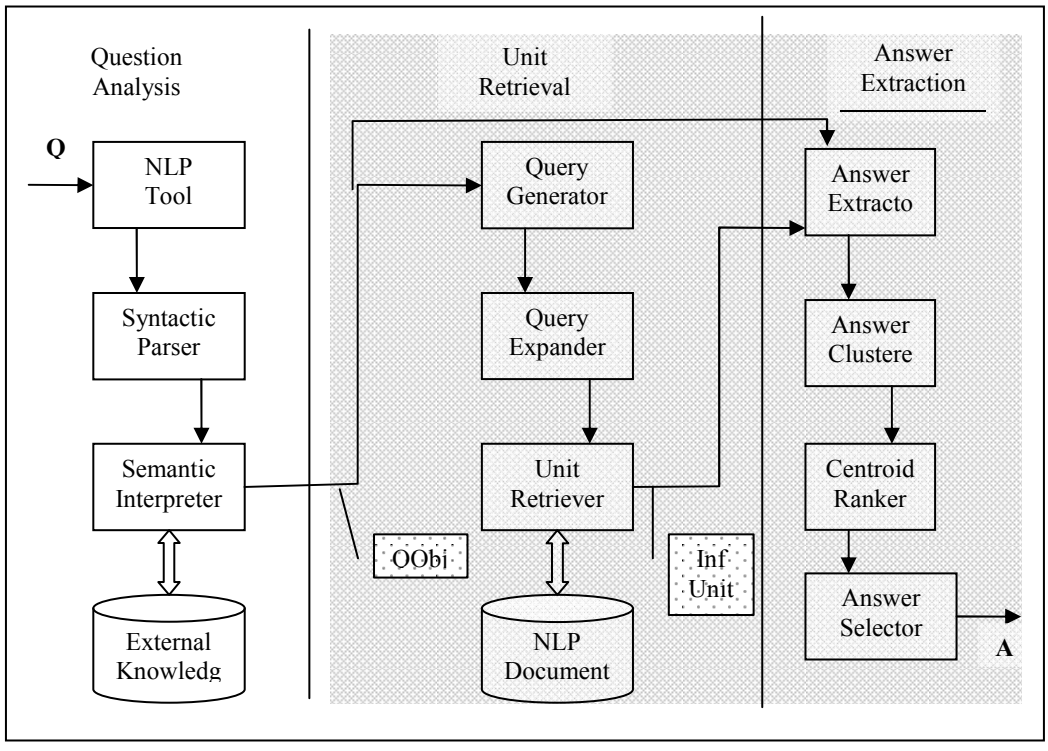


Figure 20. Question Analysis Architecture.

The Question Analysis process (Figure 20) starts with the recognition of the named entities in the question. This is important because of the special treatment offered to named entities during document indexing and search, as well as during

translation in a cross-language scenario. Therefore we use as *NLP Tool* LingPipe[4], a suite of libraries for the linguistic analysis of the human language, that will be described in more detail in Section 7.1.

For the goal of this thesis, we will consider that both English and German have a compositional semantics, i.e. the meaning of their utterances is structured according to their syntax. Therefore, the first step to describing the meaning of an utterance in a language is to analyze it and look at its analyzed form. Along this assumption we will first syntactically analyze the question and then build upon its syntax a semantic interpretation for our purpose, i.e. the *QObj*.

A natural language parser is a program that works out the grammatical structure of sentences - for instance, which groups of words go together (as "phrases") and which words are the subject or object of a verb. We use two kinds of grammars for our parsing needs: a phrase structure and a dependency grammar. The phrase grammar is used for determining the major constituents of the question, like noun phrases, while the dependency grammar gives the relations at the lexical level in form of governor/dependent pairs and grammatical functions like subject and object.

Based on the structures computed by the syntactic parser, we define a set of hand-crafted rules in order to identify the semantics of the question by determining the question type, its focus and topic, the expected answer type and further constraints in the form of keywords.

6.1 German Analysis

In the context of the monolingual German QA system we use the Semantic Interpreter described in Neumann, G. & Sacaleanu, B. (2006) to represent the result of a NL question analysis as a *declarative description of search strategy and control information*. Consider, for example, the NL question result (Figure 21) for the question *Wie heißt Russlands staatliche Nachrichtenagentur? (What is the name of the Russian governmental news agency?)*:

[4] http://alias-i.com/lingpipe/

```
<QOBJ score="1" msg="quest" lang="DE" id="qId0">
<NL-STRING id="qId0">
<SOURCE lang="DE">Wie heißt Russlands staatliche
Nachrichtenagentur?</SOURCE>
</NL-STRING>
<QA-control>
<Q-FOCUS>Nachrichtenagentur</Q-FOCUS>
<Q-TOPIC>Russlands</Q-TOPIC>
<Q-TYPE restriction="NONE">FACTOID</Q-TYPE>
<A-TYPE type="atomic">ORGANIZATION</A-TYPE>
</QA-control>
<KEYWORDS>
<KEYWORD type="UNIQUE" id="kw0">
<TK stem="heiss" pos="V">heißt</TK>
</KEYWORD>
<KEYWORD type="UNIQUE" id="kw1">
<TK stem="russland" pos="N">Russlands</TK>
</KEYWORD>
<KEYWORD type="UNIQUE" id="kw2">
<TK stem="staatlich" pos="A">staatliche</TK>
</KEYWORD>
<KEYWORD type="UNIQUE" id="kw3">
<TK stem="nachrichtenagentur" pos="N">Nachrichtenagentur</TK>
</KEYWORD>
</KEYWORDS>
<NE-LIST/>
</QOBJ>
```

Figure 21. Result of German Question Analysis.

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 (Neumann, G. & Piskorski, J., 2002) and
- in a second step, a question-specific semantic analysis is performed.

During the second step, the values for the question tags A-TYPE, Q-TYPE, Q-FOCUS and Q-TOPIC are determined on the 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*.

6.2 English Analysis

Questions in English can be asked in different forms, distinguishable by their structure:

- **Indirect Question**: *I wonder where the house is?*
- **Direct Closed**:
  - Yes/No: *Will you be in town for your appointment?*
  - Tag: *You want to join us, isn't it?*
  - Intonated: *Your friend never expects your help?*
  - Alternative: *Do you want to go or stay longer?*
- **Direct Open**:
  - Simple: *Who is your sister's boyfriend?*
  - Complex: *What happened when John came home?*

Direct questions are main clauses, whereas indirect questions are part of a larger matrix sentence, which can be a question itself. Direct questions are generally used to elicit information, while indirect questions are generally used to report about direct questions.

Direct closed questions are those questions, which demand a yes/no, true/false or right/wrong answer. Direct open questions leave more room for a

description and are more useful to obtain information. Open questions are also known as constituent or *wh*-questions because the answer to them is expressed by a constituent that corresponds to the *wh*-phrase in the question. *Wh*- phrases are so called because they generally begin with *wh*- in English (*who, what, which, where, when, why*). *How* counts as a wh- expression because of its meaning, even though it does not begin with *wh*-.

Direct open simple questions are targeted by most of the Question Answering systems because of their popularity with the users of search engines (Spink, A., & Ozmutlu, H. C., 2001). They are also called factoid questions and have short answers, typically a noun phrase or a simple verb phrase, or an enumeration of such answers. Most of these questions are object questions that ask about an object, but questions to find out about the subject are also common. Both subject and object questions are characterized by a well-defined syntactic structure that makes possible the use of parsers to extract their information need (Figure 22).

Subject Questions
WH-phrase (subject) auxiliary* main_verb

Object Questions
WH-phrase (object) auxiliary subject main_verb

Figure 22. Common Structure of Open Questions.

6.2.1 Syntactic Parser

For syntactically analyzing English questions, we employ the statistical Stanford parsers that provide typed dependencies, otherwise known as grammatical relations, as well as phrase structure trees (Figure 23). Probabilistic parsers use knowledge of language gained from hand-parsed sentences to try to produce the *most likely* analysis of new sentences. These statistical parsers still make some mistakes, but commonly work rather well (86.3% F1 score according to Klein, D. & Manning, C. D., 2003).

The Stanford parsers are a Java implementation of probabilistic natural language parsers, both highly optimized PCFG and lexicalized dependency parsers, and a lexicalized PCFG parser. The lexicalized probabilistic parser implements a factored product model, with separate PCFG phrase structure and

lexical dependency experts, whose preferences are combined by efficient exact inference, using an A* algorithm. (Klein, D. & Manning, C. D., 2003)

Phrase Structure Tree

```
(SBARQ
        (WHNP (WP What))
        (SQ (VBZ is)
                (NP
                        (NP (DT the) (NN name))
                        (PP (IN of)
                                (NP (DT the) (JJ Russian) (JJ governmental)
                                (NN news) (NN agency)))))
        (. ?))
```

Typed Dependencies

```
det(agency, the)
nn(agency, news)
amod(agency, governmental)
det(name, the)
prep_of(name, agency)
nsubj(is, name)
amod(agency, Russian)
attr(is, What)
```

Figure 23. Output of Stanford Parser.

6.2.2 Semantic Interpreter

The goal of the semantic interpreter is to provide a systematic approach to interpreting the information need of the question, building upon results of the syntactic parser. For our purposes, the semantic of a question is the synthesis of control information and constraints thereof. That is, we reduce an inquiry to a set of representative question focus and expected answer type, and a list of keywords that impose additional restrictions on the answer. Therefore, we developed hand-crafted rules that capture expected answer type, focus and keywords based on syntactic parse trees and dependency relations.

Rule Engines – DROOLS

A Rule Engine focuses on knowledge representation to express propositional and first order logic in a concise, non-ambiguous and declarative manner. Knowledge is represented as a set of rules and data is represented as a set of facts. The rule engine compares each rule in the knowledge base (the rules) with the facts. If a

rule matches a fact (conditions are fulfilled), the rule is said to “fire”, and the “then” action (consequence) is executed.

These rules are not new: they are the logic that is the core of many software applications. The primary difference with a rule engine is in how these rules are expressed; instead of embedding them within the program, these are encoded in external rule files. The problems with traditional hard-coded or hard-wired rules (in the form of *if-then-else* programming statements) include:

- Duplicate rules must be coded & maintained in many systems
- It's hard to isolate rules from code during maintenance
- It's even harder to change and test applications

The benefits of the rule engines approach include:

- Shared rules (reuse)
- Rules coded once
- Rules are isolated from code
- Externalizing rules results in smaller applications
- Smaller applications make it easier to change and test applications

Drools and other rule engines offer the benefits of letting a developer write their rules in a declarative fashion while implementing the logic in a language they are familiar with, such as Java. The key advantage of using rules is that they can make it easy to express solutions to difficult problems and consequently have those solutions verified, as rules are much easier to read then code.

The underlying nature of the rule engine comes from the algorithm that drives it; some simple ‘rule engines’ simply chain procedural logic together in an order that you specify. Most engines offer sophisticated matching algorithms like Rete, Treat and Leaps to connect facts with rules, determine which rules should be run and in what order. DROOLS uses Rete (Forgy Charles, 1982), a matching algorithm that builds a tree from the rules, like a state machine. Facts enter the tree at the top-level nodes as parameters to the rules, and work their way down

the tree if they match the conditions until they reach the leaf nodes: rule consequences.

There are two ways in which rules are executed: forward and backward chaining. Forward chaining is data-driven reasoning; it starts with the available data and uses the rules to extract more data until it has reached its goal. This is opposite to backward chaining that is goal driven, where the system has a goal and uses the rule engine to try to find the evidence to prove it.

Drools is a production rule system, a forward chaining engine where rules have actions in the consequent and are used to generate information based on existing facts.

Syntax-based Rules

The goal of using Drools is that of generating new data about the meaning of a question based on the *facts* delivered by its syntactic analysis through parse trees and dependency relations. These new data correspond to the information that we consider to represent the semantics of the question in terms of question type, expected answer type, focus and keywords. The set of hand-crafted rules designed to meet this goal assumes a well-defined structure of the questions (Figure 22) with fixed positions for the *wh-phrases* relative to the auxiliary and main verb.

Extracting keywords for a given question is the most straightforward method based on the part-of-speech tags generated by the syntactic parser. We consider open class words like nouns, verbs, adjective and adverbs as the meaning-bearing parts of a question and therefore as its constraints to a potential correct answer.

Identifying the type of a question can be regarded as a binary classification problem with two possible values: FACTOID and DEFINITION. We therefore designed a set of rules to only determine definition questions such that any input not triggering them is of factoid type. The rules have been built by inspecting a set of 100 definition questions of earlier CLEF campaigns and implement the following heuristics:

- subject questions with the *wh-word* either *what* or *who*, the main verb *to be*, and the largest constituent following is headed by a proper noun
    - Who is John Lennon? What is BASF?

- subject questions with *what* as *wh-word what*, the main verb *to be*, and the only constituent following it is headed by an indefinite noun
    - What is plastination? What is a meter?
- object questions with *what* as *wh-word*, the main verb to *stand for* or to *mean*
    - What does BASF stand for? What does "Nkosi Sikeleli Afrika" mean?

The question *focus* (Q-FOCUS) represents the property or entity that is being sought by the question and is the piece of information that determines the expected answer type. Generally, the *focus* is determined by the word being modified by the *wh-word*. This heuristic applies only for *wh-words* like *who, what* and *which*. For the rest of wh-words the *focus* is either implied (location for *where*, time for *when*) or immediately following it (*how* questions).

GENERAL CASE (*who, what, which****)****:***
Which US *president* did Francisco Duran try to kill?
What *age* did Elvis Presley die?
To which female *actor* was Arthur Miller married?
Who is the *singer* of the band U2?

Implied focus:
(*where, when,whose, how****):***
Where is the Statue of Liberty located?
When was Franz Kafka born?

Trigger Word Exceptions:
How *much* did BMW pay for Rover in pounds?
How *high* is Mount Everest?
For how *many* ***Oscars*** was the movie Schindler's list nominated?

Figure 24. Focus of factoid questions.

The case of *how* questions is somehow different than those of *where* and *when* questions, since it is not the focus determining the type of the expected answer, but a so-called *trigger* word like *much*, *many*, *far*, etc. It is this *trigger* word that specifies the EA_TYPE (*much* and *far* for MEASURE, *many* for COUNT), while the focus can be either implied by the verb (*pay* calls for currency) or explicitly mentioned (Figure 24).

We therefore build on the dependency relations generated by the syntactic parser in order to find the *focus* of the questions for the general case and on the

syntactic analysis for the rest. Once we have managed to identify either the *focus* or the *trigger* word we make use of an external association table to define the expected answer type for each of the values of those two. We use external knowledge in order to map lexicalized instances to their appropriate types (Figure 25).

```
<entry concept="LOCATION">harbor</entry>
<entry concept="LOCATION">island</entry>
<entry concept="LOCATION">location</entry>

<entry concept="COUNT">many</entry>
<entry concept="COUNT">population</entry>
<entry concept="COUNT">age</entry>

<entry concept="MEASURE">long</entry>
<entry concept="MEASURE">short</entry>
<entry concept="MEASURE">deep</entry>

<entry concept="ORGANIZATION">agency</entry>
<entry concept="ORGANIZATION">Committee</entry>
<entry concept="ORGANIZATION">University</entry>

<entry concept="PERSON">wife</entry>
<entry concept="PERSON">husband</entry>
<entry concept="PERSON">spouse</entry>
```

Figure 25. Possible instances of different EA_Types.

The algorithm used for extracting both the question and the expected answer types, along the focus of a question, can be summarized as following:

1. Find if the input is an open or closed question and exit in the case of the latter.
2. Identify the question as being an object or a subject question.
3. Find out the grammatical *subject* of the question.
4. Determine the Q_TYPE as one of the values: DEFINITION or FACTOID.
5. Identify the *focus* of the question:
    5.1. As the *subject* of DEFINITION questions.
    5.2. For FACTOID questions:
        5.2.1. As the *modifier* of the *wh-word* for the general case.
        5.2.2. Implied meaning for *when*, *where*, *whose*.
6. Analyze the extracted *focus* to determine the real *focus* of the question.
7. Identify the *trigger* word for *how* questions.
8. Look-up the *trigger* word and the *focus* in the external knowledge resource to identify the EA_TYPE.

Step 6 of the algorithm deals with two different cases: incorrect dependency parse of the supplied question and questions asking for names of somebody or something. Since we rely on correct output from the syntactic parser in order to correctly identify the *focus* of a question, errors with this component might result either in a false analysis of the question or no *focus* being generated. In order to cope with these situations we have built-in some fallback rules that try to identify the *focus* based on analysis of phrasal constituents when no modifier for a *wh-word* could be found. The second case covered by this step is that of questions of the following type:

*What is the name of the Danish **capital**?*

where the real *focus* of the question is not *name*, but its dependent through the *of* preposition: *capital*.

The Drools rules used to implement this algorithm are provided in the Annex 2 of this work.

6.3 Evaluation

We evaluate the Question Analysis sub-system by using the questions from our Gold Standard CLEF collection. We have a total of 346 questions of which 53 (15%) are definition and 293 (85%) factoid questions. The result of analyzing a question consists in both control information (question and expected answer type, focus) and keywords. Since the reference test collection contains only data about the question and the expected answer type of a question, we are going to directly test the performance of the Question Analysis components for those. Performance related to the accuracy of correctly extracting the focus and keywords of the questions will be tested later on by factoring the result of the analysis in the Unit Retrieval sub-system and evaluating them as a whole.

We evaluate at this stage the performance of the Question Analysis sub-system in three different settings:

- Monolingual German scenario (**QA_DE**)
- Monolingual English scenario (**QA_EN**)
- Crosslingual English-German scenario (**QA_EN2DE**)

The monolingual scenarios consider input questions as provided by the test collection that are created by human intervention and guaranteed to be well-formed. The cross-lingual scenario uses German questions obtained as result of machine translating original English question by Google Translate. The result of evaluating the Question Analysis sub-system on these three settings can be seen in Table 2.

	Q_TYPE	**EA_TYPE**		**Q_TYPE & EA_TYPE**	
		factoid	*definition*	*factoid*	*definition*
QA_DE	90%	89%	75%	88%	58%
QA_EN2DE	90%	80%	73%	79%	58%
QA_EN	91%	89%	77%	87%	62%

Table 2. Question Analysis Accuracy.

As the results show, both German and English analysis components are comparable in their performance of determining the right questions and expected answer type individually. The results for the cross-lingual scenario (QA_EN2DE) show a substantial drop of about 10% in performance for factoid questions compared to the other configurations. As the Question Analysis component is based on syntactic structure to compute its interpretation, we can infer that the translation process alters this structure in a destructive manner. These results support our assumption (Table 1) that transfer-based methods for translating the information need are better than those based on direct question translation, relative to the syntactic structure.

6.4 Summary

The Question Analysis component is one of the most important components of a QA system since it is responsible for interpreting the meaning of the user request. It is the result of this that drives the strategy of the system in finding the correct answer to a question: retrieving the most relevant passages, extracting candidate answers and selecting the best one based on the constraints imposed by the

question. We have used a high performance analysis component for German and devised a new one with comparable performance for English using a full syntactic parse of the question and creating a system of rules to extract information about question type, focus, expected answer type and content words.

7 Information Unit Retrieval

In general, information retrieval systems construct representations of the documents and the information need and then match those representations to find documents that are most likely to satisfy the need. The Unit Retrieval subsystem considers a *QObj* as its information need and builds a typical IR query constructed from keywords and named entities by using a *Query Generator* (Figure 26).

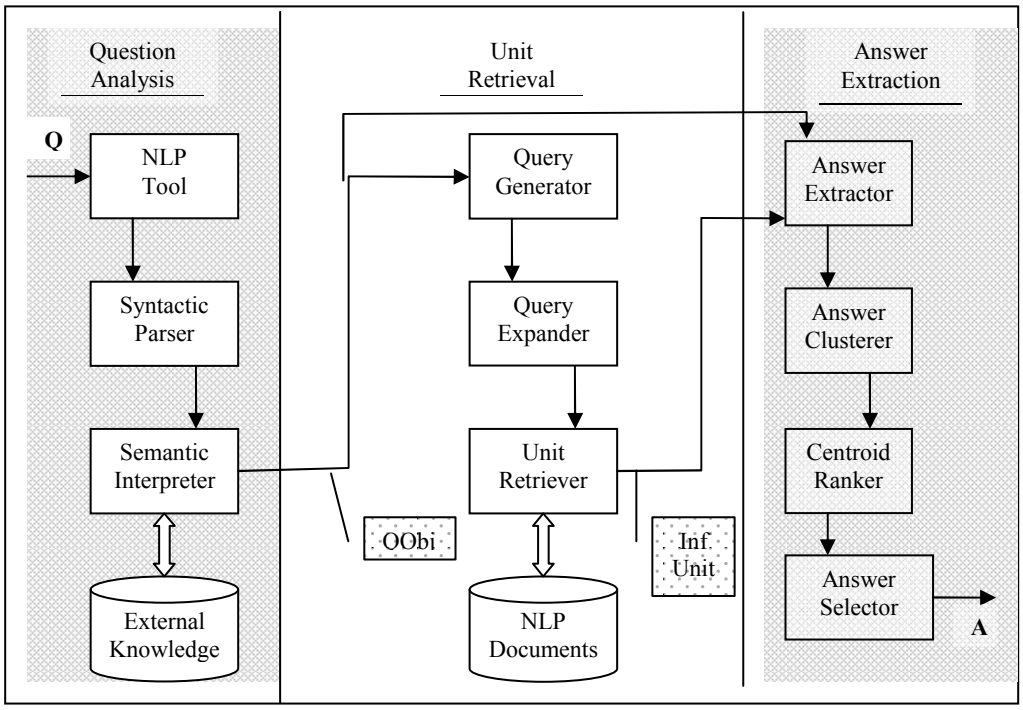


Figure 26. Unit Retrieval Architecture.

In order to cope with the different vocabularies problem that results in part from variability in style and word usage, the *Query Expander* component extends the original query with related words like synonyms and similar concepts. It is this newly generated query as a representation of the information need that guides the Unit Retriever component in finding the most relevant, best match of documents in form of *InfUnit* objects.

As previously mentioned, the match between the information need and relevant documents is based on a comparable representation for both of them. Since the *QObj* contains both keywords and named entities and the expected answer type is a named entity type itself, document content should be represented the same way for maximizing the probability of a best match between them. Toward this goal the document collection has been preprocessed with the same *NLP Tool* as the question during the Question Analysis to annotate named entities and mark the sentence boundaries.

7.1 Text processing

Text processing refers to a set of changes or restructuring techniques that are made to the documents in order to simplify searching. Its main goal is to identify beside traditional words additional terms or features relevant for search.

Identifying additional terms to be used during search to improve ranking can range from extracting noun phrases to leveraging existent markup to recognizing features that have specific semantic content for the application. Of the latter, named entities are very popular for applications like factoid question answering where they refer to concepts of interest in these particular areas. One issue when dealing with named entities is that of anaphora resolution, whose goal is to identify multiple expressions in a document that have the same referent.

Depending on the particular application, the size of the retrieval unit can vary from a whole document to a passage to a sentence. Driven by the specific needs of such an application document processing considers also methods of splitting a document into finer grained units expected to focus better on matching the information need.

In processing the document collection along the lines previously mentioned we have used LingPipe for several reasons: availability of all the required components in one software package; state-of-the-art comparable performance results; easy to extend components and train new named entity models based on annotated corpora.

7.1.1 LingPipe

LingPipe is a state-of-the-art suite of natural language processing tools written in Java that performs tokenization, sentence detection, named entity detection, co-

reference resolution, classification, clustering, part-of-speech tagging, general chunking and fuzzy dictionary matching. Of interest for our goals are the first four technologies for which we will give a short overview.

In LingPipe, sentence boundaries are identified through a heuristic that looks at a token together with the tokens that precede and follow it. If a token is a sentence-final token, then the sentence boundary is the index of the character one past the last character in that token. In order for a token to be a sentence-final token, it must be a member of the set of sentence-final punctuation tokens, such as periods (.) and question marks (?). Furthermore, it must be followed by white space, and the following token (if any) must be a legal start token for a sentence. Sentences containing abbreviations such as "Mr. Smith" are problematic because a simplistic sentence model will treat the period following "Mr." as a sentence-final token. Therefore it is necessary to check the penultimate token in the sentence, and disallow common abbreviations. The heuristic sentence model uses three sets of tokens:

- **Possible Stops**: These are tokens that are allowed to be the final token in a sentence.
- **Impossible Penultimates**: These are tokens that may *not* be the penultimate (second-to-last) token in a sentence. This set is typically made up of abbreviations or acronyms such as "Mr".
- **Impossible Starts**: These are tokens that may *not* be the first token in a sentence. This set typically includes punctuation characters that should be attached to the previous sentence such as end quotes (").

LingPipe's entity extraction is based on a Bayesian generative model that tags each token as being the beginning of a named entity, a continuation of a named entity, or not in a named entity. In its generative model, LingPipe breaks the entire sequence probability down using the chain rule, generating a token/tag pair based on the previous token/tag pairs. History is limited to a finite window of one previous tag and two previous tokens. The chain rule is used again to predict first the tag and then the token given the tag. Maximum likelihood estimates are generated using the labeled data from a training set.

The *standard* model delivered with LingPipe (version 1.7) supplies the following named entity types: PERSON, LOCATION and ORGANIZATION. This can be changed if the named entity detector is retrained.

The co-reference resolution system is based on the CogNIAC system described in the PhD thesis by Breck Baldwin (1995). His thesis is mainly concerned with the resolution of anaphoric expressions, and the underlying theoretical assumption of CogNIAC is that of Centering Theory (Brennan, S. E. et al., 1987, Grosz, B. J. et al., 1995). The core idea is of finding for every new entity or pronoun the best match against already seen mention chains, which are named entities referencing the same concept. The scoring of the best match builds upon several matcher and killer functions, depending on the gender, entity type, substring match, honorific titles and even a user-defined synonym dictionary for named entities.

Moreover these models are genre and language specific such that adaptation to new domains requires retraining the tools.

7.1.2 Preemptive linguistic annotation

LingPipe (version 1.7) comes with out-of-the-box English models both for named entity recognition and sentence boundary detection on the news domain. To fit the requirements of our application we have adapted the delivered code as follows: the named entity model was extended with further types (NUMBER, DATE) and it has been adapted to mark the gender of those entities matching a list of predefined masculine and feminine first names. The models for the German language were completely generated based on training data (named entity recognizer) and a set of manually written rules (sentence boundary), while the co-reference resolution system has been adapted to integrate German pronouns.

With these new tools available, the documents have been processed and following information has been annotated: sentence boundaries, named entities and co-reference (both among entities and pronominal) (Figure 27).

```
<TEXT>
<sent>War <ENAMEX id="1" type="PERSON">Giulio Andreotti</ENAMEX>, Ex-
Ministerpräsident und einflußreichster Politiker im <ENAMEX id="2" type="LOCATION">
Italien</ENAMEX> der Nachkriegszeit, ein Förderer der Mafia?</sent>
<sent>Hat <ENAMEX id="1" type="PERSON">er</ENAMEX> in <ENAMEX id="3"
type="LOCATION">Rom</ENAMEX> die Ermordung eines Journalisten veranlaßt?</sent>
<sent>In <ENAMEX id="4" type="NUMBER">zwei</ENAMEX> Ermittlungsverfahren
sieht sich der <ENAMEX id="5" type="NUMBER">76</ENAMEX> Jahre alte <ENAMEX
id="1" type="PERSON">Andreotti</ENAMEX> dieser Verbrechen beschuldigt; <ENAMEX
id="1" type="PERSON">er</ENAMEX> selbst fühlt sich als Opfer von Intrigen.</sent>
<sent>Doch Mafia-Experten halten die Vorwürfe für &quot;wasserdicht&quot;, außerdem
haben Kronzeugen <ENAMEX id="1" type="PERSON">Andreotti</ENAMEX> schwer
belastet.</sent>
</TEXT>
```

Figure 27. Document annotation with LingPipe.

Co-reference resolution is an important step in interpreting the semantics of a document by explicitly linking entities with the same referent. It is of greater importance when one of the referees is a pronoun, since it improves the coherence of the document by knowing to whom it refers. It is even crucial when parts of the document, like sentences, are taken apart and considered in isolation. In the example of Figure 27, the last two instances would still keep their meaning if considered out of document's context, but the second pronominal reference would become incoherent. Since our application builds on the idea of retrieving sentences as relevant information units rather than documents, the document annotation has to be adapted such that all co-referring entities are substituted by the first most complete of them (Figure 28).

```
<TEXT>
<sent>War <ENAMEX id="1" type="PERSON">Giulio Andreotti</ENAMEX>, Ex-
Ministerpräsident und einflußreichster Politiker im <ENAMEX id="2" type="LOCATION">
Italien</ENAMEX> der Nachkriegszeit, ein Förderer der Mafia?</sent>
<sent>Hat <ENAMEX id="1" type="PERSON">Giulio Andreotti </ENAMEX> in
<ENAMEX id="3" type="LOCATION">Rom</ENAMEX> die Ermordung eines Journalisten
veranlaßt?</sent> <sent>In <ENAMEX id="4" type="NUMBER">zwei</ENAMEX>
Ermittlungsverfahren sieht sich der <ENAMEX id="5" type="NUMBER">76</ENAMEX>
Jahre alte <ENAMEX id="1" type="PERSON">Giulio Andreotti </ENAMEX> dieser
Verbrechen beschuldigt; <ENAMEX id="1" type="PERSON">er</ENAMEX> selbst fühlt
sich als Opfer von Intrigen.</sent>
<sent>Doch Mafia-Experten halten die Vorwürfe für „wasserdicht“, außerdem haben
Kronzeugen <ENAMEX id="1" type="PERSON">Giulio Andreotti </ENAMEX> schwer
belastet.</sent>
</TEXT>
```

Figure 28. Document annotation with LingPipe - revised.

7.2 Search Engine

A search engine is a tool that helps you find what you are looking for faster than if you examined every candidate in a collection in turn. Generally speaking, a search engine consists of two elements: an indexing component and the search software. The indexing component takes care of processing and transforming the documents into a structure that can be looked-up very efficiently, while the search software sifts through the documents recorded in the index to find matches to a search and rank them in order of what it believes is most relevant.

The indexing component scans every document and creates a separate structure, a forward index, as a list of pairs consisting of a document and a word, collated by the document (Table 3).

Document	Words
Document1	scans, every, document, available
Document2	creates, a, separate, structure
Document3	consisting, of, a, document, and, a, word

Table 3. Forward Index Example.

Since querying the forward index would require sequential iteration through each document and to each word to verify a matching document, the index is converted into an inverted index that lists the documents per word (Table 4). The purpose of storing an inverted index is to optimize speed and performance in finding relevant documents for a search query. The index includes additional information such as the frequency of each word in each document or the positions of a word in each document, enabling word proximity searches and relevance ranking supported by word statistics.

Word	Documents
document	Document1, Document3
a	Document2, Document3
structure	Document2
…	…

Table 4. Inverted Index Example.

Index terms represent the content of a document that is used for searching and matching against the information need. General methods for creating consistent index terms are *stopping*, which ignores some words like those with little lexical meaning (functional words), and *stemming*, which reduces different forms of a word that occur because of *inflection* (run, running) or *derivation* (slow, slowness) to a common word form called stem.

Based on the inverted index and its associated data like term statistics, a score function is defined that allows for ranking the documents according to their relevance to a given query. The search software takes care of spotting out relevant documents that match the query according to some criteria and ranks them by computing for each a relevance score.

In the work described here, we have chosen Apache Lucene[5] as the search software for several reasons: it provides an extendable document structure based on index fields that can be individually configured regarding indexing procedure and processing (tokenization, stemming); it allows for custom scoring schemes; and it provides a powerful query language (weighing scheme).

7.2.1 Apache Lucene

Lucene (version 2.3.2) is a Java library that offers two main services: text indexing and text searching. These two activities are relatively independent of each other, although indexing naturally affects searching. The core of the search is the scoring scheme that uses a combination of the Vector Space Model (VSM) and the Boolean model to determine how relevant a given document is to a user's query. It uses the Boolean model to first narrow down the documents that need to be scored based on the use of Boolean operators in the query specification.

Before text is indexed, it is passed through an Analyzer. Analyzers are in charge of extracting indexable tokens out of text to be indexed, and eliminating the rest. Lucene comes with a few different Analyzer implementations. Some of them deal with skipping stop words (frequently used words that don't help distinguish one document from the other, such as *a, an, the, in, on,* etc.), some deal with converting all tokens to lowercase letters (SimpleAnalyzer), so that searches are not case-sensitive, some use suffix stripping algorithms to obtain stems of words (SnowballAnalyzer), and so on (Figure 29).

[5] http://lucene.apache.org/

The quick brown fox jumped over the lazy dogs.

WhitespaceAnalyzer:
[The] [quick] [brown] [fox] [jumped] [over] [the] [lazy] [dogs]

SimpleAnalyzer:
[the] [quick] [brown] [fox] [jumped] [over] [the] [lazy] [dogs]

StopAnalyzer:
[quick] [brown] [fox] [jumped] [over] [lazy] [dogs]

SnowballAnalyzer:
[quick] [brown] [fox] [jump] [over] [lazi] [dog]

Figure 29. Result of different Lucene analyzers.

They are also used when searching. Because the search string has to be processed the same way that the indexed text was processed, it is crucial to use the same Analyzer for both indexing and searching. Not using the same Analyzer will result in invalid search results.

An index consists of a set of *documents*, and each *document* consists of one or more *fields*. Each *field* has a name and a value, whereby a value consists of a sequence of *terms*. A *term* is the smallest piece of a particular *field*. We can think of a document as a row in a relational database and fields as columns in that row.

The score of query *Q* for document *D* correlates to the cosine-distance or dot-product between document and query vectors in a Vector Space Model of Information Retrieval. A document whose vector is closer to the query vector in that model is scored higher. The score is computed as follows:

$$score(Q,D) = coord(Q,D) * queryNorm(Q) * \sum_{t \in Q} tf * idf^2 * t.getBoost() * norm(t,D) \quad (7.1)$$

where:

- ***tf*** correlates to the term's *frequency*, defined as the number of times term *t* appears in the currently scored document D. Documents that have more occurrences of a given term receive a higher score.

- ***idf*** stands for Inverse Document Frequency. This value correlates to the inverse of *docFreq* (the number of documents in which the term *t* appears). This means rarer terms give higher contribution to the total score.

- ***coord(Q, D)*** is a score factor based on how many of the query terms are found in the specified document. Typically, a document that contains more of the query's terms will receive a higher score than another document with fewer query terms.

- ***queryNorm(q)*** is a normalizing factor used to make scores between queries comparable. This factor does not affect document ranking (since all ranked documents are multiplied by the same factor), but rather just attempts to make scores from different queries (or even different indexes) comparable

- ***t.getBoost()*** is a search time boost of term *t* in the query *q* as specified in the query text

- ***norm(t, D)*** encapsulates a few (indexing time) boost and length factors:

    - **Document boost**
    - **Field boost**
    - **lengthNorm**(field) - computed when the document is added to the index in accordance with the number of tokens of this field in the document, so that shorter fields contribute more to the score.

7.2.2 Indexing Sentences

The document collection has been pre-processed by marking sentence boundaries and by annotating named entities along with their pronominal referees (see 7.1). We build on this extracted information and consider the sentence as our retrieval unit, instead of a whole document, and use named entities types and their frequency within a sentence as additional indexing terms. We therefore define a Lucene document to consist of the following fields: a *text* field that indexes the

content of a sentence, a *neTypes* field that indexes the types of the named entities occurring in the sentence and a frequency fields for each of the *neTypes* values (Figure 30), of which the *text* field has been filtered by the SnowballAnalyzer for German before indexing.

The decision to index named entity types along the content of a sentence is based on the fact that in most of the cases answers of factoid questions are instances of these types. By allowing specifying the expected answer type as one of the constraints to be met by question relevant sentences, the result of the retrieval is more focused. Failure to contain a named entity of such a type would render a sentence irrelevant and would be discarded from the result list.

Input Document/Sentence:

<sent>War <ENAMEX id="1" type="PERSON">Giulio Andreotti</ENAMEX>, Ex-Ministerpräsident und einflußreichster Politiker im <ENAMEX id="2" type="LOCATION">Italien</ENAMEX> der Nachkriegszeit, ein Förderer der Mafia?</sent>

Lucene Document:

text:	war giulio andreotti ex ministerprasident einflussreich polit itali nachkriegszeit ford mafia
neTypes:	PERSON LOCATION
PERSON:	1
LOCATION:	1

Figure 30. Lucene Representation of a Sentence.

The frequency fields for each of the named entity types have been considered in order to account for questions of the following type: *What is the capital of Germany?*, when the expected answer type is a LOCATION. Only mentioning *neTypes:LOCATION* as a constraint of a possible relevant document would not be very effective in this case, since the scope of the question, *Germany*, is a LOCATION itself. By specifying that there should be 2 locations (LOCATION:2) mentioned in a possible relevant document we provide a further constraint that enforces the discriminative power of the *neTypes* field.

7.2.3 Scoring Schemes

While the indexing component of a search engine lays the ground for matching a query against the documents, it only provides a binary view of the problem:

documents can be either relevant or not. It is the search software that provides more information on how relevant a document is to a given query by defining a scoring function and ranking the results according to it. Lucene defines such a function in terms of word statistics (document frequency, inverse document frequency), size of overlap between query and document, and length of the document. While this measure is suitable and can be used for the general case of indexing words only, it requires some changes in our case.

Factoid Questions

As previously mentioned, during the document processing all co-referring entities are substituted by the first most complete of them, such that sentences retain their coherence when considered out of their document context (Figure 31). A secondary effect of this transformation is the relative increase of named entity frequency when several references, of which at least one is pronominal, coexist within a sentence.

<TEXT>
<sent>Hat **<ENAMEX id="1" type="PERSON">Giulio Andreotti</ENAMEX>** in <ENAMEX id="3" type="LOCATION">Rom</ENAMEX> die Ermordung eines Journalisten veranlaßt und **<ENAMEX id="1" type="PERSON">Giulio Andreotti</ENAMEX>** Verhaftung dadurch unterschrieben?</sent>
</TEXT>

Figure 31. Out-of-document sentence coherence.

Since the scoring function is dependent on the relative frequency of a search term (tf in formula 7.1) ranking will prefer those results with higher evidence of occurrence for the given term. This means that having a named entity as a search term will affect the ranking of the matched sentences depending on the frequency of pronouns in the original text. In order to cope with this potential issue, the scoring measure was adapted in such a way that the term frequency was assigned with a constant value. While this change might negatively affect the ranking of results for large retrieval units such as documents, given the formula based mainly on statistics of terms, it should remain unnoticed when dealing with smaller units, like sentences, where frequency is one in general.

Definition Questions

The scoring measure implemented by Lucene considers matches in longer fields to be less precise. The *lengthNorm(field)* function is in inverse proportion to the number of tokens within a field such that shorter fields contribute more to the score. This is a good assumption when dealing with factoid questions, but not as effective when looking for definitions. We advance the view that a good definition candidate is the one that provides enough information about the most important attributes of the term to be defined and consequently consider larger sentences having better chances to meet this requirement. Therefore, for answering definition questions we have implemented a change in the scoring measure by making the *lengthNorm(field)* function directly proportional to the number of tokens and generating a separate index to accommodate this change.

