

QUALIFIER: Question Answering by Lexical Fabric and External Resources

Hui Yang

Department of Computer Science
National University of Singapore
3 Science Drive 2, Singapore 117543
yangh@comp.nus.edu.sg

Tat-Seng Chua

Department of Computer Science
National University of Singapore
3 Science Drive 2, Singapore 117543
chuats@comp.nus.edu.sg

Abstract

One of the major challenges in TREC-style question-answering (QA) is to overcome the mismatch in the lexical representations in the query space and document space. This is particularly severe in QA as exact answers, rather than documents, are required in response to questions. Most current approaches overcome the mismatch problem by employing either data redundancy strategy through the use of Web or linguistic resources. This paper investigates the integration of lexical relations and Web knowledge to tackle this problem. The results obtained on TREC11 QA corpus indicate that our approach is both feasible and effective.

1 Introduction

Open domain Question Answering (QA) is an information retrieval paradigm that is attracting increasing attention from the information retrieval (IR), information extraction (IE), and natural language processing (NLP) communities (AAAI Spring Symposium Series 2002, ACL-EACL 2002). A QA system retrieves concise answers to open-domain natural language questions, where a large text collection (termed the QA corpus) is used as the source for these answers. Contrary to traditional IR tasks, it is not acceptable for a QA system to retrieve a full

document, or a paragraph, in response to a question. Contrary to traditional IE tasks, no pre-specified domain restrictions are placed on the questions, which may be of any type and in any topic. Modern QA systems must therefore combine the strengths of traditional IR and NLP/IE to provide an apposite way to answering questions.

The QA task in the TREC conference series (Voorhees 2002) has motivated much of the recent works focusing on fact-based, short-answer questions. Examples of such questions include: “*Who is Tom Cruise married to?*” or “*How many chromosomes does a human zygote have?*”. For the most recent TREC-11 conference, the task consists of 500 questions posed over a QA corpus containing more than one million newspaper articles. Instead of previous years’ 50-byte or 250-byte text fragments, exact answers are expected from the QA corpus with supports of documentary evidences.

One of the major challenges in TREC-style QA is to overcome the mismatch in the lexical representations between the query space and document space. This mismatch, also known as the QA gap, is caused by the differences in the set of terms used in the question formulation and answer strings in the corpus. Given a source, such as the QA corpus, that contains only a relatively small number of answers to a query, we are faced with the difficulty to map the questions to answers by way of uncovering the complex lexical, syntactic, or semantic relationships between the question and the answer strings.

Recent redundancy-based approaches (Brill et al 2002, Clarke et al 2002, Kwok et al 2001, Radev et al 2001) proposed the use of data, in-

stead of methods, to do most of the work to bridge the QA gap. These methods suggest that the greater the answer redundancy in the source data collection, the more likely that we can find an answer that occurs in a simple relation to the question. With the availability of rich linguistic resources, we can also minimize the need to perform complex linguistic processing. However, this does not mean that NLP is now out of the picture. For some question/answer pairs, deep reasoning is still needed to relate the two. Many QA research groups have used a variety of linguistic resources – part-of-speech tagging, syntactic parsing, semantic relations, named entity extraction, WordNet, on-line dictionaries, query logs and ontologies, etc (Harabagiu et al 2002, Hovy et al 2002).

This paper investigates the integration of both linguistic knowledge and external resources for TREC-style question answering. In particular, we describe a high performance question answering system called **QUALIFIER (QUESTION ANSWERING BY LEXICAL FABRIC AND EXTERNAL RESOURCES)** and analyze its effectiveness using the TREC-11 benchmark. Our results show that combining lexical information and external resources with a custom text search produces an effective question-answering system.

The rest of the paper is organized as follows. Section 2 presents related work. Sections 3 and 4 respectively discuss the design and architecture of the system. Section 5 elaborates on the use of external resources for QA, while Section 6 details the experimental results. Section 7 concludes the paper with discussions for future work.

2 Related Work

The idea of using the external resources for question answering is an emerging topic of interest among the computational linguistic communities. The TREC-10 QA track demonstrated that the use of the Web redundancy could be exploited at different levels in the process of finding answers to natural language questions. Several studies (Brill et al 2002, Clarke et al 2002, Kwok et al 2001) suggested that the application of Web search can improve the precision of a QA system by 25-30%. A common feature of these approaches is to use the Web to introduce data re-

dundancy for a more reliable answer extraction from local text collections. Radev et al [20] proposed a probabilistic algorithm that learns the best query paraphrase of a question searching the Web.

Many groups (Buchholz 2002, Chen et al 2002, Harabagiu et al 2002, Hovy et al 2002.) working on question answering also employ a variety of linguistic resources, such as the part-of-speech tagging, syntactic parsing, semantic relations, named entity extraction, dictionaries, WordNet, etc. Moldovan and Rus (2001) proposed the use of logic form transformation of WordNet for QA. Lin (2002) gave a detailed comparison of the Web-based and linguistic-based approaches to QA, and concluded that combining both approaches could lead to better performance on answering definition questions.

3 Design Consideration

To effectively perform open domain QA, two fundamental problems must be solved. The first is to **bridge the gap between the query and document spaces**. Most recent QA systems adopt the following general pipelined approach to: (a) classify the question according to the type of its answer; (b) employ IR technology, with the question as a query, to retrieve a small portion of the document collection; and (c) analyze the returned documents to detect entities of the appropriate type. In step (b), the traditional IR systems assume that there is close lexical similarity between the queries and the corresponding documents. In practice, however, there is often very little overlap between the terms used in a question and those appearing in its answer. For example, the best response to the question “*Where’s a good place to get dinner?*” might be “*McDonald’s*” and “*Jade Crystal Kitchen has nice Shanghai Tang Bao*”, which have no tokens in common with the query. Usually, the QA gap reveals itself at four different levels, namely, the *lexical, syntactic, semantic and discourse* levels. As a result, the traditional bag-of-words retrieval techniques might be less effective at matching questions to exact answers than matching keywords to documents.

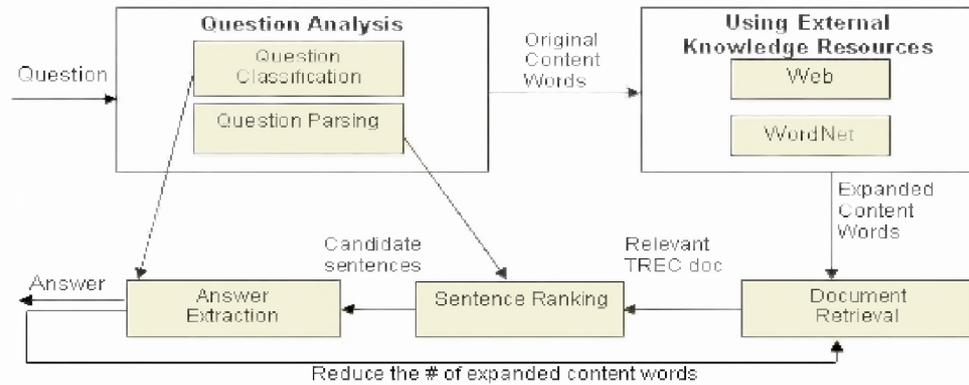


Figure 1: System Overview of QUALIFIER

The second fundamental problem is to **exploit the associations among QA event elements**. The world consists of two basic types of things: *entities* and *events*. From their definitions in WordNet, an entity is *anything having existence (living or nonliving)* and an event is *something that happens at a given place and time*. This taxonomy is also applicable to QA task, i.e., the questions can be considered as enquiries about either entities or events. Usually, the entity questions expect the entity properties or the entities themselves as the answers, such as the definition questions. More generally, questions often show great interests in several aspects of events, namely *Location, Time, Subject, Object, Quantity and Description*. Table 1 shows the correspondences of the most common WH-question classes and the QA event elements.

WH-Question	QA Event Elements
Who	Subject, Object
Where	Location
When	Time
What	Subject, Object, Description
Which	Subject, Object,
How	Quantity, Description

Table 1: Correspondence of WH-Questions & Event Elements

Our major observation is that a QA event shows great cohesive affinity to all its elements and the elements are likely to be closely coupled by this event. Although some elements may appear in different places of the text collection or may even be absent, there must be innate associations among these elements if they belong to the same event. Hence, even if we only know a portion of the elements (e.g. Time, Subject, Object), we can use this information to narrow the search process to find the rest of elements (e.g. Loca-

tion, etc). However, it is difficult to find correct unknown element(s) because of insufficient and inexact known elements.

To tackle these two problems effectively, we explore the use of external resources to extract terms that are highly correlated with the query, and use these terms to expand the query. Instead of treating the web and linguistic resources separately, we explore an innovative approach to fuse the lexical and semantic knowledge to support effective QA. Our focus is to link the questions and the answers together by discovering a portion or all of the elements for certain QA events. We explore the use of world knowledge (the Web and WordNet glosses) to find more known elements and exploit the lexical knowledge (WordNet synsets and morphemics) to find their exact forms. We would like to call our approach *Event-based QA*.

4 System Architecture

Our system, named QUALIFIER, adopts the by now more or less standard QA system architecture as shown in Figure 1. It includes modules to perform question analysis, query formulation by using external resources, document retrieval, candidate sentence selection and exact answer extraction.

During question analysis, QUALIFIER identifies detailed question classes, answer types, and pertinent content query terms or phrases to facilitate the seeking of exact answers. It uses a rule-based question classifier to perform the syntactic-semantic analysis of the questions and determines the question types in a two-level question taxonomy. The first level in the question taxonomy corresponds to the more general named entities

like *Human, Location, Time, Number, Object, Description* and *Others*. The second level contains question classes that correspond to fine-grained named entities to facilitate accurate answer extraction. Examples of second level classes for, say *Location*, are *Country, City, State, River, Mountain* etc. The taxonomy is similar to that used in Li & Roth 2002. Our rule-based approach could achieve an accuracy of over 98% on TREC-11 questions.

At the stage in query formulation, QUALIFIER uses the knowledge of both the Web and WordNet to expand the original query. This is done by first using the original query to search the web for top N_w documents and extracting additional web terms that co-occur frequently in the local context of the query terms. It then uses WordNet to find other terms in the retrieved web documents that are lexically related to the expanded query terms.

Given the expanded query, QUALIFIER employs the MG system (Witten et al 1999) to search for top N ranking documents in the QA corpus. Next, it selects candidate answer sentences from the top returned documents. These sentences are ranked based on certain criteria to maximize the answer recall and precision (Yang & Chua 2003). NLP analysis is performed on these candidate sentences to extract part-of-speech tags, base Noun Phrases, Named Entities, etc.

Finally, QUALIFIER performs answer selection by matching the expected answer type to the NLP results. Named entity in the candidate sentence is returned as the final answer if it fits the expected answer type and is within a short distance to the original query.

The following section describes the details of the query formulation and answer selection using external recourses.

5 The Use of External Knowledge

For the short, factual questions in TREC, the queries are either too brief or do not fully cover the terms used in the corpus. Given a query, $\mathbf{q}^{(0)} = [q_1^{(0)} q_2^{(0)} \dots q_k^{(0)}]$ usually with $k \leq 4$, the problem for retrieving all the documents relevant to $\mathbf{q}^{(0)}$ is that *the query does not contain most of the terms used in the document space to represent the same concept*. For example, given the ques-

tion: “*What is the name of the volcano that destroyed the ancient city of Pompeii?*”, two of the passages containing possible answer in the QA corpus are:

- a. 79 - Mount Vesuvius erupts and buries Italian cities of Pompeii and Herculaneum.
- b. In A.D. 79, long-dormant Mount Vesuvius erupted, burying the Roman cities of Pompeii and Herculaneum in volcanic ash.

As can be seen, there are very few common content words between the question and the passages. Thus we resort to using general open resources to overcome this problem. The external general resources that can be readily used include the Web, WordNet, Knowledge bases, and query logs. In our system, we focus on the amalgamation of the Web and WordNet.

5.1 Using the Web

The Web is the most rapidly growing and complete knowledge resource in the world now. The terms in the relevant documents retrieved from the Web are likely to be similar or even the same as those in the QA corpus since they both contain information about the facts of nature or the factual events in the history. Data redundancy of the web documents plays an important role to effectively retrieve the information for a certain entity or an element of an event.

Aiming to solve the question-answer chasm at the *semantic* and *discourse* levels, QUALIFIER uses the Web as an additional resource to get more knowledge of the entities and events. It uses on the original content words in $\mathbf{q}^{(0)}$ to retrieve the top N_w documents in the Web using Google and then extracts the terms in those documents that are highly correlated with the original query terms. That is, for $\forall q_i^{(0)} \in \mathbf{q}^{(0)}$, it extracts the list of nearby non-trivial words, \underline{w}_i , that are in the same sentence as $q_i^{(0)}$ or within p words away from $q_i^{(0)}$. The system further ranks all terms $w_{ik} \in \underline{w}_i$ by computing their probabilities of correlation with $q_i^{(0)}$ as:

$$Prob(w_{ik}) = \frac{ds(w_{ik} \wedge q_i^{(0)})}{ds(w_{ik} \vee q_i^{(0)})} \quad (1)$$

where $ds(w_{ik} \wedge q_i^{(0)})$ gives the number of instances that w_{ik} and $q_i^{(0)}$ appear together; and $ds(w_{ik} \vee q_i^{(0)})$ gives the number of instances that either w_{ik} or $q_i^{(0)}$ appears. Finally, QUALIFIER merges all \underline{w}_i to form \underline{C}_q for $\underline{q}^{(0)}$.

For the above *Pompeii* example, the top 10 terms extracted from the Web are: “*vesuvius 79 ad roman eruption herculaneum buried active Italian*”.

5.2 Using WordNet

The Web is useful at bridging the semantic and discourse gaps by providing the words that occur frequently with the original query terms in the local context. It however, lacks information on lexical relationships among these terms. In contrast to the Web, WordNet focuses on the lexical knowledge fabric by unearthing the “synonymous” terms. Thus to overcome the QA gap at the lexical and syntactic levels, QUALIFIER looks up WordNet to find words that are lexically related to the original content words. For the aforementioned *Pompeii* example, we find the followings by searching the glosses and synsets.

- a. Ancient
 - Gloss: “belonging to times long past especially of the historical period before the fall of the Western Roman Empire”
 - Synset: {age-old, antique}
- b. *Volcano*
 - Gloss: “a fissure in the earth’s crust (or in the surface of some other planet) through which molten lava and gases erupt”
 - Synset: {vent, crater}
- c. Destroy
 - Gloss: “destroy completely; damage irreparably”
 - Synset: {ruin}

Obviously, the glosses and synsets of the terms in $\underline{q}^{(0)}$ contain useful terms that relate to potential answer candidates in the QA corpus. Here we use WordNet to extract the gloss words \underline{G}_q and synset words \underline{S}_q for $\underline{q}^{(0)}$.

5.3 Integration of External Resources

To link questions and answers at all the four levels of gaps, i.e., the lexical, syntactic, semantic and discourse levels, we need to combine the ex-

ternal knowledge sources. One approach is to expand the query by adding the top k words in \underline{C}_q , and those in \underline{G}_q and \underline{S}_q . However, if we simply append all the terms, the resulting expanded query will likely to be too broad and contain too many terms out of context. Our experiments indicate that in many cases, adding additional terms from WordNet, i.e. those from \underline{G}_q and \underline{S}_q , adds more noise than information to the query. In general, we need to restrict the glosses and synonyms to only those terms found in the web documents, to ensure that they are in the right context. We solve this problem by using \underline{G}_q and \underline{S}_q to increase terms found in \underline{C}_q as follow:

Given $w_k \in \underline{C}_q$:

- if $w_k \in \underline{G}_q$, increase w_k by α ;
 - if $w_k \in \underline{S}_q$, increase w_k by β ;
- where $0 < \beta < \alpha < 1$.

The final weight for each term is normalized and the top m terms above a certain cut-off threshold σ are selected for expanding the original query as:

$$\underline{q}^{(1)} = \underline{q}^{(0)} + \{\text{top } m \text{ terms} \in \underline{C}_q \text{ with weights} \geq \sigma\} \quad (2)$$

where $m=20$ initially in our experiments.

For the *Pompeii* example, the final expanded query $\underline{q}^{(1)}$ is: “*volcano destroyed ancient city Pompeii vesuvius eruption 79 ad roman herculaneum*”. The expanded query contains many overlapping terms or concepts with the passages containing the answers.

QA Event Element	Query Term
Subject	<i>Volcano, vesuvius</i>
Object	<i>Pompeii</i>
Location	<i>roman</i>
Time	<i>79 ad</i>
Description	<i>Destroyed, eruption, herculaneum</i>

Table 2: Term Classification for Pompeii Example

If we classify the terms in the newly formulated query (see Table 2), they are actually corresponding to one or more of the QA event elements we discussed in Section 3. One promising advantage of our approach is that we are able to answer any factual questions about the elements in this QA event other than just “*What is the name of the volcano that destroyed the ancient city of Pompeii?*”. For instance, we can easily handle questions like “*When was the ancient city of Pompeii destroyed?*” and “*Which two*

Roman cities were destroyed by Mount Vesuvius?" etc. with the same set of knowledge. Currently, we are exploring the use of Semantic Perceptron Net (Liu & Chua 2001) to derive semantic word groups in order to form a more structured utilization of external knowledge.

5.4 Document Retrieval & Answer Selection

Given $q^{(1)}$, QUALIFIER makes use of the MG tool to retrieve up to N ($N=50$) relevant documents from the QA corpus. We choose Boolean retrieval because of the short length of the queries, and to avoid returning too many irrelevant documents when using the similarity based retrieval. If $q^{(1)}$ does not return sufficient number of relevant documents, the extra terms added is reduced and the Boolean search is repeated. Therefore, we successively relax the constraint to ensure precision.

QUALIFIER next performs sentence boundary detection on the retrieved documents. It selects the top k sentences by evaluating the similarity between each of the sentences with the query in terms of basic query terms, noun phrases, answer target, etc.

Finally, it performs the tagging of fine-grained named entity for the top K sentences. From these sentences, it extracts the string that matches the question classes (answer target) as the answer. Once an answer is found in the top i^{th} sentence, the system will stop the search for the rest of ($K-i$) sentences. Sometimes, there may be more than one matching strings in a single sentence. We will choose the string, which is nearest to the original query terms.

For some questions, the system cannot find any answer and so we reduce the number of extra terms ($m \leq 20$ in Equation 2) added to $q^{(0)}$ by p ($p=1$). This is to ensure that the Boolean retrieval process can retrieve more documents from the QA corpus. It repeats the document/sentence retrieval and answer extraction process for up to L such iterations ($L=5$). If it still cannot find an exact answer at the end of 5 iterations, a *NIL* answer is returned. We call this method *successive constraint relaxation*. This strategy helps to increase recall while preserving precision.

As an alternative to the *successive constraint relaxation* using Boolean retrieval, similarity-

based search may be used to improve recall possibly at the expense of precision. We will investigate some of these issues in the next Section.

6 Experiments

We use all the 500 questions of TREC-11 QA track as our test set. The performance of QUALIFIER without the use of WordNet and web is considered as the baseline.

6.1 Effects of Web Search Strategies

We first study the effects of employing different strategies to search the web on the QA performance. For Web search, we adopt Google as the search engine and examine only snippets returned by Google instead of looking at full web pages. We study the performance of QUALIFIER by varying the number of top ranked web pages returned N_w , and the cut-off threshold σ (see Equation 2) for selecting the terms in \underline{C}_q to be added to $q^{(0)}$. The variations are:

- The number of top ranked web pages returned (N_w): 10, 25, 50, 75 and 100.
- The cut-off thresholds (σ): 0.1, 0.2, 0.3, 0.4, and 0.5.

Table 3 summarizes the effects of these variations on the performance of TREC-11 questions. Due to space constraint, Table 3 only shows the precision score, P , which is the ratio of correct answers returned by QUALIFIER. From the results, we can see that the best result is obtained when we consider the top 75 ranked web pages, and a term weight cut-off threshold of 0.2. The finding is consistent with the results reported in (Lin 2002) for the definition type questions.

$\sigma \setminus N_w$	10	25	50	75	100
0.1	0.492	0.492	0.494	0.500	0.504
0.2	0.536	0.536	0.538	0.548	0.544
0.3	0.506	0.506	0.512	0.512	0.512
0.4	0.426	0.426	0.430	0.432	0.428
0.5	0.398	0.398	0.412	0.418	0.412

Table 3: The Precision Score of 25 Web Runs

6.2 Using External Resources

To investigate the performance of combining lexical knowledge such as WordNet and external resource like the Web, we conduct several ex-

periments to test different uses of these resources:

- **Baseline:** We perform QA without using the external resources.
- **WordNet:** Here we perform QA by using different types of lexical knowledge obtained from WordNet. We use either the glosses \underline{G}_q or synset \underline{S}_q or both. In these tests, we simply add all related terms found in \underline{G}_q or \underline{S}_q into $\underline{q}^{(1)}$.
- **Web:** Here we add up to top m context words from \underline{C}_q into $\underline{q}^{(1)}$ based on Equation (2).
- **Web + WordNet:** Here we combine both Web and WordNet knowledge, but do not constrain the new terms from WordNet. This is to test the effects of adding some WordNet terms out of context.
- **Web + WordNet with constraint** as defined in Section 5.3.

In these test, we examine the top 75 web snippets returned by Google with a cut-off threshold σ of 0.2. Also, we use the answer patterns and the evaluation script provided by NIST to score all runs automatically. For each run, we compute P , the precision, and CWS , the confidence-weighted score. Table 4 summarizes the results of the tests.

Method	P	CWS
Baseline	0.438	0.440
Baseline + WordNet Gloss	0.442	0.448
Baseline + WordNet Synset	0.438	0.446
Baseline + WordNet (Gloss,Synset)	0.442	0.446
Baseline + Web	0.548	0.578
Baseline + Web + WordNet	0.552	0.588
Baseline + Web + WordNet + constraint	0.588	0.610

Table 4: Different Query Formulation Methods

From Table 4, we can draw the following observations.

- The use of lexical knowledge from WordNet without constraint does not seem to be effective for QA, as compared to baseline. This is because it tends to add too many terms out of context into $\underline{q}^{(1)}$.
- Web-based query formulation improves the baseline performance by 25.1% in Precision and 31.5% in CWS. This confirms the results of many studies that using Web to extract highly correlated terms generally improves the QA performance.
- The use of WordNet resource without constraint in conjunction with Web again does not help QA performance.

- The best performance (P : 0.588, CWS : 0.610) is achieved when combining the Web and WordNet with constraint as outlined in Section 5.3.

6.3 Boolean Search vs. Similarity Search

In all the above experiments, we employ *successive constraint relaxation* technique to perform up to 5 iterations of Boolean search on the QA corpus as outlined in Section 5.4. The intuition here is that similarity-based search tends to return too many irrelevant QA documents, thus degrades the overall precision of QA. Our observation of the Boolean-based approach is that we tend to return too many NIL answers prematurely. In order to test our intuition and to maximize the chances of finding exact answers, we conduct a series of tests by employing a combination of Boolean search and/or similarity-based search.

The results are presented in Table 5. As can be seen, the best result is obtained when performing up to 5 successive relaxation iterations of Boolean search followed by a similarity-based search. This is the most thorough search process we have conducted with the aim of finding an exact answer if possible and only returning a NIL answer as the last resort. It works well as our answer selection process is quite strict.

Search Method	P	CWS
Boolean	0.386	0.426
Boolean+5iterations	0.580	0.610
Similarity	0.266	0.240
Boolean+Similarity	0.450	0.466
Boolean+5iterations+Similarity	0.602	0.632

Table 5: Results of Boolean vs Similarity Search

7 Conclusion and Future Directions

We have presented the QUALIFIER question answering system. QUALIFIER employs a novel approach to QA based on the intuition that there exists implicit knowledge that connects an answer to a question, and that this knowledge can be used to discover the information about a QA entity or different aspects of a QA event. Lexical fabric like WordNet and external recourse like the Web are integrated to find the linkage between questions and answers.

Our results obtained on the TREC-11 QA corpus correlate well with the human assessment of

answers' correctness and demonstrate that our approach is feasible and effective for open domain question answering.

We are currently refining our approach in several directions. First, we are improving our query formulation by considering a combination of local context, global context and lexical term correlations. Second, we are working towards template-based approach on answer selection that incorporates some of the current ideas on question profiling and answer proofing, etc. Third, we will explore the structured use of external resources using the *semantic perceptron net* approach (Liu & Chua 2001). Our long-term research plan includes Interactive QA, and the handling of more difficult analysis and opinion type questions.

References

- AAAI Spring Symposium Series. 2002. *Mining Answers from Text and Knowledge Bases*.
- ACL-EACL. 2002. Workshop on Open-domain Question Answering.
- E. Brill, J. Lin M. Banko, S. Dumais, and A. Ng. 2002. *Data-intensive question answering*. Text Retrieval Conference (TREC 2001)
- E. Brill, S. Dumais and M. Banko. 2002. *An analysis of the AskMSR question-answering system*. In Proceedings of 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002).
- S. Buchholz. 2002. *Using grammatical relations, answer frequencies and the World Wide Web for TREC question Answering*. In Proceedings of the Tenth Text Retrieval Conference (TREC 2001).
- J. Chen, A. R. Diekema, M. D. Taffet, N. McCracken, N. E. Ozgencil, O. Yilmazel, E. D. Liddy. 2002. *Question answering: CNLP at the TREC-10 question answering track*. In Proceedings of the Tenth Text Retrieval Conference (TREC 2001).
- C. Clarke, G. Cormack and T. Lynam. 2002. *Web reinforced question answering*. In Proceedings of the Tenth Text Retrieval Conference (TREC 2001).
- C. Clarke, G. Cormack and T. Lynam. 2001. *Exploiting redundancy in question answering*. In Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'2001), 358-365.
- S. Harabagiu, D. Moldovan, M. Pasca, R. Mihalcea, M. Surdeanu, R. Bunescu, R. Girju, V. Rus and P. Morarescu. 2001. *FALCON: Boosting knowledge for question answering*. In Proceedings of the Ninth Text Retrieval Conference (TREC-9), 479-488.
- E. Hovy, U. Hermjakob and C. Lin. 2002. *The use of external knowledge in factoid QA*. In Proceedings of the Tenth Text REtrieval Conference (TREC 2001).
- C. Kwok, O. Etzioni and D. Weld. 2001. *Scaling question answering to the Web*. In Proceedings of the 10th World Wide Web Conference (WWW'10), 150-161.
- X. Li and D. Roth. 2002. *Learning Question Classifiers*. In Proceedings of the 19th International Conference on Computational Linguistics, 2002
- C.Y. Lin. 2002. *The Effectiveness of Dictionary and Web-Based Answer Reranking*. In Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002).
- J. