Automated Question Answering: Review of the Main Approaches

Andrea Andrenucci, Eriks Sneiders
Stockholm University/ Royal Institute of Technology, Sweden
{andrea, eriks}@dsv.su.se

Abstract

Automated Question- Answering aims at delivering concise information that contains answers to user questions. This paper reviews and compares three main question-answering approaches based on Natural Language Processing, Information Retrieval, and question templates, eliciting their differences and the context of application that best suits each of them.

1. Introduction

The question-answering (QA) paradigm, i.e. the process of retrieving precise answers to natural language (NL) questions, was introduced in late 1960-ies and early 1970-ies within the framework of NL understanding. It was mainly implemented in interfaces to problem solving systems for specific domains. The advent of WWW has reintroduced the need for user-friendly querying techniques that reduce information overflow, and poses new challenges to the research in automated QA. Important QA application areas are information extraction from the entire Web ("intelligent" search engines), online databases, and inquiries on individual websites.

This paper discusses three main research approaches utilized in automated QA. Natural Language Processing (NLP) maps user questions into a formal world model and ensures the most reliable answers. Information Retrieval (IR) powered QA, together with NLP, focus on fact extraction from large amounts of text. Template-based QA matches user questions to a number of templates that cover often queried parts of the knowledge domain. The next three sections review the approaches; the last section compares them eliciting the context of application that best suits each of them.

2. Natural Language Processing

NLP techniques are used in applications that make queries to databases, extract information from text, retrieve relevant documents from a collection, translate from one language to another, generate text responses, or recognize spoken words converting them into text. From the QA point of view, NL interfaces to databases [1] and information extraction [28] are most interesting. A common feature of NLP systems is that they convert text input into formal representation of meaning such as logic (first order predicate calculus), semantic networks, conceptual dependency diagrams, or frame-based representations [16, p. 502]. Since the early days of NLP, QA systems simulated human intelligence within the NL understanding research field. They worked as NL front-end to databases [48], dialogue systems [47] or story comprehension systems [21]. QA systems were limited to specific and narrow domains such as algebra, astronomy and natural science. Figure 1 on the next page shows the architecture of a typical NLP QA system (the figure originally from [1]). Two main areas, the linguistic front-end and the domain-dependent knowledge, are clearly identified in the architecture.

The linguistic front-end parses and analyses the user input in NL before translating it into an intermediate logical query. The logical query is then converted into a database query language, supported by the underlying database management system. The domain dependent knowledge contains information specific for the domain of interest: a lexicon and a world model. The lexicon contains admissible vocabulary words from the knowledge domain, often divided in word stems and word affixes (i.e. suffixes or prefixes), plus a representation of how they fit together [16, p. 66]. The world model describes the structure of the domain of interest, i.e. the hierarchy of classes of the domain objects, plus the properties and the constraints that characterize the relationship between them. Traditionally the world model is expressed through ontologies – explicit specifications of the conceptualization of the knowledge base [14].

NLP-based QA systems may utilize machine learning to improve their syntax rules [25], lexicon [43], semantic rules [50], or the world model [46].

For a detailed overview of NL front-ends to databases see [1], [8]. Most traditional NL interfaces that do thorough language processing remain research prototypes as their commercially added value does not cover the costs. Some commercially available systems were Symantec's Q&A, IBM's Language Access [2] and Bim's Loqui [30].

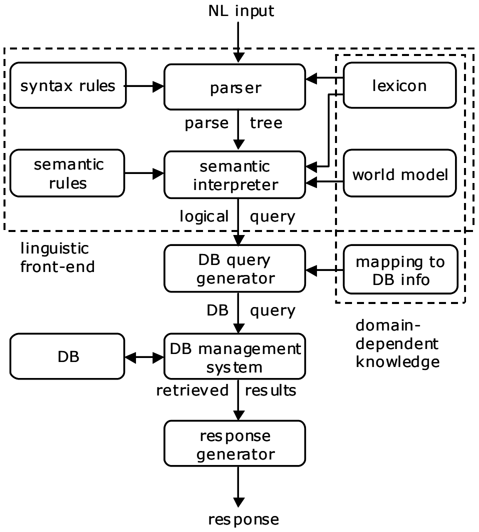

Figure 1. Architecture of a typical NLP QA system, from [1]

Information Extraction (IE) from large amounts of unstructured text became especially important along with the expansion of WWW. It has moved the interest of the QA community to a more shallow approach to NLP [41, p.8]. Shallow NLP does not imply text understanding, i.e. semantic analysis of NL input. Instead, it focuses on extracting text chunks matching patterns or entities, thus providing the answer to user questions [32]. For instance in a question like "Who won the 1998 Nobel Peace Prize?", the presence of the interrogative pronoun "who" implies the extraction of an entity of type "person name" associated with the keywords, "won", "1998", "Nobel", "peace" and "prize" (see [33] about IE in QA). In some cases machine learning has been applied in order to learn patterns to extract specific phrases [11]. The shallow NLP approach is more domain-independent than traditional NLP, but requires the answer to be explicitly present in the text [44]. This lighter approach is implemented in Information Extraction systems such as CIRCUS [22] and LaSIE [12], which recognize entities and their relationship in running text, with the purpose of storing them in a structured format for data mining or question answering.

3. Information Retrieval

Information Retrieval deals with the representation, storage, organization of and access to information items. Automated QA is related to IR as users submit queries in order to find answers to questions they have in mind [15]. IR systems are traditionally seen as document retrieval systems, i.e. systems that return documents that are relevant to the user's information need, but that do not supply direct answers. A further step towards the QA paradigm is the development of document retrieval systems into passage retrieval systems [35], which focus on retrieving text passages rather than entire documents. Passage retrieval is now a standard component of modern IR-based QA applications (IR QA), such as [37], [31] and [7].

SMART [34] is one of the most significant systems for IR research: its techniques, such as the vector-space model, have become standards in the IR community. SMART is still used in modern systems such as the ones described in [4] and [5]. FAQ Finder [4] is among the first systems for retrieving Frequently Asked Questions (FAQ) from text files. It utilizes traditional statistical methods such as term frequency and inverted document frequency, plus semantic knowledge of English words provided by WordNet [26].

4. Information Retrieval meets NLP

The IR and NLP research communities have common interests [15]. The joint interest towards QA has increased during the latest years as a rising number of technically inexperienced people have gained access to online information sources. A number of conferences, such as ANLP-NAACL [23] and TREC [44] have promoted this research. The Text REtrieval Conferences (TREC) aim at comparing IR systems implemented by academic and commercial research groups. The systems participating at various TREC tracks run pre-selected queries and retrieve text documents. The results are evaluated manually. The QA track was introduced in TREC-8 (1999) with the purpose of testing retrieval of short answers to fact-based questions instead of ranked lists of documents. Voorhees [45] summarizes the main steps taken by the TREC QA systems as follows:

- Distinguishing the expected answer type from the user question. For example questions starting with "who" imply the search for "person names" as answers.

- Reducing the text corpus to a smaller set of relevant documents or passages that most likely contain the answer(s) to the question.
- Ranking the passages containing the answer(s) and extracting the answer(s) to be presented to the user.

Some TREC participants use shallow NLP based on pattern matching ([3], [37], [49]) while others rely on heavier NLP techniques ([36], [27], [9]). The best performing system within the two latest TREC, Power Answer [27], had reached 83% accuracy in TREC 02 and 70% in TREC 03. Accuracy is defined as the fraction of answers judged correct. Starting from 2001, TREC's QA track uses the Web as the primary information source. Systems such as the ones described in [3], [7] exploit the Web data redundancy. The advantages of data redundancy have been listed in [3] and can be summarized as follows: 1) The occurrence of multiple linguistic formulations of similar statements increases the chances of finding an answer that matches the submitted question 2) Answers that occur more often on the Web are usually more reliable.

Mulder [20] is one of the first open domain Web-based QA systems. Mulder is similar to the systems evaluated at TREC and combines both IR and NLP techniques. The system pre-process user questions in NL with a syntactic parser and classify them into nominal, numerical and temporal categories. The result of the syntactic parsing is translated into a series of keywords to be submitted to Google. Mulder then parses the retrieved Web pages and extracts relevant snippets, generating a list of possible candidate answers from these snippets. The candidates are then scored according to the proximity of relevant keywords, whose relevance is based on a common IR metric, inverted document frequency.

5. Template based Question Answering

Template-based QA extends the pattern matching approach of NL interfaces to databases [1, p.14] where the intelligence of the system is embodied in a collection of manually created question templates. The START system, which has answered more than a million questions since 1993, makes use of knowledge annotations as "computer-analyzable collections of natural language sentences and phrases that describe the content of various information segments" [17]. An annotation mimics the structure "subject-relationship-object". START matches a user question to the annotation entries on both the word level (using additional lexical information about synonyms, IS-A trees, etc.) and the structure level (rules for paraphrased arguments of verbs, nominalization, etc.). When the user question matches an annotation entry, START follows the pointer to the information segment tagged by the annotation and returns it as the answer. Omnibase [18] is a virtual database on top of the START's knowledge annotations that comprises a number of online semi-structured data sources, such as CIA Fact Book (Country-Information-At system, www.cia.gov/cia/publications/factbook/) or Internet Movie Database (www.imdb.com). Omnibase adopts a stylized "object-property-value" data model where data sources have objects, which have properties. With the help of Omnibase, START translates questions into a structured request for the values of these properties. Each data source has a wrapper that translates Omnibase's requests into local queries. Sneiders [39] presents a similar approach for querying databases. The system operates question templates, which have open concepts to be filled with data instances, mapped into the conceptual model of the database. Each question template has a representative set of keyword expressions similar to regular expressions [38] that match relevant user questions, and it is paired with a query template – an unfilled database query. A question template is viewed as a predicate with variable and fixed parameters:

$\exists\ data_1,\ldots data_n\mathrm{:Q(fixed_1,\ldots fixed}_m\mathrm{,variable_1,\ldots variable}_n)$

During the process of matching a template to a user question, the fixed parameters ($\mathrm{fixed_1,\ldots fixed_m}$) are bound to the user question. If there are database data instances ($\mathrm{data_1,\ldots data_n}$) that fit the variable parameters ($\mathrm{variable_1,\ \ldots variable_n}$) and make the statement Q true, then these data instances constitute the answer. This is an n-dimensional case of looking for the Omnibase-kind of property values.

The NL annotations of START and Omnibase, and the question templates share the same feature – they are labels, attached to pieces of answer information, that mimic the structure of NL queries and map this structure into the underlying data model. The answer can be static text, data returned by an SQL query or a multimedia presentation.

The semantic web is a promising source of answers [19]. Sneiders [40, pp. 253-269] proposes a generic model of template-based QA that shows the relations between a knowledge domain, its conceptual model, structured databases, question templates, user questions, and describes about 24 constituents of template-based QA.

Individual question templates are autonomous although they may profit from a common knowledge base. Synonymy and linguistic ambiguities may be resolved locally within the context of a template [38, p. 102]. Question templates embody the major part of the "intelligence" of the system and therefore are created manually; automatic generation of the templates would require another manually created knowledge base. Machine learning may help, but we haven't found any evidence of extensive use of machine learning in creating question templates. A fine-tuned system can reach as high recall/precision levels as 81/98 and 98/93 percent [40, p.154, 229]. Lin [24] has noted that template-based QA is effective because the distribution of user queries follows Zipf's law – a small fraction of question types account for a large number of questions asked. Lin presents a graph that plots the number of question types into the number of questions they match. 50 question types account for more than 45% of TREC-2001 questions. Lin [24] proposes also the federated and distributed approaches to QA on the Web combined into a single system. According to the federated approach, a number of online databases are covered by question templates. Omnibase is an example of a federated-approach system. IR QA is the technique for the distributed approach.

Although there hasn't been much academic research done in template-based QA, the approach has become commercially successful. Ask Jeeves (www.ask.com), which licensed its technology from START, became the first large scale question-answering system, until it moved to a more traditional search engine business. Kiwilogic (www.kiwilogic.de) offers chat robots that organize question templates in a tree structure, which allows maintaining a human-like dialogue. The robot remembers the previously retrieved question template and its subject, which enabled so called elliptic questions with implied subject. QuickAsk (www.askology.com) is a collection of systems for question answering on individual websites using structured databases and FAQ lists. An installation on www.trygghansa.se (December 2004) combines the answers with keyword search results, which puts forward most relevant information while rendering an overview of related information not covered by question templates.

6. Differences between the approaches

To our knowledge, no formal comparison of the three QA approaches has been done so far. The comparison presented here is based on judgments on individual publications and analysis of the features described in the previous sections. This comparison can be seen as a starting point for further discussion on QA approaches and better development of future systems. We have chosen the following criteria: quality and source of the answers, size of the domain, domain and language portability.

NLP-based QA provides the most reliable human-computer dialogue and answers, which is achieved by the means of mapping text into a formal semantic real-world representation. The primary focus of NLP has been natural language interfaces for databases and expert systems in a narrow domain. Molla et al. [28] show that narrow and technical domains require thorough (deep) NLP as opposed to data intensive IR QA. The knowledge base of an NLP system poses portability difficulties. Figure 1 shows six components (linguistic front-end) that change when the input language changes, and three components (domain knowledge) that change when the knowledge domain changes. The changes require involvement of qualified personnel – programmers, knowledge engineers, and database administrators [1, pp. 25-27]. Winewater [46] increases the portability using two different lexicons, a domain dependent and a domain independent one. Chu and Meng [6] improve portability with the help of a semantic graph that represents the structure of the database, keywords manually assigned to the nodes and attributes in the graph. This is, in essence, a move towards the template-based approach.

IR QA, enhanced with shallow or deep NLP techniques, focuses on fact retrieval from a large text corpus. Systems that exploit shallow techniques do not comprehend either user questions or the text. Data redundancy – a number of similar statements that contain the answer – in a large corpus and use of shallow NLP techniques increase the chance of finding the right answer without any guarantee that the answer is correct. Shallow NLP techniques have also been largely used in the extraction of definition answers in TREC 03 (e.g. [13] and [42]). IR enhanced with deep NLP techniques overcomes the "understanding" deficiency of the shallow approach. For instance in Power Answer [27], candidate answers and user questions are semantically represented in logic form and compared. A logic proof based on abductive justifications is then performed among the valid answers. The world model is provided both by WordNet ontologies [26] and lexical chains [29], i.e. groups of semantically related words that link together two concepts. The commercial version of this system requires however a special ontological module that extends the WordNet ontology with domain specific concepts.

IR techniques are generally considered language and domain independent. Nevertheless, the amount of NLP applied in order to increase the quality of the answers influences the portability of the system. Applying deep NLP techniques on open domains and large corpus of data is both compute intensive and time-consuming [10].

Template-based QA does not process text. Like IR enhanced with shallow NLP, it presents relevant information without any guarantee that the answer is correct. The answer can be anything attached to question templates – static text, data returned by an SQL query, multimedia presentations. It is advisable to combine a template-based QA system with other search techniques to cover less frequent user questions not foreseen by the templates. Question templates conform user questions in a certain language and a certain knowledge domain; therefore they are not language or domain portable. Most domains overlap, therefore individual autonomous templates or their parts can be reused in a new domain. It does not take highly qualified labour to create and maintain a question template if the core of the template is keyword-based regular expressions [38]. Easy maintenance of the templates helps Kanisa advocate the concept of pre-packaged knowledge (white-papers on www.kanisa.com). Kanisa sells customizable template-based knowledge bases designed for different businesses.

Table 1 summarizes the three QA approaches from the application point of view. On a large domain such as the *entire Web*, IR QA is most appropriate. Thorough NLP is technically possible, but the costs of building a formal semantic representation of information on the entire Web are rather high. Ask Jeeves tried to cover the Web with a large number of question templates. The coverage gaps were considerable, therefore question answering was supported by keyword search. Question templates on the Web are most useful if linked to online databases (the federated approach, [24]).

Small domains (e.g., a scientific discipline) fit NLP and template-based QA (e.g., 200 templates cover 50% of all queries). Both techniques can form an interface of a *structured database* as well. *Fact extraction from text* is done by NLP (small domains) and IR QA (large domains), but not by template-based QA, since this approach does not process text.

Table 1. Comparison of the main approaches

	Thorough NLP	IR & NLP	Templates
Entire Web	No	Yes	Yes
Structured data	Yes	No	Yes
Facts from text	Yes	Yes	No
Advice-giving	Yes	No	Yes
Reliability %	Close to 100	Accuracy > 70	Recall > 80
Small domains	Yes	No	Yes

Advice-giving means answering questions without focusing on facts, e.g. "how to do …" type of questions. FAQ answering usually implies advice-giving. NLP and template-based QA are appropriate for that, while experiments within IR QA did not yield high quality [4].

In conclusion there is no best or worst QA approach. Each approach has its own niche, application environment, and tasks. If the quality of the answers is crucial, NLP should be applied. If facts need to be extracted from text, IR QA should be used. Limited domains, step-by-step growing knowledge bases (e.g. Ask Jeeves), or lack of highly qualified labour may take advantage of the template-based approach. Future research trend should involve the combination of the approaches into one single system and adapt the techniques to the application domain.

7. References

[1] I. Androutsopoulos, G. D. Ritchie, P. Thanisch, "Natural Language Interfaces to Databases: An Introduction", Journal of Natural Language Engineering, 1 (1), 1995.

[2] J.L. Binot et al., "Natural language interfaces: a new philosophy", Sun Expert Magazine, 1991.

[3] E. Brill, J et al., "Data-intensive question answering", in D. Harman and E. Voorhees (eds): Proc of TREC 2001, NIST, Gaithersburg, USA, 2001.

[4] R. Burke et al., "Question Answering from Frequently Asked Question Files: Experiences with the FAQ Finder System", AI Magazine, 18 (2), 1997, pp. 57-66.

[5] C. Cardie et al., "Examining the Role of Statistical and Linguistic Knowledge Sources in a General-Knowledge Question-Answering System", Proc. of ANLP-2000, ACL press, USA, 2000, pp. 180–187.

[6] W. Chu, and F. Meng, *Database Query Formation from Natural Language using Semantic Modeling and Statistical Keyword Meaning Disambiguation*, Technical Report 990003, UCLA, USA.

[7] C. Clarke et al., "Web reinforced Question Answering", in D. Harman and E. Voorhees (eds): Proc of TREC 2001, NIST, Gaithersburg, USA, 2001.

[8] A. Copestake and K. Sparck Jones, "Natural language interfaces to databases", The Knowledge Engineering Review, 5 (4), 1990, pp. 225-249.

[9] A. R. Diekema et al., "CNLP at the TREC-2002 Question Answering Track", in L.P. Buckland and E. Voorhees (eds): Proc. of TREC 2002, NIST, Gaithersburg, USA, 2002.

[10] O. Ferret et al., "Document Selection Refinement based on linguistic features for QALC, a question answering system", Proc. of RANLP 2001, John Benjamin, USA, 2001.

[11] D. Freitag, "Toward General-purpose learning for information extraction", in Proc. of ACL/COLING´98, ACL press, USA, 1998.

[12] R. Gaizauskas and Y. Wilks, "Information Extraction: Beyond Document Retrieval", Journal of Documentation, 54 (1), 1998, pp. 70-105.

[13] R. Gaizauskas et al., "The University of Sheffield's TREC 2003 Q&A Experiment", in L.P. Buckland and E. Voorhees (eds): Proc of TREC 2003, NIST, Gaithersburg, USA, 2003.

[14] T.R. Gruber, "A translation approach to portable ontology specifications", Knowledge Acquisition, 5 (2), 1993.

[15] L. Hirschman and R. Gaizauskas, "Natural language question answering: the view from here", Natural Language Engineering, 7 (4), 2001, pp. 275-300.

[16] D. Jurafsky and J.H. Martin, *Speech and language processing*, Prentice Hall, NJ, USA, 2000.

[17] B. Katz, "Annotating the World Wide Web using natural language", Proc. of RIAO 97, MCGill University, Canada, 1997.

[18] B. Katz et al., "Omnibase: Uniform access to heterogeneous data for question answering", Proc. of NLDB'2002, Springer-Verlag, Germany, 2002.

[19] B. Katz, J. Lin, and D. Quan, "Natural Language Annotations for the Semantic Web", Proc. of ODBASE 2002, Springer-Verlag, Germany, 2002.

[20] C. Kwok, O. Etzioni, D. Weld, "Scaling Question Answering to the Web", ACM Trans. on Information Systems, 19 (3), 2001, pp. 242-262.

[21] W. Lehnert, "A conceptual theory of QA", Proc. of IJCAI-77, William Kaufmann, USA, pp. 158-164.

[22] W. Lehnert et al., "Umass/Hughes: Description of the CIRCUS system as User for MUC-5", Proc. of MUC-5, Morgan Kaufman, USA, 1993, pp. 277-291.

[23] M. Light et al., *Proceedings of ANLP-NAACL 2000*, ACL press, USA, 2000.

[24] J. Lin, "The Web as a Resource for Question Answering: Perspective and Challenges", Proc. of LREC 2002.

[25] R.M. Losee, "Learning Syntactic Rules and Tags with Genetic Algorithms for Information Retrieval and Filtering", Information Processing & Management*,* 32 (2), 1996, pp. 185-197.

[26] G.A. Miller, "WordNet: A Lexical Database for English", Comm. of the ACM, 38 (11), ACM press, USA, 1995.

[27] D. Moldovan et al., "LCC Tools for Question Answering", in L.P. Buckland and E. Voorhees (eds): Proc of TREC 2003, NIST, Gaithersburg, USA, 2003.

[28] D. Molla et al., "NLP for Answer Extraction in Technical Domains", Proc. of EACL 2003, Morgan Kaufmann, USA, 2003.

[29] J. Morris and G. Hirst, "Lexical cohesion computed by Thesaural Relations as an indicator of the Structure of Text", Computational Linguistics, 17 (1), 1991, pp. 21-48.

[30] N. Ott, "Aspects of the Automatic Generation of SQL statements in a Natural Language Query Interface", Information Systems, 17 (2), 1992, pp. 147-159.

[31] D. Ravichandran and EH. Hovy, "Learning Surface Text Patterns for a Question Answering System", Proc. of ACL-2002, ACL press, USA, 2002.

[32] E. Riloff and W. Lehnert, "Information Extraction as a Basis for High-Precision Text Classification", ACM Transactions on Information Systems 12 (3), 1994, pp. 296-333.

[33] S. Rohini and L. Wei, "Information Extraction Supported Question Answering", in D. Harman and E. Voorhees (eds): Proc. of TREC 1999, NIST, Gaithersburg, USA, 1999.

[34] G. Salton, *The SMART Retrieval system – Experiments in Automatic document processing,* Prentice Hall Inc., NJ, USA, 1971.

[35] G. Salton, J. Allan and C. Buckley, "Approaches to passage retrieval in full text information systems", in Proc. of SIGIR´93, ACM Press, N Y, USA, 1993.

[36] S. Scott and R. Gaizauskas, "University of Sheffield TREC-9 Q & A System", in D. Harman and E. Voorhees (eds): Proc. of TREC 2000, NIST, Gaithersburg, USA, 2000.

[37] M. M. Soubbotin and S.M. Soubbotin, "Use of Patterns for Detection of Answer Strings: A Systematic Approach", in L.P. Buckland and E. Voorhees (eds): Proc. of TREC 2002, NIST, Gaithersburg, USA, 2002.

[38] E. Sneiders, "Automated FAQ Answering: Continued Experience with Shallow Language Understanding", Proc. of the 1999 AAAI Fall Symp., AAAI press, USA, 1999, pp. 97-107.

[39] E. Sneiders, "Automated Question Answering Using Question Templates that Cover the Conceptual Model of the Database", Proc. of NLDB'2002, Springer-Verlag, Germany. 2002.

[40] E. Sneiders, *Automated Question Answering: Template-Based Approach*, PhD thesis, Stockholm University / KTH press, Sweden, 2002.

[41] K. Sparck Jones, "Natural Language Processing: a historical review", Current Issues in Computational Linguistics, 2001.

[42] R. Sutcliffe et al., "Question Answering using the DLT System at TREC 2003", in L.P. Buckland and E. Voorhees (eds): Proc. of TREC 2003, NIST, Gaithersburg, USA, 2003.

[43] C. A. Thompson, and R. J. Mooney, "Automatic construction of semantic lexicons for learning natural language interfaces", Proc. of the 1999 AAAI Fall Symposium, AAAI press, USA, 1999.

[44] E.M. Voorhees, "The TREC Question Answering Track", Natural Language Engineering, 7 (4), 2001, pp. 361-378.

[45] E. M. Voorhees, "Overview of the TREC 2003 Question Answering Track", in L.P. Buckland and E. Voorhees (eds): Proc. of TREC 2003, NIST, Gaithersburg, USA, 2003.

[46] W. Winiwater, "An Adaptive Natural Language Interface Architecture to Access FAQ Knowledge Bases", Proc. of the 4th Int. Conf. on Applications of N L to Information Systems, 1999.

[47] T. Winograd, "Understanding Natural Language", Cognitive Psychology, 3 (1), 1972.

[48] W.A Woods, R.M. Kaplan and B. Nash-Webber, *The lunar sciences natural language information system*, BBN Rep. 2378, Bolt Beranek and Newman, Cambridge, Mass., USA, 1977.

[49] L. Wu et al., "FDUQA on TREC2003 QA task", in L.P. Buckland and E. Voorhees (eds): Proc. of TREC 2003, NIST, Gaithersburg, USA, 2003.

[50] J. M Zelle and R. J. Mooney, "Learning to parse database queries using inductive logic programming", Proc. of the 13th Nat. Conf. on AI 1999.