7.3 Query Formulation

The *Unit Retrieval* subcomponent is the place where the matching between the information need and the possible answer-bearing documents takes place. At this stage the question has been interpreted and reduced to a structured representation, the *QObj* that captures the semantics of the request in terms of question and expected answer types, focus, named entities and keywords. On the other side, the documents have been processed and indexed for quick access and can be searched for using Lucene's query language. What we need is a way of converting the information seized in the *QObj* into a well-formed IR query, based on which question relevant documents could be retrieved.

7.3.1 Query Generation

The Query Generation component assigns the information from the *QObj* to the appropriate fields of the indexed documents and takes notice of the named entity type frequencies when multiple instances of the same type are likely to appear in the results (Figure 32). When building the IR-query, information about part-of-speech is considered to decide the salience of a term, such that only nouns, adverbs and adjectives are required to appear, while verbs are optional. The expected answer type is mapped to the *neTypes* field and is a required attribute of matching documents. Named entities are also mandatory to appear in relevant

documents, while the focus of the question is more important than the rest of the terms and accordingly boosted in the query.

The focus of a question represents a feature of the expected answer and can either appear in the answer-bearing document, as *Hauptstadt* does in the above example, or be implied by the semantics of the answer (i.e., *Which country do the Galápagos Islands belong to?*). To cover the cases when the focus is implied by the answer-bearing documents, the IR-query can be automatically relaxed by making the focus optional when no results are retrieved that explicitly mention it.

QObj:

```
<QOBJ score="1" msg="quest" lang="DE" id="qId0">
  <NL-STRING id="qId0">
    <SOURCE lang="DE">Wie heißt die Hauptstadt von Deutschland ?</SOURCE>
  </NL-STRING>
  <QA-control>
    <Q-FOCUS>Hauptstadt</Q-FOCUS>
    <Q-TOPIC>Deutschland</Q-TOPIC>
    <Q-TYPE restriction="NONE">FACTOID</Q-TYPE>
    <A-TYPE type="atomic">LOCATION</A-TYPE>
  </QA-control>
  <KEYWORDS>
    <KEYWORD type="UNIQUE" id="kw0">
      <TK stem="heiss" pos="V">heißt</TK>
    </KEYWORD>
    <KEYWORD type="UNIQUE" id="kw1">
      <TK stem="hauptstadt" pos="N">Hauptstadt</TK>
    </KEYWORD>
  </KEYWORDS>
  <EXPANDED-KEYWORDS/>
  <NE-LIST>
    <NE type="LOCATION" id="ne0">Deutschland</NE>
  </NE-LIST>
</QOBJ>
```

IR-Query: +text:hauptstadt +text:deutschland^4 text:heiss +neTypes:LOCATION +LOCATION:2

Document: Seit der Wiedervereinigung am 3. Oktober 1990 ist Berlin auch Hauptstadt von Deutschland.

Lucene Document:

text: wiederverein 3 oktob 1990 ist berlin hauptstadt deutschland
neTypes: LOCATION DATE
DATE: 1
LOCATION: 1 2

Figure 32. Query Generator Data.

7.3.2 Query Extension

The rough idea of automatically finding relevant documents to a given query is based on measuring the matching degree of a document representation against that of the information need. In the IR context, *indexing* is the process of developing a document representation by assigning content descriptors or terms to the document. These terms are used in assessing the relevance of a document to a user query and contribute directly to the retrieval effectiveness of an IR system. Since they are intended to reflect the information manifested in the document, these are also known as *content terms*.

In most IR models content terms are words that literally occur in the document and therefore are directly related to the lexical representation of the information rather than to its semantics. This becomes quickly an issue when vocabularies used for expressing the information need and those of the document collection are different. Searches for information related to words like *court* and *suit* will not match any documents with content terms such as *tribunal* and *lawsuit*, resulting in a lower recall and therefore possible lower performance.

One way of dealing with this kind of problem is by making use of external lexical resources, either task specific or general purpose, that provide semantically related concepts and their lexical realization. For this purpose we employ GermaNet, a general purpose lexical database for German, and a manually generated association list of nation related terms.

GermaNet

GermaNet (Hamp, B**.** & Feldweg, H., 1997) is a broad-coverage lexical-semantic net that relates German nouns**,** verbs**,** and adjectives semantically by grouping lexical units expressing the same concept into so-called *synsets* and by defining semantic relations between these synsets. Lemmas are the lexical units of the net, assuming that inflected forms are mapped to base forms by an external morphological analyzer. Two basic types of relations can be distinguished:

- lexical relations (i.e. *synonymy*, *antonymy* and *pertains to*), which hold between different lexical realizations of concepts, and
- conceptual relations (i.e. *hypernymy*, *hyponymy*, *meronymy*, etc.), which hold between different concepts in all their particular realizations.

The basic framework of GermaNet is similar to the Princeton WordNet (Miller et al., 1993) and it has been built from scratch rather than translated from its English counterpart. It currently contains about 58,000 synsets with almost 82,000 lexical units, of which approximately 41,000 are nouns, 11,000 are verbs and 6,000 are adjectives.

GermaNet aims at modeling at least the base vocabulary of German and it is primarily intended to serve as a resource for word sense disambiguation, which is crucial for natural language applications like information retrieval.

Task Specific Resources

As a task specific resource of lexical knowledge we have automatically created from online available data a 205-entries list of nation related terms that associate a nation (*France*) to its people (*Frenchman*, *Frenchwomen*) and to concepts pertaining to it or its people (*French*). By doing this we try to reveal variations in language for expressing the concept of nationality, variations that are not covered by the other type of lexical resource.

Lexical vs. Conceptual Extension

Given the likely different vocabularies of the information request and of the document collection it is hard to predict the most appropriate method to abridge the lexical gap between them. In most of the cases, enriching the IR query with synonyms for the question keywords will probably suffice, but there are still cases when the information need uses narrower or broader semantic concepts to either inquire for specific details or more general facts. In order to cope with these cases we need an additional expansion by narrower (hyponyms) and broader (hypernyms) concepts than those explicitly captured by the request, for relevant documents to be matched.

7.4 Evaluation

For evaluation of the Information Unit Retrieval component, a set of 293 factoid questions and 53 definition questions from the CLEF collections of the years 2007 and 2008 have been considered. The effect of varying the unit retrieval size, extending the query with synonyms and related concepts and using different

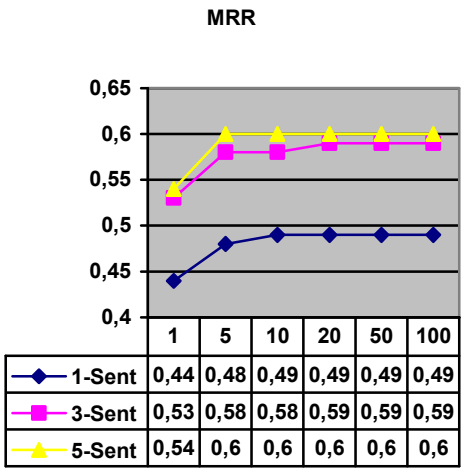
translation approaches of the question have been investigated, under the assumption that only questions that passed the Q_TYPE test of the Question Analysis component count for performance testing (see Table 2. Question Analysis Accuracy.). The measures used for the evaluation are Mean Reciprocal Rank (MRR), which should give us a figure about the ranking of the first relevant match, and Mean Average Precision (MAP) that describes the overall precision and distribution of relevant matches.

7.4.1 Monolingual Experiments

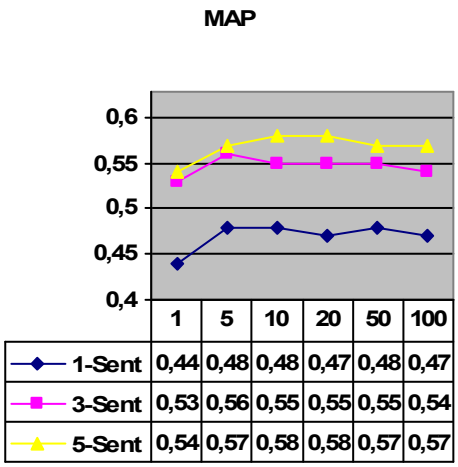
The goal of the monolingual experiments was twofold: to investigate the effect of varying the retrieval unit size (1-sentence, 3-sentences and 5-sentences) on the performance of the component and to assess the use of query extension techniques.

The results of the evaluation reveal two things: the document retrieval has a good accuracy in finding relevant units of information in the top 10 matches (MRR figure) and while the majority lies within these limits there are still some relevant units down the list of ranked results (MAP figure). On the MAP figure we can see that after a rising of the curve in the top 10 units the measure has a decreasing tendency, which points to the existence of some relevant units in this range as well.

Moreover, we can observe quite a substantial increase in performance with larger sizes of unit retrieval, though the most relevant is the one between 1-sentence and 3-sentences retrieved. It is to be expected that increasing the size of the retrieval unit will yield better results, but the difference between 1-sent and 3-sent runs was impressive. A closer look at the potential causes for this surprising improvement revealed two things: first, about 10% of the factoid questions in the Gold Standard assumed a unit size of length 3 in order to answer the question and second, the sentence boundary detection module failed to correctly detect sentences ending with a newline (\n), a common practice in the news corpus considered.



	1	5	10	20	50	100
1-Sent	0,44	0,48	0,49	0,49	0,49	0,49
3-Sent	0,53	0,58	0,58	0,59	0,59	0,59
5-Sent	0,54	0,6	0,6	0,6	0,6	0,6

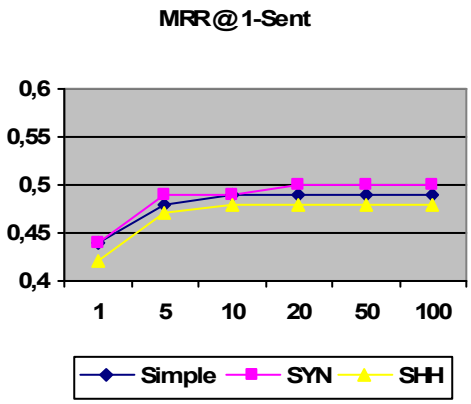


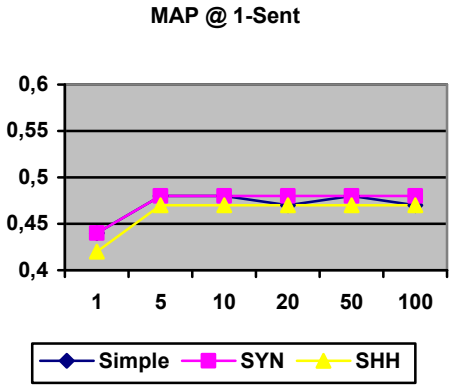
	1	5	10	20	50	100
1-Sent	0,44	0,48	0,48	0,47	0,48	0,47
3-Sent	0,53	0,56	0,55	0,55	0,55	0,54
5-Sent	0,54	0,57	0,58	0,58	0,57	0,57

The query expansion techniques considered were both at the lexical level, by using synonyms (SYN runs), and at the conceptual level employing a combination of synonyms, hypernyms and hyponyms (SHH runs). No method of word sense disambiguation has been used before extending the keywords, relying on the assumption that small contexts provided by the unit retrieval will support the *one sense per collocation* property of human languages, according to which words tend to exhibit only one sense in a given collocation.

The results of evaluating the query expansion over retrieval units of 1-sentence length have showed a slight improvement in ranking the relevant units by way of using synonyms. The expansion at conceptual level though did not bring any improvements, but slightly decreased the performance of the component.

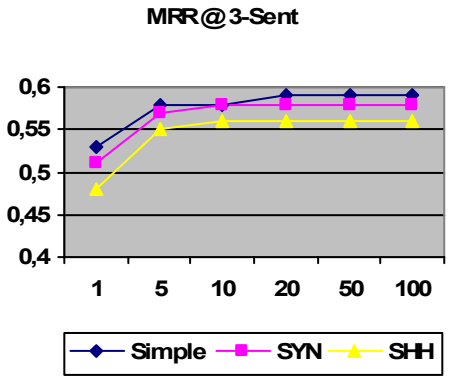
1-Sent	Top 1		Top 5		Top 10		Top 20		Top 50		Top 100	
	MRR	**MAP**	**MRR**	**MAP**	**MRR**	**MAP**	**MRR**	**MAP**	**MRR**	**MAP**	**MRR**	**MAP**
Simple	0,44	0,44	0,48	0,48	0,49	0,48	0,49	0,47	0,49	0,48	0,49	0,47
SYN	0,44	0,44	0,49	0,48	0,49	0,48	0,50	0,48	0,50	0,48	0,50	0,48
SHH	0,42	0,42	0,47	0,47	0,48	0,47	0,48	0,47	0,48	0,47	0,48	0,47

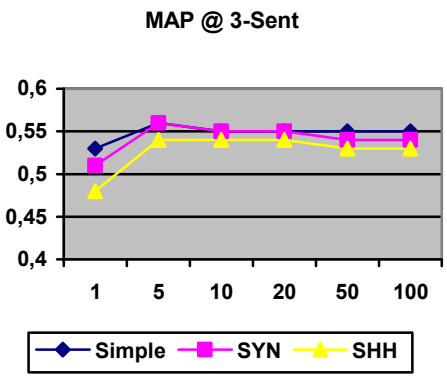




For retrieval units of 3-sentences length, the query expansion did not manage to improve the results, but rather decreased the performance of the component. Moreover, the falling slope of the MAP curve for SYN and SHH runs shows that relevant matches have been even pushed down the ranking list.

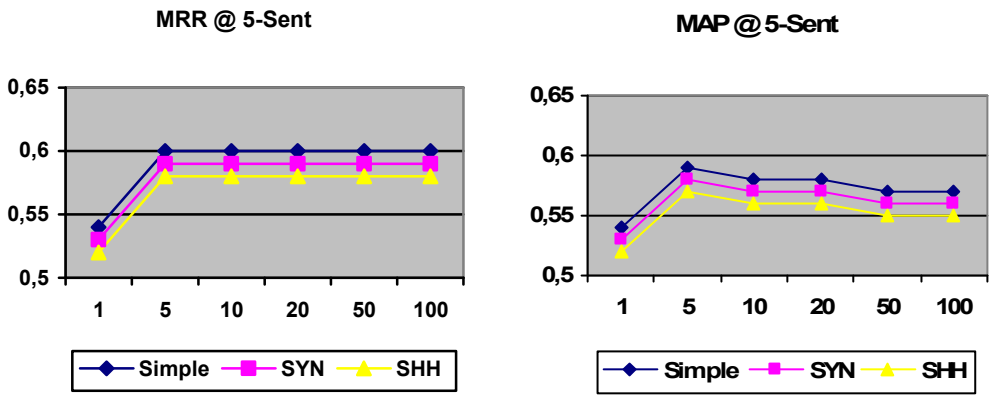
3-Sent	Top 1		Top 5		Top 10		Top 20		Top 50		Top 100	
	MRR	**MAP**	**MRR**	**MAP**	**MRR**	**MAP**	**MRR**	**MAP**	**MRR**	**MAP**	**MRR**	**MAP**
Simple	0,53	0,53	0,58	0,56	0,58	0,55	0,59	0,55	0,59	0,55	0,59	0,55
SYN	0,51	0,51	0,57	0,56	0,58	0,55	0,58	0,55	0,58	0,54	0,58	0,54
SHH	0,48	0,48	0,55	0,54	0,56	0,54	0,56	0,54	0,56	0,53	0,56	0,53





For retrieval units consisting of 5 adjacent sentences, the performance of the component using query expansion dropped down again.

5-Sent	Top 1		Top 5		Top 10		Top 20		Top 50		Top 100	
	MRR	**MAP**	**MRR**	**MAP**	**MRR**	**MAP**	**MRR**	**MAP**	**MRR**	**MAP**	**MRR**	**MAP**
Simple	0,54	0,54	0,6	0,59	0,6	0,58	0,6	0,58	0,6	0,57	0,6	0,57
SYN	0,53	0,53	0,59	0,58	0,59	0,57	0,59	0,57	0,59	0,56	0,59	0,56
SHH	0,52	0,52	0,58	0,57	0,58	0,56	0,58	0,56	0,58	0,55	0,58	0,55



These results show that using query expansion might slightly improve performance of a retrieval unit component, even without doing any word sense disambiguation, though using a window of 1-sentence for collocated concepts is essential.

One important outcome of evaluating the monolingual Unit Retrieval component is that there are still some relevant matches down the ranking list, between position 20 and 100, as pictured by the MAP measure. Since the idea of our Answer Selection and Extraction component builds upon the assumption that redundant data is a good indicator for its suitability as a potential answer, we need to make sure that we do not constrain the list of relevant units to higher ranks (i.e. top 10 or 20).

7.4.2 Cross-lingual Experiments

The goal of the cross-lingual experiments was to assess the performance of the retrieval component in the view of using different techniques for crossing the language barrier by question translation. The following component configurations have been defined for empirical comparison:

MT Google: English question translated into German by Google and the result analyzed

Align Google: English question analyzed and the result mapped into German using the alignment table of English-German Google translation

MRD: English question analyzed and the result mapped into German using MRD (machine readable dictionaries)

In order to tackle the potential issues of ambiguity associated with using machine readable dictionaries for translating individual question words, the following configurations have been considered as well:

MRD + PoS: MRD + part-of-speech filtered translations
MRD + PoS + MI: MRD + part-of-speech + mutual information filtered translations

The first configuration uses part-of-speech to filter only those translations sharing common information, and the second one further filters the list of accepted translation by using Mutual Information that measures the mutual dependence of two words over the corpus of data. Since 90% of the questions contain named entities, which by their nature are not ambiguous, we have considered the mutual information between the translation of a named entity and translations of other question keywords as a measure of selecting only those reciprocal dependent.

The result of evaluating these configurations has revealed the following facts:

- Alignment techniques are better than both machine translation and machine readable dictionary approaches.
- Machine readable dictionary techniques are better than machine translation for lower size retrieval units (1, 3) and comparable for higher unit sizes (5) as measured by MRR figures.
- Use of part-of-speech and Mutual Information filtering methods for translation by way of machine readable dictionaries is not consistently improving the performance.

According to these results we can recast the values in (Table 1) for crossing the language barrier by way of question translation into following:

	Syntactic Structure	Word Translation
Direct Translation	o	o
Method 2	+	o
Method 1	+	-

Table 5. Comparison of question translation methods (revised).

Evaluation has rendered the alignment method 2 better than the MRD method 1 and this can be explained by considering word translation responsible for that, since both methods use the same syntactic structure as starting point. This temporarily invalidates our assumption that local context automatically sorts out irrelevant meanings of collocated word translations. A closer look at the results shows the MRD-based method performing almost as good as the alignment-based one for a retrieval unit of 1-sentence length. The question that arises at this point is what local context of collocated meanings is and if a better specification of it could reinstall the true value of our assumption. We postpone this discussion to a later point in this work (chapter 9) when additional evidence would shed light on it.

The prevalence of the MRD-based method 1 over direct translation is due to the better syntactic structure of the question, which for smaller retrieval unit sizes seems to overcome the disadvantage brought in by the ambiguity of the translated words.

The effect of query expansion has been evaluated for the cross-language scenario as well. The monolingual evaluation showed that expansion makes sense only when considering retrieval units of 1-sentence length. The cross-language evaluation confirms this result, but only for translating questions by way of alignment. This result supports our assumption of one sense per collocation in small contexts that we have made for the query extension and somehow contradicts the findings above for the MRD method. A closer look at these assumptions discloses two different settings: the monolingual one, for German only, and the cross-lingual one for translations from English to German. A viable explanation for the contradictory results is the higher rate of polysemy for English compared to German.

1-Sent	Top 1		Top 5		Top 10		Top 20		Top 50		Top 100	
	MRR	**MAP**	**MRR**	**MAP**	**MRR**	**MAP**	**MRR**	**MAP**	**MRR**	**MAP**	**MRR**	**MAP**
MT Google	0,29	0,29	0,34	0,34	0,35	0,34	0,35	0,35	0,36	0,35	0,36	0,35
Align Google	0,32	0,32	0,38	0,37	0,39	0,37	0,39	0,38	0,39	0,39	0,39	0,39
MRD	0,32	0,32	0,38	0,36	0,38	0,36	0,38	0,36	0,38	0,36	0,38	0,36
MRD + PoS	0,32	0,32	0,37	0,36	0,37	0,36	0,38	0,36	0,38	0,36	0,38	0,36
MRD + PoS + MI	0,29	0,29	0,34	0,34	0,35	0,34	0,35	0,35	0,36	0,35	0,36	0,35

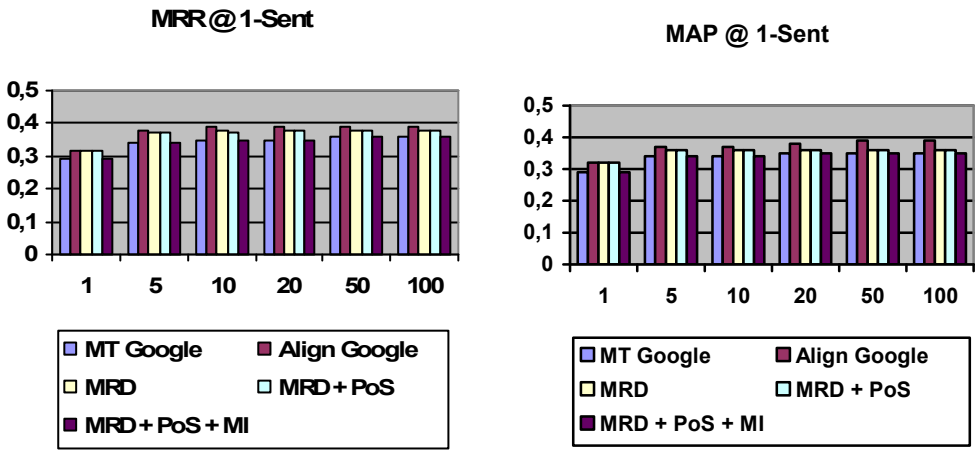


Figure 33. Comparison of different techniques of cross-linguality for retrieval units of 1-sentence length.

3-Sent	Top 1		Top 5		Top 10		Top 20		Top 50		Top 100	
	MRR	**MAP**	**MRR**	**MAP**	**MRR**	**MAP**	**MRR**	**MAP**	**MRR**	**MAP**	**MRR**	**MAP**
MT Google	0,35	0,35	0,42	0,41	0,42	0,41	0,43	0,41	0,43	0,4	0,43	0,4
Align Google	0,4	0,4	0,45	0,44	0,46	0,45	0,46	0,44	0,46	0,43	0,46	0,43
MRD	0,38	0,38	0,43	0,42	0,44	0,41	0,44	0,39	0,44	0,37	0,44	0,37
MRD + PoS	0,37	0,37	0,43	0,42	0,43	0,41	0,43	0,39	0,43	0,37	0,43	0,37
MRD + PoS + MI	0,38	0,38	0,43	0,42	0,43	0,41	0,43	0,39	0,44	0,38	0,44	0,38

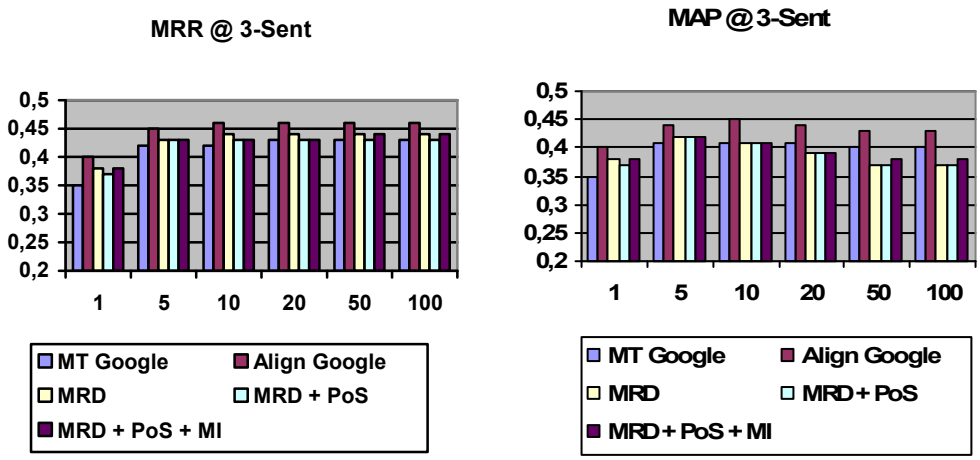


Figure 34. Comparison of different techniques of cross-linguality for retrieval units of 3-sentences length.

5-Sent	Top 1		Top 5		Top 10		Top 20		Top 50		Top 100	
	MRR	**MAP**	**MRR**	**MAP**	**MRR**	**MAP**	**MRR**	**MAP**	**MRR**	**MAP**	**MRR**	**MAP**
MT Google	0,38	0,38	0,43	0,43	0,44	0,42	0,44	0,42	0,44	0,42	0,44	0,42
Align Google	0,44	0,44	0,49	0,48	0,5	0,47	0,5	0,47	0,5	0,46	0,5	0,46
MRD	0,38	0,38	0,43	0,41	0,44	0,4	0,44	0,4	0,44	0,38	0,44	0,38
MRD + PoS	0,39	0,39	0,42	0,4	0,43	0,39	0,43	0,39	0,43	0,38	0,43	0,37
MRD + PoS + MI	0,37	0,37	0,41	0,38	0,41	0,38	0,42	0,38	0,42	0,37	0,42	0,37

Figure 35. Comparison of different techniques of cross-linguality for retrieval units of 5-sentences length.

1-Sent	Top 1		Top 5		Top 10		Top 20		Top 50		Top 100	
	MRR	MAP	MRR	MAP	MRR	MAP	MRR	MAP	MRR	MAP	MRR	MAP
MT Google	0,29	0,29	0,34	0,34	0,35	0,34	0,35	0,35	0,36	0,35	0,36	0,35
MT Google + SYN	0,29	0,29	0,34	0,34	0,35	0,34	0,35	0,35	0,35	0,34	0,35	0,34
MT Google + SHH	0,28	0,28	0,34	0,34	0,35	0,34	0,35	0,34	0,35	0,34	0,35	0,34

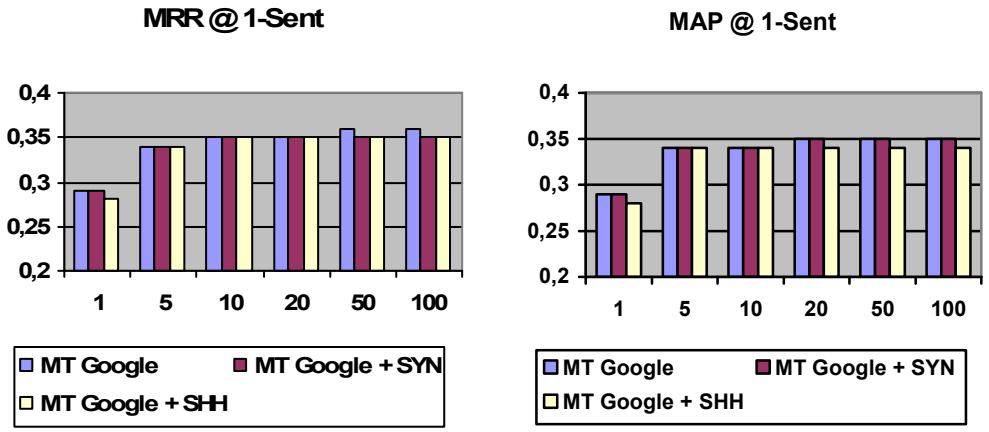


Figure 36. Results of lexical and conceptual query extension for direct translation and retrieval units of 1-sentence length.

1-Sent	Top 1		Top 5		Top 10		Top 20		Top 50		Top 100	
	MRR	MAP	MRR	MAP	MRR	MAP	MRR	MAP	MRR	MAP	MRR	MAP
Align	0,32	0,32	0,38	0,37	0,39	0,37	0,39	0,38	0,39	0,39	0,39	0,39
Align + SYN	0,34	0,34	0,38	0,38	0,39	0,38	0,39	0,38	0,40	0,38	0,40	0,38
Align + SHH	0,31	0,31	0,36	0,35	0,37	0,36	0,37	0,36	0,37	0,36	0,37	0,37

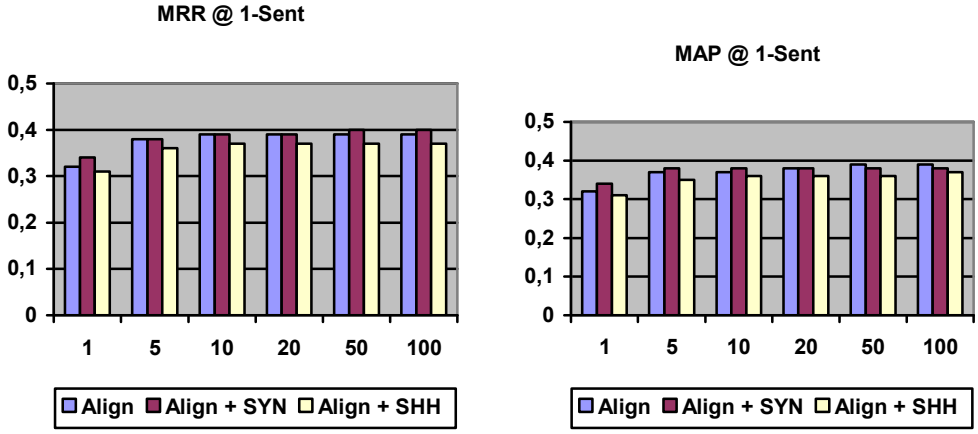


Figure 37. Results of lexical and conceptual query extension for transfer-based translation by alignment and retrieval units of 1-sentence length.

1-Sent	Top 1		Top 5		Top 10		Top 20		Top 50		Top 100	
	MRR	**MAP**	**MRR**	**MAP**	**MRR**	**MAP**	**MRR**	**MAP**	**MRR**	**MAP**	**MRR**	**MAP**
MRD	0,32	0,32	0,37	0,36	0,38	0,36	0,38	0,36	0,38	0,36	0,38	0,36
MRD + SYN	0,3	0,3	0,36	0,35	0,37	0,36	0,37	0,36	0,37	0,35	0,37	0,35
MRD + SHH	0,28	0,28	0,34	0,34	0,35	0,35	0,36	0,35	0,36	0,35	0,36	0,35

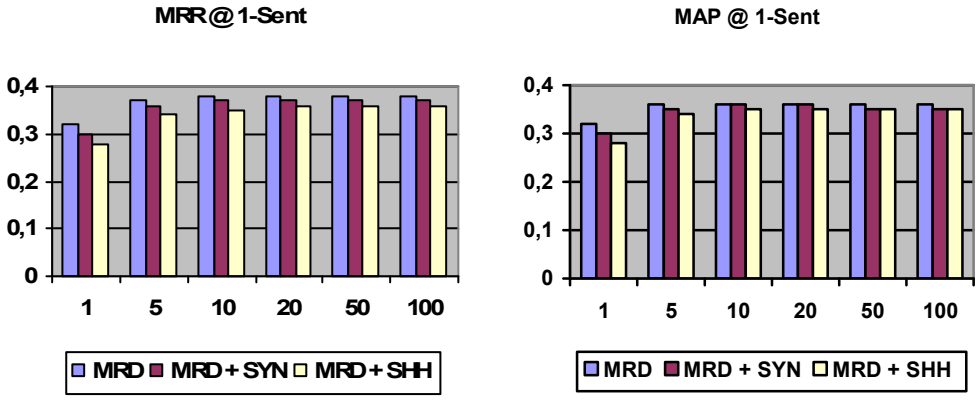


Figure 38. Results of lexical and conceptual query extension for transfer-based translation by MRD and retrieval units of 1-sentence length.

3-Sent	Top 1		Top 5		Top 10		Top 20		Top 50		Top 100	
	MRR	**MAP**	**MRR**	**MAP**	**MRR**	**MAP**	**MRR**	**MAP**	**MRR**	**MAP**	**MRR**	**MAP**
MT Google	0,36	0,36	0,42	0,42	0,43	0,42	0,43	0,42	0,43	0,4	0,43	0,41
MT Google + SYN	0,35	0,35	0,42	0,41	0,42	0,41	0,43	0,41	0,43	0,4	0,43	0,4
MT Google + SHH	0,35	0,35	0,42	0,41	0,42	0,41	0,43	0,40	0,43	0,39	0,43	0,39

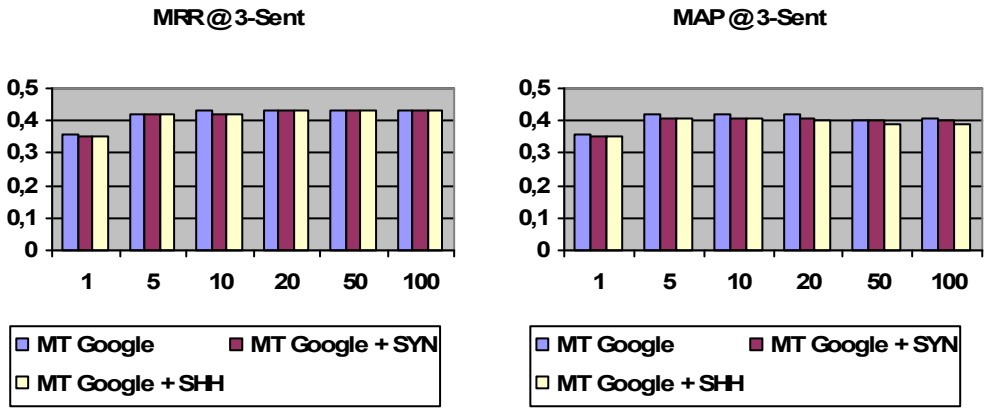


Figure 39. Results of lexical and conceptual query extension for direct translation and retrieval units of 3-sentences length.

3-Sent	Top 1		Top 5		Top 10		Top 20		Top 50		Top 100	
	MRR	**MAP**	**MRR**	**MAP**	**MRR**	**MAP**	**MRR**	**MAP**	**MRR**	**MAP**	**MRR**	**MAP**
Align	0,4	0,4	0,45	0,44	0,46	0,45	0,46	0,44	0,46	0,43	0,46	0,43
Align + SYN	0,39	0,39	0,44	0,43	0,45	0,43	0,45	0,43	0,45	0,42	0,45	0,43
Align + SHH	0,34	0,34	0,4	0,4	0,41	0,4	0,41	0,39	0,41	0,39	0,41	0,4

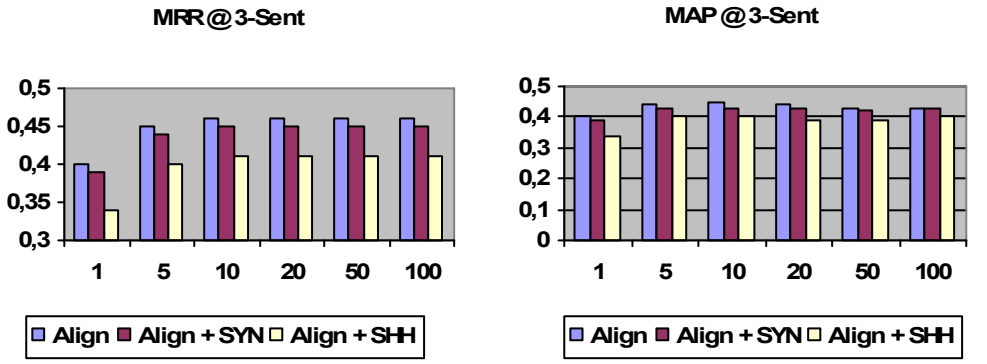


Figure 40. Results of lexical and conceptual query extension for transfer-based translation by alignment and retrieval units of 3-sentences length.

3-Sent	Top 1		Top 5		Top 10		Top 20		Top 50		Top 100	
	MRR	**MAP**	**MRR**	**MAP**	**MRR**	**MAP**	**MRR**	**MAP**	**MRR**	**MAP**	**MRR**	**MAP**
MRD	0,38	0,38	0,43	0,42	0,44	0,41	0,44	0,39	0,44	0,37	0,44	0,37
MRD + SYN	0,34	0,34	0,4	0,39	0,41	0,38	0,41	0,37	0,41	0,36	0,41	0,36
MRD + SHH	0,33	0,33	0,38	0,36	0,38	0,36	0,39	0,36	0,39	0,35	0,39	0,35

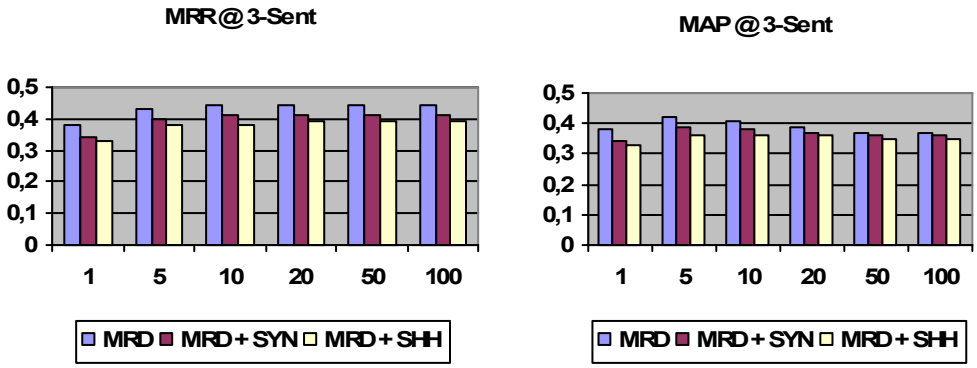


Figure 41. Results of lexical and conceptual query extension for transfer-based translation by MRD and retrieval units of 3-sentences length.

5-Sent	Top 1		Top 5		Top 10		Top 20		Top 50		Top 100	
	MRR	**MAP**	**MRR**	**MAP**	**MRR**	**MAP**	**MRR**	**MAP**	**MRR**	**MAP**	**MRR**	**MAP**
MT Google	0,38	0,38	0,42	0,42	0,43	0,42	0,44	0,42	0,44	0,42	0,44	0,41
MT Google + SYN	0,39	0,39	0,43	0,42	0,44	0,42	0,44	0,41	0,44	0,41	0,44	0,41
MT Google + SHH	0,38	0,38	0,43	0,42	0,44	0,42	0,44	0,42	0,44	0,42	0,44	0,41

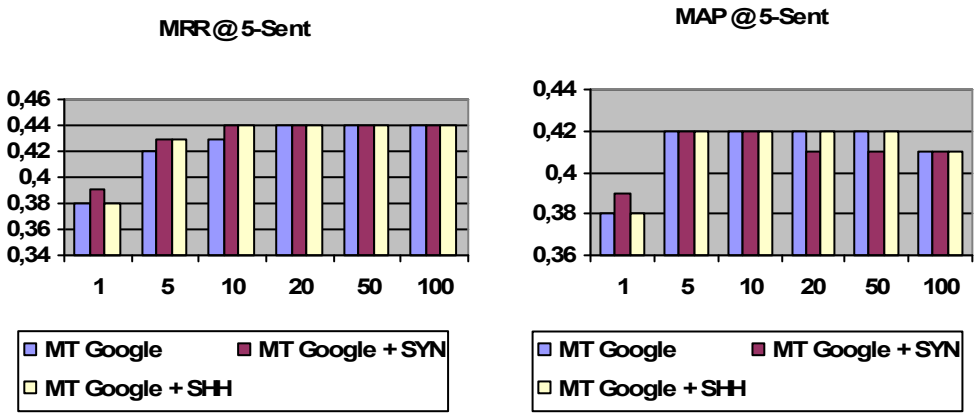


Figure 42. Results of lexical and conceptual query extension for direct translation and retrieval units of 5-sentences length.

5-Sent	Top 1		Top 5		Top 10		Top 20		Top 50		Top 100	
	MRR	**MAP**	**MRR**	**MAP**	**MRR**	**MAP**	**MRR**	**MAP**	**MRR**	**MAP**	**MRR**	**MAP**
Align	0,44	0,44	0,49	0,48	0,5	0,47	0,5	0,46	0,5	0,46	0,5	0,46
Align + SYN	0,43	0,43	0,48	0,47	0,49	0,46	0,49	0,45	0,5	0,45	0,5	0,45
Align + SHH	0,38	0,38	0,43	0,42	0,44	0,42	0,45	0,42	0,45	0,42	0,45	0,42

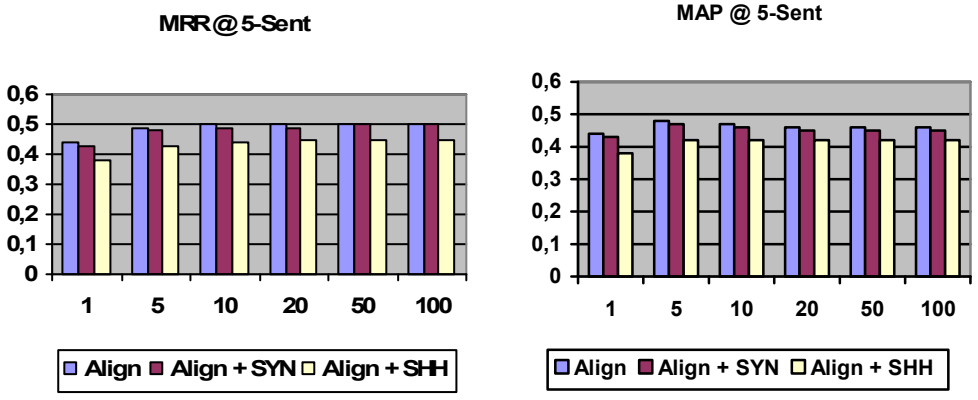


Figure 43. Results of lexical and conceptual query extension for transfer-based translation by alignment and retrieval units of 5-sentences length.

5-Sent	Top 1		Top 5		Top 10		Top 20		Top 50		Top 100	
	MRR	**MAP**	**MRR**	**MAP**	**MRR**	**MAP**	**MRR**	**MAP**	**MRR**	**MAP**	**MRR**	**MAP**
MRD	0,38	0,38	0,43	0,41	0,43	0,4	0,44	0,4	0,44	0,38	0,44	0,38
MRD + SYN	0,36	0,36	0,41	0,4	0,42	0,38	0,42	0,38	0,42	0,37	0,42	0,36
MRD + SHH	0,32	0,32	0,37	0,37	0,38	0,36	0,38	0,36	0,39	0,34	0,39	0,34

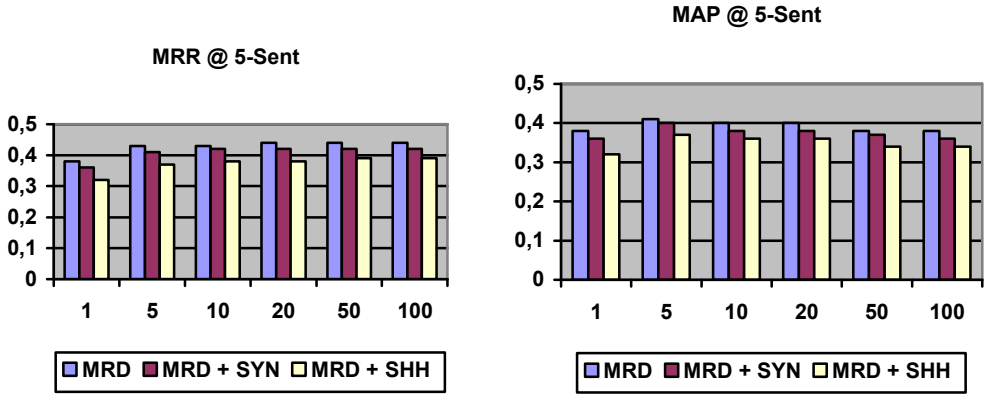


Figure 44. Results of lexical and conceptual query extension for transfer-based translation by MRD and retrieval units of 5-sentences length.

7.5 Summary

We have presented a Unit Retrieval component that centers on the idea that small retrieval units are sufficient for finding relevant information. A potential lexical gap between the question and the document collection is handled by expanding the question with related lexical items, a method that leverages the small sized context of the retrieval units to inherently select the intended meaning of an ambiguous word. In a cross-lingual scenario, analyzing the question upfront and translating the result outperforms methods of direct question translation. The prevalence of the MRD-based method over direct translation is due to the better syntactic structure of the question, which for smaller retrieval unit sizes seems to overcome the disadvantage brought in by the ambiguity of the translated words.

8 Answer Extraction

The answer extraction component of a question answering system is one of the most critical but also one of the most difficult stages in the process of finding exact correct answers to questions. Given a piece of text (e.g. document, passage, sentence), an answer extractor identifies candidate answers and makes a decision whether each candidate is a correct answer or not (Figure 45).The answer extractors of most question answering systems compute scores based on their content and structure, as well as on the content and structure of the corresponding textual contexts.

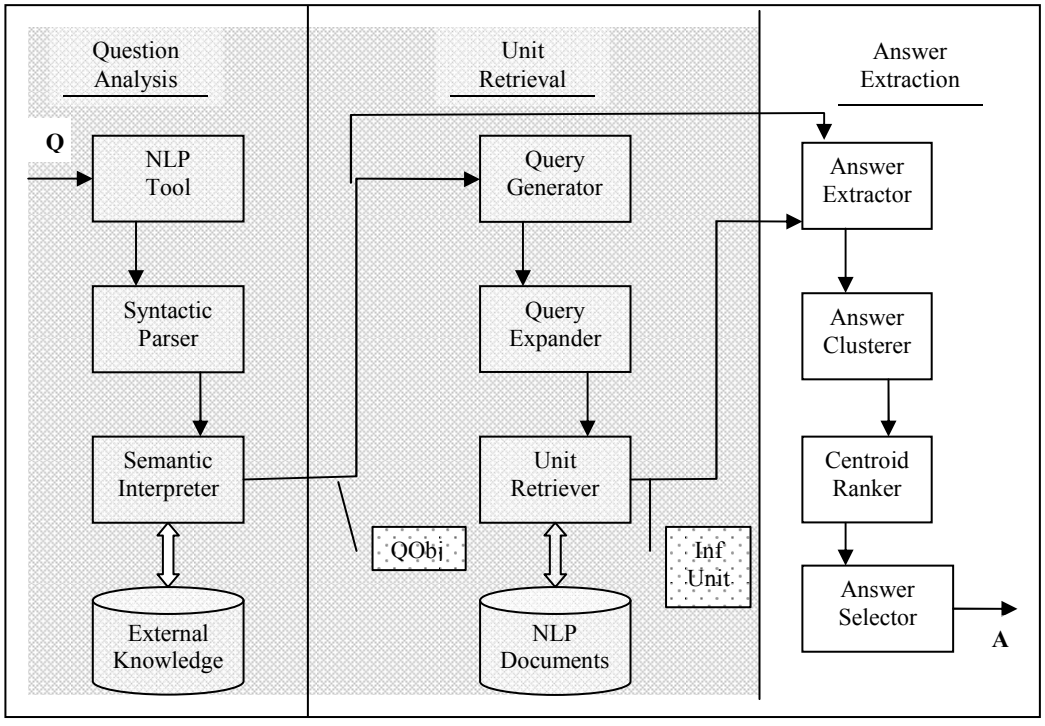


Figure 45. Answer Extraction Architecture.

The performance of an answer extraction component is intertwined with the performance of the retrieval component of a QA system. If the retrieved documents are not relevant, the answer extractor becomes insignificant since the overall performance will certainly be low. However, if the retrieved documents

are all relevant, but the structure of the text is too complex, the correct answers also cannot be extracted and the performance of the retrieval component is irrelevant. Hence, the goal is to find a retrieval-extraction strategy that yields the best performance for a particular QA system.

A good trade-off between retrieving many relevant documents and having a content structure not too complex but sufficient for extracting possible correct answers is to consider the size of a document in terms of a small number of adjacent sentences. As Unit Retrieval experiments have shown, both a 1-sentence and a 3-sentences document length offer enough relevant information, while keeping structural complexity low. The *Answer Extraction* sub-system builds on two presumptions: that *redundancy* is a good indicator of answer suitability and *proximity* a good approximation of conceptual relatedness. *Redundancy* will be used as a fitness criterion for answer candidates, with more frequent answers being considered more suitable, and *proximity* will deliver the means by which the relationship between possible correct answers and question concepts is measured. In other words, we will consider answer candidates frequently co-occurring with question keywords and in their immediate vicinity as a good educated guess for answers being correct.

The *Answer Extractor* component collects all instances of a specific EAType (expected answer type) from the relevant *InfUnit*s as likely answer candidates and passes the result over to the *Answer Clusterer*, which groups them together based on common referred entities (“John B. Doe” ~ “J. B. Doe” ~ “John Doe” ~ “Doe”). The *Centroid Ranker* component assigns relevance scores to these newly formed answer clusters based on statistics of occurrence over different sentences and documents and the *Answer Selector* scores the most representative instances of the best ranked clusters based on a *proximity* measure defined over the possible answer and the question keywords.

Up to the *Answer Selector* component, this workflow is valid when looking for answers of both factoid and definition questions. Since the answer type for definition questions varies from words to phrases to whole sentences, we use different extraction strategies for each of these possible structures. The strategies are built either upon syntactic structures with explanatory role in natural languages (i.e. appositions, acronym extensions) that implicitly incorporate the notion of *proximity* or upon lexico-syntactic patterns, where *proximity* is not

relevant anymore, and therefore makes the *Answer Selector* unnecessary for definition questions. The *Centroid Ranker* provides the end scores for deciding on the correct answers based on their *redundancy*.

8.1 Answers to Factoid Questions

We consider factoid questions as questions that have a short answer, which is a noun phrase typically referring to a named entity. Therefore, the *Answer Extractor* for factoid questions is targeting only noun phrases that are either named entities or first-order chunk structures (i.e. no other such structures embedded). Since there are different ways of referring to a specific named entity or concept, the *Answer Clusterer* takes care of normalizing those to a common representation, by grouping either co-referencing named entities (i.e. *lieutenant John M. Eisner ~ John Eisner*) or chunks with the same head (i.e. *the world's largest semiconductor company ~ a US-based multinational company*). This kind of action might have unforeseeable results if taken out of the context, but it is well grounded given the fact that all the answers satisfy the same set of constraints as imposed by the question. Through this normalization we want to gather enough evidence for a redundancy-based answer candidate extraction.

8.1.1 Candidates Extraction by Redundancy in Centroid Ranker

As previously mentioned, we consider redundancy a good indicator for the answer's suitability. Candidate answers supported by different lexical contexts relevant to a specific question provide more evidence for their possible suitability to correctly answer the question. Redundancy of information is computed in terms of occurrence frequency over unique information retrieval units; that is, redundancy of a candidate answer is directly proportional to the number of times it appears in the non-duplicated relevant sentences retrieved by the *Unit Retrieval* sub-system.

We devised three alternative ways of computing the redundancy value of an answer candidate depending on its frequency of occurrence: over documents, over sentences and over weighted sentences. The first method considers redundancy to be equal to the frequency of the answer candidate over unique documents and is equivalent to the *df* (document frequency) known from the IR models. The second method defines redundancy in terms of sentence frequency

and is different from the previous method by counting sentences within the same document as evidence. This approach will prefer answer candidates that are mentioned more often in relevant segments of a document, whereby the answer distribution over documents is similar to that of the first method. While these methods are based solely on statistics of answer candidate occurrence, they fail to consider the relevance ranking of the *Unit Retrieval* sub-system. The third method factors the answer's statistics and its relevance score into a single measure to reflect the goodness of fit as provided by the *Unit Retrieval* into the redundancy measure. We are therefore interested in redundant highly relevant rather than frequent occurring information. For this case we define *redundancy* as:

$$redundancy(A) = \sum_{D} \sum_{S} \frac{rel(S, A)}{\#S \times \#D} \qquad (8.1)$$

where A is the answer candidate, D is a document, S a relevant sentence within document, $rel(S, A)$ is the relevance score of sentence S containing answer A as delivered by the *Unit Retrieval* sub-component, $\#S$ is the number of relevant sentences from document D and $\#D$ is the number of relevant documents containing answer A. Empirical results have shown that the latter method of computing redundancy clearly outperforms the previous ones and therefore it will be the one referenced throughout this work.

8.1.2 Answer Selection by Proximity

Proximity matters because words that are close to each other in the text are more likely to be closely connected in the meaning structure of the text. It is true that words in a question have some explicit or implied linguistic relationship between them, and that a good match for such questions is likely to be one that has the same relationship between those words.

To select the best answer among those identified in answer extraction, we use a *weighting* measure based on how distant the candidate answer is to significant terms from the question. The *distance* measure marks each term in an answer sentence that matches a keyword from the question and then looks how far this term is from the candidate answer, measured as the number of words

that have to be traversed in the sentence. The *weight* of each matched question term is then defined as:

$$weight(T, A) = \begin{cases} 1 - \exp(dist - K), dist < K \\ 0.5, otherwise \end{cases} \quad (8.2)$$

whereby *dist* is the distance above-mentioned between the term T and the candidate answer A and K is a constant marking the limit from where words do not play a role anymore in building relevant relations with the candidate answer. Through empirical observation, the value of K has been set to 5, meaning that words within this distance seemed to stay in a relationship relevant to our goal.

The idea of *proximity* is to provide an approximation to matching the linguistic relations between words, in that if an answer were closely related to the matched question terms, then it would have a small proximity, whereas if it had an indirect relation, the proximity would be higher. The overall *proximity* is calculated by averaging these weights for each of the question terms, factoring it with a measure of their textual cohesion and taking its reciprocal:

$$proximity = \frac{1}{avg(\sum_{T \in Q} weight(T, A)) * cohesion} \quad (8.3)$$

whereby

$$cohesion = \max_{T_i, T_j \in Q} (weight(T_i, T_j)) \quad (8.4)$$

The cohesion factor is simply a way of taking into account the relationships between the question terms that were matched, beside their relationship to the candidate answer.

8.2 Answers to Definition Questions

According to search engine user logs, about one third of the information need consists of definitions. Hence, techniques to handle this category of questions in a question answering system are very important.

Most difficulties in answering definition questions arise from the lack of a clearly defined semantic category that restricts the candidate answers. In contrast to factoid questions that categorize the answering strategies according to their expected answers, candidate answers of definition questions rarely fall in

separate semantic categories. Moreover, definition questions (e.g. *What is the Grammy?*, *Who was Al Capone?*) contain very few non-stop-words rendering answering strategies based on query words co-occurrence useless. Therefore different extraction techniques than those for factoid questions have to be considered.

The use of surface patterns for answer extraction has proven to be an effective strategy for definition question answering (Fleischman, M. et al., 2003; Hildebrandt, W. et al., 2004). The patterns operate both at the word and part-of-speech level (lexico-syntactic patterns) and involve shallow text processing and conventional definition cues. They leverage both linguistic structures (e.g. appositions) and heuristics (e.g. use of hypernyms) known to be used in describing relevant features of the entities to be defined. The range of such patterns varies from domain specific to general and can be either restricted to certain types of terms (e.g. acronyms for ORGANIZATION) or applied with no restrictions.

Given the fact that the document collection in our case is made up of news from all possible domains, we have opted for general lexico-syntactic patterns with no explicit domain specificity. Following is the list of domain independent patterns that have been considered.

8.2.1 Appositions

The task of identifying the parts of documents that contain definitions of entities is difficult even for humans, but we provisionally adopted the assumption that the definition of an entity is expressed through a figure of speech called apposition that often results when the verbs (especially verbs of being) in supporting clauses are eliminated to produce shorter descriptive phrases.

Apposition is a grammatical construction in which two noun phrases are placed side by side with one element serving to define or modify the other. Appositions can either be *restrictive*, or *non-restrictive*, depending on the role of the second element either to limit or clarify the foregoing one, or to provide additional information about the first element. While for a non-restrictive apposition the second noun phrase must be preceded or set off by commas (e.g. *Helmut Kohl, the German chancellor, visited ...*), for a restrictive apposition the

following element is not set off by commas (e.g. *the German chancellor Helmut Kohl visited ...*).

In detecting non-restrictive appositive relations, punctuation disambiguation plays an important role. By *punctuation disambiguation*, we mean distinguishing the syntactic roles of commas, of which three are relevant in our case: as appositive markers, movement markers, and coordination markers. For example, in the sentence "*When John met Marry, the situation has changed."* the comma is used as a marker of syntactic movement. On the contrary, in the sentence *"George, Marry, John and Paula joined the meeting."* the comma shows a coordinative relation between *George* and *Marry*. In both cases, the commas are placed between noun phrases showing that more information is required in order to disambiguate their intended usage.

We therefore created several heuristics to disambiguate punctuations and then to identify non-restrictive appositive relations. Here are examples of the heuristics:

- If a sentence starts with a subordinating conjunction, the leftmost comma in the sentence is a movement marker.
- If a sentence contains the sequence of *"NP, NP CC NP"*, these commas are coordination markers.

As previously mentioned for restrictive apposition the following element is not set off by commas and therefore we cannot rely anymore on punctuation for identifying the corresponding instances (e.g. *the German chancellor Helmut Kohl visited ...*). However by using a chunk parser we can spot those cases of immediately following noun phrases. By considering the second one (*Helmut Kohl*) a likely instance of the foregoing phrase (*the German chancellor*), we can deliver the latter as a possible definition of the entity asked for.

8.2.2 Acronyms

An abbreviation is a shortened form of a word or phrase, usually consisting of a letter or group of letters taken from it. Acronyms are abbreviations that are formed using the initial elements in a phrase or name. These elements may be individual letters (as in *NATO*) or parts of words (as in *Interpol*) and are

frequently used in both spoken and written language, making for a fair percentage of the definition questions as well.

Similar to appositions, acronyms are marked by punctuation mixed with specific lexical items that ease their extraction automatically. Following are the most common markers that have been used in our case:

- left marker: "*(*" right marker: "*)*"
- left marker: "*, or*" right marker: "*,*"

For example, for definition questions asking for acronyms (e.g. *What does NATO stand for?*), we retrieve all sentences in which the acronym (NATO) appears. Then, a regular expression is used to extract all contexts of the acronym matching one of the following patterns:

- STRING+ left_marker ACRONYM right_marker
- ACRONYM left_marker STRING+ right_marker

Finally, the sequence of characters in the acronym (*NATO*) is compared to a sequence of characters in the full name (*North Atlantic Treaty Organization*) making sure that all characters of the short form occur in the extension of the acronym. However, this simple test might lead to inappropriate extensions for some cases (*HUGO* vs. *Human Genome Organization*), as shown below:

- HUGO
- **HU**man **G**en**O**me

when the extension is matched only partially. Therefore, an additional test has been considered that compares the short form with the full form backwards to make sure that every word spanned by the match contains at least one character from the short form and to be sure that the last match (first character in this case) is uppercase:

- HUGO
- **HU**man Genome Or**G**anizati**O**n (discarded by the first constraint)
- **HU**man **G**enome **O**rganization (both constraints met)

The longest match meeting the constrains mentioned above delivers the final result.

8.2.3 Lexical Definition

The lexical definition of a term, also known as the dictionary definition, is the meaning of the term in common usage. There are several ways to define a term, of which a few most common options have been considered:

1. Define by *function*. Explain what something does or how something works.
2. Define by *structure*. Tell how something is organized or put together.
3. Define by *analysis*. Compare the term to other members of its class and then illustrate the differences. These differences are special characteristics that make the term stand out. For example, compare a Siberian husky to other dogs, such as lap dogs, mutts, or sporting dogs.

While the first two methods have been implemented by way of lexico-syntactic patterns, the latter only uses lexical information derived from GermaNet.

Functional and structural definitions make use of the following lexico-syntactic pattern:

CONSTITUENT definition_verb STRING+

whereby the CONSTITUENT has to include the entity to be defined as its syntactic head and the *definition_verb* must belong to a predefined list of verbs commonly used for such purposes:

- Functional verbs (English translations): *use*, *perform*, *provide*, etc.
- Structural verbs (English translations): *comprise*, *consist of*, *made of*, etc.

The same pattern can be used for questions asking for the definition of PERSON instances, in which case verbs are used with known explanatory roles, such as: *be*, *become*, etc.

Analytical definitions do not use any syntactic patterns at all, but they presume that co-occurrence of similar entities is a good indicator for possible definitions. Similar entities have been considered both synonyms and hyponyms of its father for any term being looked for. The relations of synonymy and hyponymy were extracted from the German EuroWordNet.

8.2.4 Hypernyms

Earlier research showed that hypernyms could be used as good answers to definition questions of the type "*What is*" (Prager, J. et al., 2001). For example *tsunami* is a *wave*, where the latter is a hypernym of the former in WordNet.

Deciding which hypernym to consider as an appropriate answer is highly dependent on its co-occurrence statistic with the target entity. Moreover, there are cases when hyponym-hypernym relations were not entirely encoded in the lexical resources, especially when dealing with domain-specific terminology and proper names. Similarly, a method proposed by Hearst (1998) to identify patterns that signal particular lexical semantics relations can be used to discover a set of high precision hyponym-hypernym patterns that are common across text genres.

The patterns (English translations) are shown below, with qt (query term) and dp (descriptive phrase) being phrases containing hyponyms and hypernyms:

1. (*dp* such | such *dp*) as *qt*
   e.g., "mental illness such as schizophrenia"
2. *qt* (and | or) other *dp*
   e.g., "schizophrenia and other mental illnesses"
3. *dp* like *qt*
   e.g., "mental illnesses like schizophrenia"
4. *dp* (called | known as) *qt*
   e.g., "mental illnesses called schizophrenia"
5. *dp* including *qt*
   e.g., "mental illnesses including schizophrenia"

Some of the surface patterns presented above are restricted to special types of entities being asked for, while others have general validity. Acronym patterns apply with predilection to entities of type ORGANIZATION and structural lexical definitions to entities of type OBJECT. The rest of the patterns are used for all entity types supported by the system, with possible exceptions depending solely on the particular entity.

8.3 Evaluation

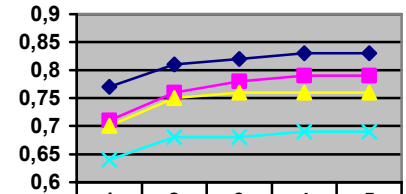
The evaluation of the Answer Extraction and Selection component assumed only those questions that passed the previous components in the monolingual scenario (Question Analysis and Information Unit Retrieval) with at least one match against the Gold Standard. The performance of the different configurations has been measured by way of Mean Reciprocal Rank (MRR), which gives a figure of how good the component is in ranking the relevant answer on the top of the answer list. Since our component works on the top 5 most frequent candidate answers, from which it selects the right answer, we have measured performance at each of the top 5 ranks.

Evaluation of the Information Unit Retrieval has shown a quick increase in performance over the top 10 documents retrieved, with a flattening curve for the rest of the documents. That means that by increasing the number of retrieved documents over a threshold of 10, the number of relevant documents does not increase very much. The same evaluation showed that by increasing the size of the retrieval unit, the number of relevant documents increases as well. For our purpose we have chosen measuring performance of the Answer Extraction and Selection component with different numbers of retrieved information units: 5, 10, 100 and 500. While the last two configurations should reveal the proof of concept for the answer selection, which uses linear distance combined with extraction by redundancy, the first two configurations should give a figure about the power of unit retrieval and answer selection.

The result of the evaluation has showed that best results are to be attained by using an information unit consisting of 1-sentence and building on a search engine with high accuracy on top matches. Compared to the results of Unit Retrieval it looks as if the Answer Selection component cannot maintain the

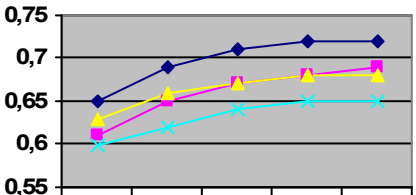
upward monotonicity when the retrieval unit's size increases. Responsible for this effect is the way that we defined the weight measure (8.2) that marks as relevant only terms within a distance of five words from the answer candidate. Increasing the unit size will allow for more question-relevant contexts, but the query words will be more widespread, as well. The result of measuring performance for higher numbers of information units (100 and 500) shows that the Answer Selection component is robust enough to determine the right answers even when the redundancy concept may be distorted by huge amounts of data.

MRR @ 1-sent



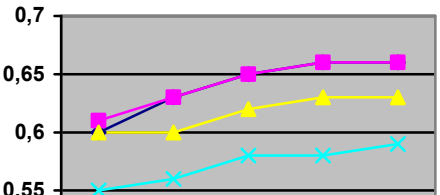
	1	2	3	4	5
5	0,77	0,81	0,82	0,83	0,83
10	0,71	0,76	0,78	0,79	0,79
100	0,7	0,75	0,76	0,76	0,76
500	0,64	0,68	0,68	0,69	0,69

MRR @ 3-sent



	1	2	3	4	5
5	0,65	0,69	0,71	0,72	0,72
10	0,61	0,65	0,67	0,68	0,69
100	0,63	0,66	0,67	0,68	0,68
500	0,6	0,62	0,64	0,65	0,65

MRR @ 5-sent

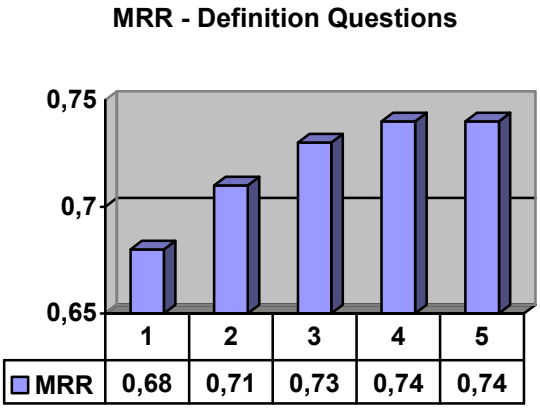


	1	2	3	4	5
5	0,6	0,63	0,65	0,66	0,66
10	0,61	0,63	0,65	0,66	0,66
100	0,6	0,6	0,62	0,63	0,63
500	0,55	0,56	0,58	0,58	0,59

Evaluating systems that answer definition questions is much more difficult than evaluating systems that answer factoid questions because it is no longer useful to judge a system response as simply right or wrong. Assigning partial credit to a response requires some mechanism for matching the concepts in the desired response to the concepts present in a possible response. The issues are

similar to those that arise in the evaluation of machine translation and automatic summarization. Therefore we have opted for a manual evaluation of the definition questions and we have used the Mean Reciprocal Rank (MRR) as a figure of measuring performance.

Since all the methods implied for answering definition questions either make use of a full sentence or apply patterns of local syntax, the component has been evaluated only for the information unit retrieval of 1-sentence length.



8.4 Summary

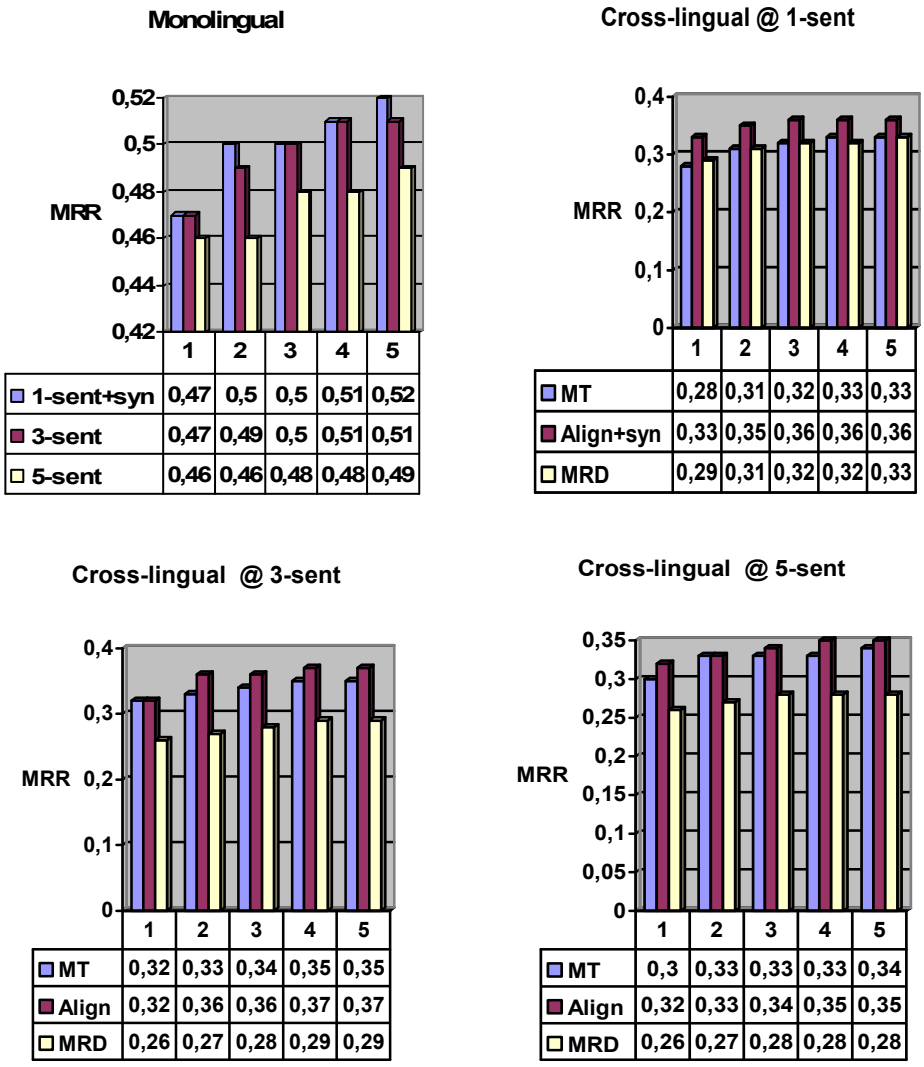
The Answer Extraction component builds on the assumptions that redundancy of candidate answers is a good indicator for their suitability and proximity of concepts within a sentence approximates their semantic dependency. Regarding sentences as the most compact forms of expressing facts, it is possible to correctly answer questions based solely on sentences that match the given topic. For definition questions we have shown that using general lexico-syntactic patterns in extracting potential answer candidates does a fairly good job without being very specific about the various types of questions.

9 Conclusions and Future Work

Throughout this work we have evaluated each component individually, making some assumptions of independence over previous components in the workflow. However, in order to have a clear picture of the system's performance as a whole, evaluation of the integrated components for both monolingual and cross-lingual scenarios has been pursued. The evaluation took into consideration the best results of individual components in defining the final configuration:

- measuring performance by way of Mean Reciprocal Rank (MRR) over different unit retrieval sizes for the top 5 results;
- using query expansion with synonyms for retrieval units of 1-sentence length in the monolingual scenario and in the cross-lingual scenario for the alignment method.

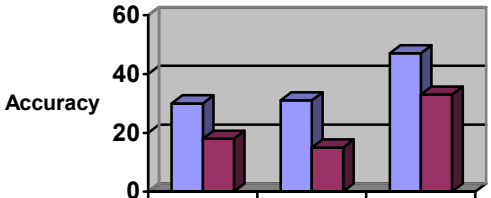
Results of evaluating the whole system (see next page) reinforced the results of individual evaluations that for crossing the language barrier from English to German the most efficient method is through alignment of the original question analysis into a similar structure. The black box evaluation also shows that small retrieval units of 1-sentence length benefit translation with MRD without any disambiguation involved. While performance decreases for the MT and Align methods by varying the retrieval unit length from 3 to 1, it rises for the MRD method. Given the similarity between the Align and MRD methods, except for their way of translating the relevant terms, we can conclude that the increase in performance for the latter is due to the reduced ambiguity of collocating words in local contexts. The local context is constrained in this case by the definition of the weight measure (formula 8.2) to a window of 5 words around the correct answer.



A relative evaluation of our system's performance compared to those evaluated in the CLEF 2007 (Forner, P. et al., 2007) and CLEF 2008 (Giampiccolo, D. et al., 2008) forums, based on the same set of questions for the language pairs German – German and English – German, shows that our assumptions and their integration into a Question Answering framework are good enough to outperform state-of-the-art approaches for factoid questions(Figure 46) and definition questions on the average (Figure 47). The measure used for comparison is accuracy and can be interpreted as the MRR for the answers ranked first.

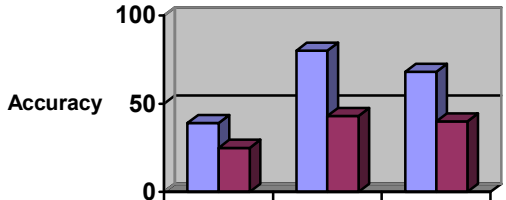
The systems that participated in the CLEF evaluation forum of the year 2007 and year 2008 for the above mentioned language pairs were based on the following approaches for Question Answering: matching of semantic network representations of both the question and the documents (Hartrumpf, S. et al.,

2007, 2008), proximity-based answer selection (Sacaleanu, B. et al., 2007, 2008) and logic-based answer extraction (Glöckner, I. and Pelzer, B., 2008).



	2007	2008	Quantic
Monolingual	30	31	47
Crosslingual	18	15	33

Figure 46. Comparison to CLEF-DE best results for factoid question.



	2007	2008	Quantic
Monolingual	39	80	68
Crosslingual	25	43	40

Figure 47. Comparison to CLEF-DE best results for definition questions.

The system described by Hartrumpf, S. et al. (2007, 2008) builds on a main precision oriented sub-system that uses full sentence parses, rule-based inferences on semantic representations and matching of those representations for questions and documents. The main component is backed by two further answer producers based on pattern matching and answer redundancy, and the results of all three are merged by an answer validator that uses deep linguistic processing and logical reasoning. For the cross-lingual task (2008), the system uses a machine translation service for direct question translation. The overall performance registered by this complex system was of 19% for year 2007, respective 23% for year 2008, in the monolingual scenario and 14% for the cross-lingual one.

Sacaleanu, B. et al. (2007, 2008) use an earlier version of the Quantico system that differs from the one presented in this work by using a simpler method of computing answer redundancy (method 2 of section 8.1.1) and the weight of a term (formula 8.2) being inversely proportional to the distance from the answer candidate. For the cross-lingual scenario, the system uses several

direct machine translations of the question that are individually interpreted and the outcome ranked according to linguistic well-formedness and completeness with respect to question information (question type, question focus, answer type). The overall performance of this system was of 30% for year 2007, respective 37% for year 2008, in the monolingual scenario and 18.5% for year 2007, respective 14.5% for year 2008, in the cross-lingual one.

Glöckner, I. and Pelzer, B. (2008) make extensive use of logic for simultaneously extracting answer bindings and validating corresponding answers. It builds upon the same semantic representation as that of Hartrumpf, S. et al. (2007), but fails to use the prover for an incomplete parse analysis. The results registered by this system were of 14.5% for year 2008 for the monolingual scenario.

A direct comparison of the system described in this work with the QA-systems that were evaluated in the CLEF 2007 and 2008 forums shows that our proximity measure is a good and robust approximation for the linguistic relationship among words when selecting the correct answer, and transfer-based translation methods are effective approaches of crossing the language barrier beside direct question translation.

9.1 Summary of contributions and answers to research questions

In this work we presented an open domain cross-lingual English to German Question Answering system that leverages the performance of a mono-lingual German system by translating the question into the target language. We compared two different techniques of translation, by directly translating the question and by translating the result of interpreting the question. We integrated information extraction results in the form of Named Entities into the retrieval mechanism and investigated two methods of query expansion for document retrieval through synonyms and through related concepts. Issues of term ambiguity during expansion have been dealt with by reducing the limits of the retrieved textual unit to those of a sentence and decreasing the probability of inappropriate meanings to co-occur with keywords from the question. We explored several strategies of extracting answers, based on proximity and lexico-

syntactic patterns, and combined them in a framework for factoid and definition questions. The Question Answering system developed along the previous-mentioned results has been shown to outperform state-of-the-art QA systems on the same data.

It has been shown that small sized sentence-based retrieval units are an alternative to document-based ones when retrieving relevant information for Question Answering. Moreover, small contexts seem to benefit implicit selection of the right meaning when employing extension methods without any word sense disambiguation. While "one sense per collocation" holds for 1-sentence sized retrieval units for a monolingual scenario, it does not hold for a cross-lingual setting when the ambiguous words and their extensions (i.e., translations) are from different languages with different levels of polysemy. However, results have shown that further restricting the meaning of a small context to a window of 5 words around the targeted terms will reinforce the truth of our assumption.

Proximity, as a method of approximation for linguistic relationship among words, has been shown to be an efficient measure for selecting the correct answer, while redundancy was used as a good indicator for the answer candidate's suitability.

As for the cross-lingual task of Question Answering, a new defined method that first extracts the semantics of a question and then translates it using automatically generated alignment lists of source- to target-language keywords outperformed the traditional widespread machine translation method for all different settings.

9.2 Future Work

For the goal of documenting future work we have done an error analysis of the system's performance. 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,
- online 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,
- proximity measure is inversely proportional to the average weight for all matched question terms.

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), being a statistical based entity extractor, has a better coverage and precision on annotating the document collection, where lots of context data are available, compared to its performance 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 to correctly translate 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 mistranslations that resulted in incorrect IR-queries generation and therefore wrong answers.

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

birthplace <> born

homeland <> born

monarch <> king

profession <> designer

Conceptual – Proximity is inversely proportional to the average weight of all matched question terms

The way proximity was defined works just fine for questions that were not expanded with lexical or conceptual items. Query expansion and MRD translations add related terms that might occur together with the original question word and therefore decrease the average weight for the intended concept. Accordingly, proximity increases and answers get lower ranked though they are equally relevant as the ones matching only the question word. This drawback is more noticeable as the retrieval unit size increases and with it the probability of term/extension and alternative translations co-occurrences.

We expect that further work along the above-mentioned issues will reinforce the presumptions made throughout this work and will improve the overall results of the system.

Annexes

Annex 1 – Sound Shifts between English and German

German	English	Examples
b	f	Dieb - thief halb - half
b	v	eben - even Grab - grave sieben - seven
ch	k	Buch - book Elch - elk sprechen - speak
ch	gh	acht - eight lachen - laugh Licht - light Fracht - freight
d	th	Bad - bath drei - three Erde - earth Leder - leather
f	p	Bischof - bishop helfen - help scharf - sharp
ff	p/pp	Affe - ape Pfeffer - pepper Schiff - ship

g	y	Tag - day Weg - way
k	c	Karte - card Keller - celler
k	ch	Kapelle - chapel Kinn - chin
mm	mb	Lamm - lamb Nummer - number
pf	p/pp	Apfel - apple Pfad - path Pfanne - pan
sch	s/sh	falsch - false Fleisch - flesh Schnee - snow
ss	t/tt	besser - better dass - that Wasser - water
t/th	d	Taler/Thaler - dollar vorwärts - forward Wort - word
t/tt	d	Bett - bed gut - good reiten - ride
tt	th	Mutter - mother Wetter - weather
v	f	Vater - father vier - four Volk - folk
z/tz	t	Herz - heart zehn - ten

		zwei - two Katze - cat
z	c	Eleganz - elegance zirka - circa Zirkus - circus

Table 6. Spelling consonant shifts.

German	English	Examples
a	au	lachen - laugh schlachten - slaughter
a	ea	schwach - weak Waffe - weapon
a	i	Macht - might Nacht - night
a	o	alt - old Kamm - comb
ä	e	Ägypten - Egypt Äquator - equator
au	ou	laut - loud Maus - mouse sauer - sour
e	ea	Feder - feather Herz - heart
e	i	geben - give leben - live
ei	i	beißen - bite reißen - rip
ei	o	Heim - home Stein - stone

ie	ee	Bier - beer Knie - knee
ie	o	Liebe - love vierzig - forty
o	ea	Ost - east tot - dead
u	oo	Buch - book Blut - blood Fuß - foot
u	ou	jung - young Pfund - pound

Table 7. Spelling vowel shifts.

Annex 2 – DROOLS Rules for English Question Analysis

```
package sbe.test.drools
import java.util.List;
import java.util.Set;
import sbe.util.ObjectPair;
import edu.stanford.nlp.ling.TaggedWord;

rule "Auxilliary verbs"
   salience 100
   no-loop true
   when
      $auxVerbs : List()
   then
      $auxVerbs.add("do");
      $auxVerbs.add("have");
      $auxVerbs.add("be");
      $auxVerbs.add("are");
      update($auxVerbs);
end

rule "Object WH-words"
   salience 100
   no-loop true
   when
      $objWH : Set()
   then
      $objWH.add("where");
      $objWH.add("when");
      $objWH.add("whose");
      $objWH.add("whom");
      update($objWH);
end
```

```
////////////////////////////////////////////////////////////////////////////
// Rules for OPEN/CLOSED questions
////////////////////////////////////////////////////////////////////////////

rule "Closed Question"
    salience 90
    no-loop true
    when
        $auxVerbs : List()
        $q : Question (firstToken memberOf $auxVerbs, value matches ".*\?$")
    then
        System.out.println("CLOSED QUESTION");
        $q.setOpenQuestion(false);
        update($q);
end

rule "Open Question"
    salience 90
    no-loop true
    when
        $auxVerbs : List()
        $q : Question (firstToken not memberOf $auxVerbs, value matches ".*\?$",
openQuestion != true)
    then
        System.out.println("OPEN QUESTION");
        $q.setOpenQuestion(true);
        update($q);
end

////////////////////////////////////////////////////////////////////////////
// Rules for OBJECT/SUBJECT questions
////////////////////////////////////////////////////////////////////////////

// ?word auxiliary subject main_verb_missing
```

```
rule "OBJECT Question: open question, starts with WH-word: WHERE, WHEN,
WHOM"
   salience 80
   no-loop true
   auto-focus true
   //agenda-group "qType"
   when
      $objWH : Set()
      $q : Question (openQuestion == true,
                  objectQuestion != true,
                  firstToken memberOf $objWH
                  )
   then
      System.out.println("OBJECT QUESTION 1");
      $q.setObjectQuestion(true);
      update($q);
end

// ?word subject auxiliary main_verb
rule "SUBJECT Question: open question, auxilliary immediately followed by main
verb"
   salience 80
   no-loop true
   auto-focus true
   //agenda-group "qType"
   when
      $auxVerbs : List()
      $q : Question (openQuestion == true,
                  subjectQuestion != true,
                  firstVerb != null,
                  secondVerb != null,
                  firstVerb.value memberOf $auxVerbs,
                  eval(secondVerb.getIndex() - firstVerb.getIndex() == 1)
                  )
```

```
    then
        System.out.println("SUBJECT QUESTION 1");
        $q.setSubjectQuestion(true);
        update($q);
end

// ?word auxiliary subject main_verb
rule "OBJECT Question: open question, auxilliary and main verb separated by the
subject"
    salience 80
    no-loop true
    auto-focus true
    when
        $auxVerbs : List()
        $q : Question (openQuestion == true,
                    objectQuestion != true,
                    firstVerb != null,
                    secondVerb != null,
                    firstVerb.value memberOf $auxVerbs,
                    eval(secondVerb.getIndex() - firstVerb.getIndex() > 1)
                    )
    then
        System.out.println("OBJECT QUESTION 2");
        $q.setObjectQuestion(true);
        update($q);
end

// ?word subject main_verb
rule "SUBJECT Question: open question, only main verb"
    salience 80
    no-loop true
    auto-focus true
    //agenda-group "qType"
    when
```

```
        $auxVerbs : List()
        $q : Question (openQuestion == true,
                        firstVerb != null,
                        //firstVerb.value not memberOf $auxVerbs,
                        secondVerb == null,
                        subjectQuestion != true
                        )
    then
        System.out.println("SUBJECT QUESTION 2");
        $q.setSubjectQuestion(true);
        update($q);
end

////////////////////////////////////////////////////////////////////////////
// Rules for determining DEFINITION questions
////////////////////////////////////////////////////////////////////////////

rule "Definition Question: open question, starting WH-word: WHO or WHAT, one
verb = BE"
// EXAMPLES: What is the Taj Mahal? What is BASF? Who was John Lenon?
    salience 70
    no-loop true
    auto-focus true
    agenda-group "qType"
    when
        $q : Question (openQuestion == true,
                        $1stVerb : firstVerb != null,
                        firstToken in ("who", "what"),
                        firstVerb.value in ("be", "are"),
                        secondVerb == null,
                        definitionQuestion == false
                        )
```

```
eval($q.getLargestConstituentStarting($1stVerb.getIndex()+1).getHead().getTag().toS
tring().equals("NNP"))
   then
      System.out.println("DEFINITION QUESTION1");
      $q.setDefinitionQuestion(true);
      $q.setSubjectQuestion(true);
      update($q);
end

rule "Definition Question: open question, starting WH-word: WHAT, one verb = BE,
only one constituent after verb, undetermined head"
// EXAMPLES: What is plastination? What is a meter? What is BASF?
// EXCEPTIONS:  What is the Braille lettering? What are the pyramids?
   salience 70
   no-loop true
   auto-focus true
   agenda-group "qType"
   when
      $q : Question (openQuestion == true,
                  $1stVerb : firstVerb != null,
                  firstToken == "what",
                  firstVerb.value in ("be", "are"),
                  secondVerb == null,
                  definitionQuestion == false
                  )
         eval ($q.getLargestConstituentStarting($1stVerb.getIndex()).getEnd() ==
$q.getSmallestConstituentStarting($1stVerb.getIndex()+1).getEnd())
      eval
(!$q.getDeterminant($q.getLargestConstituentStarting($1stVerb.getIndex()+1).getHea
d().getWord()).equals("the"))
   then
      System.out.println("DEFINITION QUESTION2");
      $q.setDefinitionQuestion(true);
```

```
        $q.setSubjectQuestion(true);
        update($q);
end

////////////////////////////////////////////////////////////////////////////
// Rules for determining FACTOID questions
////////////////////////////////////////////////////////////////////////////

rule "Factoid Question: anything that is not a Definition Question"
    salience 60
    no-loop true
    auto-focus true
    agenda-group "qType"
    when
        $q : Question (openQuestion == true, definitionQuestion == false,
factoidQuestion != true)
    then
        System.out.println("FACTOID QUESTION");
        $q.setFactoidQuestion(true);
        update($q);
end

////////////////////////////////////////////////////////////////////////////
// Rules for determining FOCUS of questions
////////////////////////////////////////////////////////////////////////////

rule "FOCUS of definition questions"
    salience 20
    no-loop true
    agenda-group "qType"
    when
        $q : Question (definitionQuestion == true, focus == null)
    then
        System.out.println("FOCUS set");
```

```
$q.setFocus($q.getLargestConstituentStarting($q.getFirstVerb().getIndex()+1).getVal
ue());
        update($q);
end

rule "FOCUS of factoid questions"
    salience 20
    no-loop true
    agenda-group "qType"
    when
        $q : Question (factoidQuestion == true, focus == null)
        eval
($q.getLargestConstituentEnding($q.getFirstVerb().getIndex()).getHead().getTag().st
artsWith("W"))
    then
        System.out.println("FOCUS2 set");

$q.setFocus($q.getGovernerOfDet($q.getLargestConstituentEnding($q.getFirstVerb()
.getIndex()).getHead().getWord()));
        update($q);
end

rule "FOCUS of factoid questions starting with WHOSE"
// EXAMPLE: Whose car did you see? Whose wife came in?

    salience 25
    no-loop true
    agenda-group "qType"
    when
        $q : Question (factoidQuestion == true, focus == null, $focus : firstToken ==
"whose")
    then
        System.out.println("FOCUS31 set");
```

```
        $q.setFocus($focus);
        update($q);
end

rule "FOCUS of factoid questions starting with HOW Word[POS=JJ|RB]"
// EXAMPLE: How many students took part at the demonstration? How old is the
wife of John? How far away from Paris is Metz?

    salience 25
    no-loop true
    agenda-group "qType"
    when
        $q : Question (factoidQuestion == true, focus == null)
        eval ($q.getValue().startsWith("How"))
        eval ($q.getTaggedQuestion().get(1).getTag().startsWith("JJ") ||
$q.getTaggedQuestion().get(1).getTag().startsWith("RB"))
    then
        System.out.println("FOCUS32 set");
        $q.setFocus($q.getTaggedQuestion().get(1).value());
        update($q);
end

rule "FOCUS of factoid questions with incorect parse tree: WHNP not recognized"
// EXAMPLE: Which country of the world has the largest population in the world?
// EXAMPLE: How many students took part at the demonstration?
// PARSE-TREE: (ROOT (FRAG (SBAR (WHNP (WDT Which)) (S (NP (NP (NN
country)) (PP (IN of) (NP (DT the) (NN world)))) (VP (VBZ has) (NP (NP (DT the)
(JJS largest) (NN population)) (PP (IN in) (NP (DT the) (NN world)))))))) (. ?)))
// (ROOT (SBARQ (WHADVP (WRB How)) (S (NP (JJ many) (NNS students)) (VP
(VBD took) (NP (NN part)) (PP (IN at) (NP (DT the) (NN demonstration))))) (. ?)))

    salience 20
    no-loop true
    agenda-group "qType"
```

```
    when
        $q : Question (factoidQuestion == true, focus == null)
        eval
($q.getLargestConstituentEnding($q.getFirstVerb().getIndex()).getHead().getTag().st
artsWith("NN"))
    then
        System.out.println("FOCUS4 set");

$q.setFocus($q.getLargestConstituentEnding($q.getFirstVerb().getIndex()).getHead().
getWord());
        update($q);
end

////////////////////////////////////////////////////////////////////////////
// Analysis of FOCUS
////////////////////////////////////////////////////////////////////////////

rule "real FOCUS of factoid subject questions with copula verb"
// EXAMPLE: What is the population of Germany?

    salience 10
    no-loop true
    //auto-focus true
    agenda-group "qType"
    when
        $q : Question (factoidQuestion == true,
                    subjectQuestion == true,
                    focus in ("what", "which", "who"),
                    $1stVerb : firstVerb != null,
                    firstVerb.value in ("be", "are"))
    then
        System.out.println("FOCUS5 set");
        $q.setFocus($q.getLargestConstituentStarting($q.getFirstVerb().getIndex() +
1).getHead().getWord());
```

```
        update($q);
end

rule "real FOCUS of factoid subject questions with copula verb and asking for a
NAME of something not a proper noun (NNP)"
// EXAMPLE: What is the name of the Danish capital?

    salience 9
    no-loop true
    //auto-focus true
    agenda-group "qType"
    when
        $q : Question (factoidQuestion == true,
                       subjectQuestion == true,
                       focus == "name",
                       $1stVerb : firstVerb != null,
                       firstVerb.value in ("be", "are"))
        eval ($q.getDependentForPrepOf("name") != null
              && !$q.getDependentForPrepOf("name").getLabel().equals("NNP")
        )
    then
        System.out.println("FOCUS6 set");
        $q.setFocus($q.getDependentForPrepOf("name").getValue());
        update($q);
end
```

Bibliography

Abney, S., Collins, M. and Singhal A. (2000). Answer extraction. *Conference on Applied Natural Language Processing (ANLP)*, 2000.

Akiba, T., Shimizu, K. & Fujii, A. (2008). Statistical machine translation based passage retrieval for cross-lingual question answering. *IJCNLP 2008: Third International Joint Conference on Natural Language Processing*, 2008.

Alias-i. (2003). LingPipe 1.7. http://alias-i.com/lingpipe (accessed October 1, 2003)

Baldwin, B. (1995). CogNIAC: A discourse processing engine. *University of Pennsylvania, Department of Computer and Information Science, Ph. D. dissertation*.

Bernhard, D. & Gurevych, I. (2009). Combining lexical semantic resources with question \& answer archives for translation-based answer finding. *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP* (ACL-IJCNLP), 2009.

Bos, J & Nissim, M. (2007). Answer Translation: An Alternative Approach to Cross-Lingual Question Answering. *Evaluation of Multilingual and Multi-modal Information Retrieval*, pp 290–299, Lecture Notes in Computer Science (Vol 4730).

Bowden, M., Olteanu, M., Suriyentrakorn, P., Clark, J. & Moldovan, D. (2006). LCC's PowerAnswer at QA@CLEF 2006, *Working Notes for the CLEF 2006 Workshop*, 2006.

Bourdil, G., Elkateb, F., Grau, B., Illouz, G., Monceaux, L., Robba, I. & Vilnat, A. (2004). How to answer in English to questions asked in French: by exploiting results from several sources of information. *Working Notes for the CLEF 2004 Workshop,* 2004.

Brants, Thorsten.(2000). TnT - A Statistical Part-of-Speech Tagger. *6th Applied Natural Language Processing (ANLP '00),* 2000.

Brennan, S. E., Friedman, M.W. & Pollard, C.J. (1987). A centering approach to pronouns. *Proceedings of the 25th annual meeting on Association for Computational Linguistics*, 1987.

Buscaldi, D. & Rosso, P. (2006) Mining Knowledge from Wikipedia for the Question Answering Task. *LREC 2006 Proceedings*, 2006.

Celikyilmaz, A., Thint, M. & Huang, Z. (2009). A graph-based semi-supervised learning for question-answering. *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 (ACL-IJCNLP '09)*, 2009.

Clarke, C., Cormack, G., Kisman, D. & Lynam, T. (2000). Question answering by passage selection (multitext experiments for TREC-9), *Proceedings 9th Text Retrieval Conference (TREC-9)*, NIST Special Publication 500-249, 2000.

Cleverdon, C. W. (1991). The significance of the Cranfield tests on index languages. *Proceedings of the 14th annual international ACM SIGIR conference on Research and development in information retrieval*, 1991.

Corrada–Emmanuel, A., Croft, W.B., and Murdock, V. (2003). Answer Passage Retrieval for Question Answering, *CIIR Technical Report*, University of Massachusetts, 2003.

Croft, W. B. & Harper, D. (1979). Using probabilistic models of document retrieval without relevance information, *Journal of Documentation*, 35(4), pp.282-295.

Cui, H., Kan, M.-Y. & Chua, T.-S. (2007). Soft pattern matching models for definitional question answering. *ACM Transactions on Information Systems Journal.* 2007.

Cunningham, H., Maynard, D., Bontcheva, K. & Tablan, V. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics* (ACL'02).

Echihabi, A., Oard, D. W., Marcu, D. & Hermjakob, U. (2003). Cross-Language Question Answering at the USC Information Sciences Institute, *Working Notes for the CLEF 2003 Workshop*, 2003.

Ferrández, S., Toral, A., Ferrández, I., Ferrández, A. & Muñoz, R. (2009). Exploiting Wikipedia and EuroWordNet to solve Cross-Lingual Question Answering. *Inf. Sci.* 179, 2009.

Ferret, O., Grau, B., Hurault-Plantet, M., Illouz, G. & Jacquemin, C. (2001). Terminological variants for document selection and question answer matching, *Proceedings of the Association for Computational Linguistics, Workshop on Open-Domain Question Answering*, 2001.

Finkel, J. R., Grenager, T. & Manning, C. (2005). Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics* (ACL 2005).

Fleischman, M., Hovy, E. H., Echihabi, A. (2003). Offline Strategies For Online Question Answering: Answering Questions Before They Are Asked, *Proceedings of the Association for Computational Linguistics Conference*. 2003.

Forgy Charles. (1982). Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem. *Artificial Intelligence* 19, 1982.

Forner, P., Peñas, A., Alegria, I., Forăscu, C., Moreau, N., Osenova, P., Prokopidis, P., Rocha, P., Sacaleanu, B., Sutcliffe, R., Sang. E. T. K. (2008). Overview of the CLEF 2008 Multilingual Question Answering Track. *Working Notes for the CLEF 2008 Workshop*, 2008.

Gaizauskas, R., Greenwood, M. A., Hepple, M., Roberts, I., Saggion, H., & Sargaison, M. (2003). The University of Sheffield's TREC 2003 Q&A Experiments, *Online proceedings of the 2003 Text Retrieval Conference,* 2003.

Giampiccolo, D., Forner, P., Peñas, A., Ayache, C., Cristea, D., Jijkoun, V., Osenova, P., Rocha, P., Sacaleanu, B., Sutcliffe, R. (2007). Overview of the CLEF 2007 Multilingual Question Answering Track. *Working Notes for the CLEF 2007 Workshop*. 2007.

Glöckner, I. & Pelzer, B. (2008) The LogAnswer Project at CLEF 2008: Towards Logic-Based Question Answering. *Working Notes for the CLEF 2008 Workshop*, 2008.

Grosz, B. J., Weinstein, S. & Aravind K. J. (1995). Centering: A Framework for Modeling the Local Coherence of Discourse. *Computational Linguistics*, v.21 n.2, 1995.

Hearst, M. (1998). Automated Discovery of WordNet Relations, *WordNet: An Electronic Lexical Database,* Christiane Fellbaum (ed.), MIT Press, 1998.

Hamp, B. & Feldweg, H. (1997). GermaNet - a Lexical-Semantic Net for German. *Proceedings of ACL workshop Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*. 1997.

Harabagiu, S., Moldovan, D., Pasca, M., Mihalcea, R., Surdeanu, M., Bunescu, R., Girju, R., Rus, V. & Morarescu, P. (2000). FALCON: Boosting knowledge for Answer Engines, *The Ninth Text REtrieval Conference (TREC 9), NIST Special Publication*, 2000.

Harabagiu, S. & Lacatusu, F. (2004). Strategies for Advanced Question Answering. *Proceedings of the Workshop on Pragmatics of Question Answering at HLT-NAACL,* 2004.

Harabagiu, S. & Hickl, A. (2006). Methods for using textual entailment in open-domain question answering. *International Conference of the Association for Computational Linguistitics* (*ACL*), 2006.

Hartrumpf, S. (2005). Question answering using sentence parsing and semantic network matching. *Multilingual Information Access for Text, Speech and Images: Results of the Fifth CLEF Evaluation Campaign*, volume 3491 of Lecture Notes in Computer Science (LNCS), 2005.

Hartrumpf, S., Glöckner, I & Leveling, J. (2007). University of Hagen at QA@CLEF 2007: Coreference Resolution for Questions and Answer Merging. *Working Notes for the CLEF 2007 Workshop*, 2007.

Hartrumpf, S., Glöckner, I & Leveling, J. (2008). University of Hagen at QA@CLEF 2008: Efficient Question Answering with Question Decomposition and Multiple Answer Streams. *Working Notes for the CLEF 2008 Workshop*, 2008.

Hermjakob, U.(2001). Parsing and question classification for question answering, *Proceedings of the Workshop on Open-Domain Question Answering at ACL-2001*, 2001.

Hildebrandt, W., Katz, B. & Lin, J. (2004). Answering definition questions using multiple knowledge sources. *Proceedings of HLT-NAACL,* 2004.

Hovy, E., Gerber, L., Hermjakob, U., Junk, M. & Lin, C. (2001). Question Answering in Webclopedia, *The Ninth Text REtrieval Conference(TREC-9)*, 2001.

Ittycheriah, A., Franz, M., Zhu, W, J. & Ratnaparkhi, A. (2001). IBM's statistical question answering system. *Proceedings 9th Text Retrieval Conference (TREC-9)*, NIST Special Publication, 2001.

Jijkoun, V. & Rijke, de M., (2004). Answer selection in a multi-stream open domain question answering system. *Advances in Information Retrieval: 26th European Conference on IR research*, ECIR 2004.

Joho, H. & Sanderson, M. (2000). Retrieving Descriptive Phrases from Large Amounts of Free Text. *In 9th ACM conference on Information and Knowledge Management*, pages 180–186, 2000.

Katz, B. & Lin, J. 2003. Selectively using relations to improve precision in question answering. *Proceedings of the workshop on Natural Language Processing for Question Answering (EACL 2003)*, 2003.

Khalid, M., & Verberne, S. (2008). Passage Retrieval for Question Answering using Sliding Windows. *Proceedings of COLING 2008 IR4QA Workshop*. 2008.

Klein, D. & Manning, C. D. (2003). Fast Exact Inference with a Factored Model for Natural Language Parsing. *Advances in Neural Information Processing Systems 15 (NIPS 2002)*, 2003.

Kwok, K., Grunfeld, L., Dinsl, N. & Chan, M. (2000). TREC-9 cross language, web and question answering track experiments using PIRCS. *Proceedings of the Ninth Text REtrieval Conference (TREC-9)*, NIST Special Publication, 2000.

Lao, N., Shima, H., Mitamura, T. & Nyberg, E. (2009). Query Expansion and Machine Translation for Robust Cross-Lingual Information Retrieval, *Proceedings of NTCIR-7*, 2009.

Lee, Y.-H., Lee, C.-W., Sung, C.-L., Tzou, M.-T., Wang, C.-C., Liu, S.-H., Shih, C.-W., Yang R.-Y., & Hsu, W.-L. (2008). Complex Question Answering with ASQA at NTCIR 7 ACLIA. *Proceedings of NTCIR-7*, 2008.

Li, F., Zhang, X., Yuan, J. & Zhu, X. (2008). Classifying what-type questions by head noun tagging. *Proceedings of the 22nd International Conference on Computational Linguistics - Volume 1* (COLING '08), 2008.

Li, W. (2002). Question classification using language modeling, *Technical Report IR-259, Center for Intelligent Information Retrieval*, University of Massachusetts.

Li, X. & Roth, D. (2002) Learning Question Classifiers. *Proceedings of the International Conference on Computational Linguistics (COLING)*, 2002.

Lin, J. and Katz, K. (2003). Question answering from the web using knowledge annotation and knowledge mining techniques. *Conference on Information and Knowledge Management (CIKM)*, 2003.

Lita, L. V., Hunt, W. and Nyberg, E. (2004). Resource analysis for question answering. *The Association for Computational Linguistics Conference (ACL)*, 2004.

Lita, L. V., Rogati, M. & Barbonell, J. (2003). Cross Lingual QA: Modular Baseline, *Working Notes for the CLEF 2003 Workshop*, 2003.

Litkowski, K. C. (2004). Use of metadata for question answering and novelty tasks. *Proceedings of the eleventh Text Retrieval Conference (TREC 2003)*, 2004.

Lopez, V., Nikolov, A., Sabou, M., Uren, V. & Motta, E. (2010). Scaling up question-answering to linked data. *EKAW 2010 - Knowledge Engineering and Knowledge Management by the Masses*, 2010.

Magnini, B., Negri, M., Prevete, R. & Tanev, H.(2002). Is it the right answer? Exploiting web redundancy for answer validation. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL-2002)*, 2002.

Magnini, B., Romagnoli, S., Vallin, A., Herrera, J., Peñas, A., Peinado, V., Verdejo, F., Rijke, M. de. (2003), Creating the DISEQuA Corpus: a Test Set for Multilingual Question Answering, *Working Notes for the CLEF 2003 Workshop*, 2003.

Merkel A. & Klakow D. (2007). Comparing Improved Language Models for Sentence Retrieval in Question Answering. *Proceedings of Computational Linguistics in the Netherlands CLIN*, 2007.

Metzler, D. & Croft, W. B. (2005). Analysis of Statistical Question Classification for Fact-Based Questions. *Information. Retrieval* 8, 3, 2005.

Mikhailian, A., Dalmas, T. & Pinchuk, R. (2009). Learning foci for question answering over topic maps. *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers* (ACL-IJCNLP '09). 2009.

Miller, G. A. et al. (1993)**.** Five Papers on WordNet. *Technical Report, Cognitive Science Laboratory*, Princeton University, 1993.

Min, J., Jiang, J., Leveling, J. & Jones, G., J., F. (2010). DCU's Experiments for the NTCIR-8 IR4QA Task. *Procedings of NTCIR-8 Workshop Meeting*, 2010.

Moldovan, D., Harabagiu, S., Pasca, M., Mihalcea, R., Goodrum, R., Girji, R. & Rus, V. (2000). LASSO: A tool for surfing the answer net, *Proceedings 8th Text Retrieval Conference (TREC-8)*, NIST Special Publication, 2000.

Moldovan, D., Clark, C. & Harabagiu, S. (2003). COGEX: a logic prover for question answering. *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, 2003.

Mollá, D., Berri, J. & Hess, M. (1998). A real world implementation of answer extraction, *Proceedings 9th International Conference on Database and Expert Systems Applications,* Workshop on Natural Language and Information Systems (NLIS'98), 1998.

Mollá, D. & Gardiner, M. (2004). AnswerFinder - Question Answering by Combining Lexical, Syntactic and Semantic Information (2004). *Proceedings of ALTW04,* 2004.

Monz, C. & Rijke, M. de.(2001). Tequesta: The University of Amsterdam's textual question answering system, *Proceedings of Tenth Text Retrieval Conference (TREC-10)*, 2001.

Monz, C. (2004). Minimal Span Weighting Retrieval for Question Answering, *ACM SIGIR Workshop on Information Retrieval for Question Answering*, 2004

Negri, M., Tanev, H. & Magnini, B. (2003). Bridging Languages for Question Answering: DIOGENE at CLEF-2003, *Working Notes for the CLEF 2003 Workshop*, 2003.

Neumann, G. & Piskorski, J. (2002). A shallow text processing core engine. *Computational Intelligence* 18(3), 2002.

Neumann, G. & Sacaleanu, B. (2003). A Cross-Language Question/Answering-System for German and English, *Working Notes for the CLEF 2003 Workshop*, 2003.

Neumann, G. & Sacaleanu, B. (2006). Experiments on Cross-Linguality and Question-Type Driven Strategy Selection for Open-Domain QA. *CLEF 2005*. Lecture Notes in Computer Science, vol. 4022, 2006.

Nguyen, M. L., Nguyen, T. T. & Shimazu, A. (2007). Subtree mining for question classification problem. *Proceedings of the 20th international joint conference on Artificial intelligence* (IJCAI'07), 2007.

Nyberg, E. , Mitamura, T., Callan, J., Carbonell, J., Frederking, R., Collins-Thompson, K., Hiyakumoto, L., Huang, Y., Huttenhower, C., Judy, S., Ko, J., Kupsc, A., Lita, L. V., Pedro, V., Svoboda, D., & Durme, B. V. (2003). The Javelin question-answering system at TREC 2003: A multi strategy approach with dynamic planning. *Text Retrieval Conference (TREC)*, 2003.

Pinchak, C. & Lin, D. (2006). A probabilistic Answer Type Model. *European Chapter of the ACL*, 2006.

Pinchak, C., Lin, D. & Rafiei, D. (2009). Flexible answer typing with discriminative preference ranking. *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics* (EACL '09). 2009.

Plamondon, L. & Foster, G. (2003). Quantum, a French/English Cross-language Question Answering System, *Working Notes for the CLEF 2003 Workshop*, 2003.

Prager, J., Radev, D. & Czuba, K. (2001). Answering What-Is Questions by Virtual Annotation, *Proceedings of the Human Language Technology Conference*, HLT 2001.

Punyakanok, V., Roth, D. & Yih, W. (2004). Mapping dependency trees: An application to question answering. *The 8th International Symposium on Artificial Intelligence and mathematics (AI&Math 04)*, 2004.

Radev, D., Fan, W., Qi, H., Wu, H., & Grewal, A.(2002). Probabilistic question answering on the web, *WWW '02: Proceedings of the eleventh international conference on World Wide Web*, 2002.

Ravichandran, D. & Hovy, E. (2002). Learning surface text patterns for a question answering system, *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL-2002)*, 2002.

Ren, H., Ji, D. & Wan, J. (2010). WHU Question Answering System at NTCIR-8 ACLIA Task. *Proceedings of NTCIR-8 Workshop Meeting*, 2010.

Riezler, S., Vasserman, A., Tsochantaridis, I., Mittal, V. O. & Liu, Y. (2007). Statistical machine translation for query expansion in answer retrieval. *Association for Computational Linguistics*, 2007.

Roberts, I. (2002). Information retrieval for question answering. *Master's thesis, University of Sheffield*, 2002.

Robertson, S. E. (1977). The probabilistic ranking principle in IR, *Journal of Documentation*, 33, 1977.

Robertson, S. E. & Sparck Jones, K .(1976). Relevance weighting of search terms, *Journal of the American Society for Information Science*, 27(3), 1976.

Robertson, S. E., Walker, S. Hancock-Beaulieu, M. & Gatford, M.(1994a). Okapi at TREC-3. *Proceedings of TREC-3*, 1994.

Robertson, S. E. & Walker, S.(1994b). Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval, *Proceedings of the 17th annual international ACMSIGIR conference on Research and Development in Information Retrieval*, 1994.

Sacaleanu, B., Neumann, G. & Spurk, C. (2007). DFKI-LT at QA@CLEF 2007. *Working Notes for the CLEF 2007 Workshop*, 2007.

Sacaleanu, B., Neumann, G. & Spurk, C. (2008). DFKI-LT at QA@CLEF 2008. *Working Notes for the CLEF 2008 Workshop*, 2008.

Saias, J. & Quaresma, P. (2008). The Senso Question Answering System at QA@CLEF 2008. *Working Notes for the CLEF 2008 Workshop*, 2008.

Salton, G., Wong, A. & Yang, C. (1975). A vector space model for automatic indexing, *Communications of the ACM*, 18(11), 1975.

Scott, S. & Gaizauskas, R. (2001). University of Sheffield TREC-9 Q&A System. *Proceedings 9th Text Retrieval Conference (TREC-9)*, NIST Special Publication 2001.

Shima, H. & Mitamura, T. (2010). Bootstrap Pattern Learning for Open-Domain CLQA. *Proceedings of NTCIR-8 Workshop Meeting*, 2010.

Shimizu, K. & Akiba, T. (2005). Bi-directional Cross Language Question Answering using a Single Monolingual QA System, *Proceedings of NTCIR-5 Workshop Meeting*, December 6-9, 2005.

Soubbotin, M. & Soubbotin, S. (2001). Patterns of potential answer expressions as clues to the right answer, *Proceedings of the Tenth Text Retrieval Conference (TREC 2001)*, NIST Special Publication 2001.

Spink, A., & Ozmutlu, H. C. (2001). Ask Jeeves query analysis: What do people ask? *ASIST 2001: Annual Conference of the American Society for Information Science and Technology*, 2001.

Srihari, R. & Li,W. (2000). Information extraction supported question answering, *Proceedings 8th Text Retrieval Conference (TREC-8)*, NIST Special Publication 2000.

Surdeanu, M., Ciaramita, M. & Zaragoza, H. (2008) Learning to Rank Answers on Large Online QA Collections. *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL-2008),* 2008.

Sutcliffe, R., Gabbay, I. & O'Gorman, A. (2003). Cross-Language French-English Question Answering using the DLT System at CLEF 2003, *Working Notes for the CLEF 2003 Workshop*, 2003.

Suzuki, J., Taira, H., Sasaki, Y. & Maeda, E. (2003). Question classification using HDAG kernel, *Proceedings of the ACL 2003 Workshop on Multilingual Summarization and Question Answering*, 2003.

Téllez, A., Juárez, A., Hernández, G., Denicia, C., Villatoro, E., Montes, M. & Villaseño, L. (2007). INAOE's Participation at QA@CLEF 2007. Working Notes for the CLEF 2007 Workshop.

Tiedemann, J., & Mur, J. (2008). Simple is Best: Experiments with Different Document Segmentation Strategies for Passage Retrieval. *Proceedings of COLING 2008 IR4QA Workshop*. 2008.

Verberne, S., Boves, L., Oostdijk, N. & Coppen, P.-A. (2008). Using syntactic information for improving why-question answering. *Proceedings of the 22nd International Conference on Computational Linguistics - Volume 1 (COLING '08),* Vol. 1. 2008.

Wang, R., & Neumann, G. (2007). DFKI–LT at AVE 2007: Using Recognizing Textual Entailment for Answer Validation, *Working Notes for the CLEF 2007 Workshop*.

Wang, R., & Neumann, G. (2008). Information Synthesis for Answer Validation, *Working Notes for the CLEF 2008 Workshop*.

Xu, J., Cao, Y., Li, H. & Zhao, M. (2005). Ranking definitions with supervised learning methods. *In WWW2005*, pages 811–819, 2005.

Yarowsky, D. (1995). Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. *Proceedings of the 33rd Meeting of the Association for Computational Linguistics*, 1995.

Zajac, R. (2001). Towards ontological question answering, *Proceedings of the Association for Computational Linguistics,* Workshop on Open-Domain Question Answering, 2001.

Zhang, D. & Lee, W. (2003). Question classification using support vector machines, *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2003.

Resume

Personal information

First name / Surname	**Bogdan Eugen Sacaleanu**
Address	Im Heimeck 13
Telephone	+496819104797
E-mail	bogdan.sacaleanu@gmail.com
Website	http://www.dfki.de/~bogdan
Nationality	Romanian, German
Date of birth	19.03.1975
Gender	male

Work experience

Dates	**from**: 09 / 2000 **to**: 01 / 2012
Occupation or position held	Researcher / Software Engineer
Main activities and responsibilities	- conversion of requirements into architecture and design; - development of the logical and physical structure of solutions and their components; - creation and testing of software implementations; - implementation of graphical user interfaces; - maintenance of distributed architectures and integration of legacy systems; - SOA-enablement of existing and legacy components; - development of cross-language search engines for general and medical domain; - development of Natural Language Interface / Question Answering systems - development of tools for the Semantic Web (ontology population) - development of Natural Language Interfaces tools

Research Interests	Information Access (Information Retrieval & Question Answering) Natural Language Understanding for Multimodal Interaction and Dialogue Management Knowledge Discovery in Texts (data mining, information extraction) Textual Entailment Lexical Semantics Machine Learning
Engineering Interests	Analysis & Design (OOAD, UML) Software Lifecycle (OpenUP) Component Plugin Framework (OSGi) Distributed Computing (Hadoop)
Projects	THESEUS-ORDO – project on information access based on semantic annotations THESEUS-CTC – project on the semantic infrastructure for Web 3.0 for easy access to the structured global knowledge and to novel services QALL-ME – project on Question Answering from structured knowledge SMARTWEB – project on semantic Web Services (Question Answering) for mobile devices QUETAL – project on Question Answering over semantic data MUCHMORE – project on cross-language Information Access in the medical domain MULINEX – project on cross-language Information Access on the WWW
Name and address of employer	German Research Center for Artificial Intelligence (DFKI) Stuhlsatzenhausweg 3 66123, Saarbrücken Germany
Type of business or sector	Research & Development of Artificial Intelligence Applications

Education and training

Dates	**from**: 09 / 2003 **to**: 09 / 2011
Title of qualification pursued	PhD (Computational Linguistics)
Principal subjects/occupational skills covered	Cross-language Question Answering

Name and type of organisation providing education and training	Saarland University Department of Computational Linguistics and Phonetics Saarbrücken, Germany
Dates	**from**: 09 / 1997 **to**: 03 / 2001
Title of qualification awarded	Diplom/Master of Science (Computational Linguistics)
Principal subjects/occupational skills covered	Natural Language Processing Syntactic theory Text understanding Semantics of Language Speech Recognition
Name and type of organisation providing education and training	Saarland University Department of Computational Linguistics and Phonetics Saarbrücken, Germany
Dates	**from**: 09 / 1993 **to**: 06 / 1997
Title of qualification awarded	Bachelor of Science (Computer Science)
Principal subjects/occupational skills covered	Software Engineering Object Oriented Programming Databases Computer Architecture and Operating Systems Computer Networks Artificial Intelligence
Name and type of organisation providing education and training	Alexandru Ioan Cuza University Faculty of Computer Science Iasi, Romania

Personal skills and competences

Mother tongue(s) **Romanian**

Other language(s) **English, German**

Self-assessment	**Understanding**	**Speaking**	**Writing**
English	Excellent	Excellent	Excellent
German	Excellent	Excellent	Very good

Social skills and competences	Communication skills (1) acquired during teaching of courses at the university, supervising undergraduate students and working close to project partners, (2) developed giving oral presentations on the results of my research both within the company and at international conferences, (3) tested in giving demos of applications before shareholders and establishing contacts at fares (like CEBIT). I am working in a multi-cultural environment, with daily contact to people of different cultures, and I am interested in finding out more about their way of thinking and their traditions.
Organisational skills and competences	I have very good organizational skills, knowing to structure complex circumstances, to plan assignments, to set priorities, to estimate time and resources, to provide accurate information, having high self-organizing ability and being familiar with management tasks. Co-organizer of evaluation forums for language technology systems (CLEF). Reviewer of research and technology-related papers for conferences. Good skills in coordinating groups of work, by teaching lab activities and supervising research assistants. Taking responsibility for work packages within projects and coordinating partners for integrating systems and submitting deliverables.
Technical skills and competences	Programming: Java (expert), C++ (competent), Perl (advanced beginner) Server Programming: J2EE (proficient) Web App Frameworks: Java Server Faces (proficient) Design: UML, object-oriented analysis & design (OOAD), business process modeling (BPEL), GOF Design Patterns, GRASP patterns Distributed Computing: Hadoop (advanced beginner) Search Engines: Apache Lucene, SOLR Business Rules Management Systems: DROOLS Information Management Frameworks: UIMA, SMILA/Eclipse Machine Learning Tools: WEKA, Mahout NLP Tools: LingPipe, OpenNLP, Stanford NLP, BALIE Dependency Parsers: MALT, Stanford Parser Semantic Web: Jena, DBpedia, Linked Data

Computer skills and competences	Knowledge of LAN networks setup.
Other skills and competences	Ability to work under pressure, I am proactive and highly motivated, spontaneous and reliable. I work very well in teams, I enjoy meeting new people and staying in contact with them, and I can take responsibility and assume risk.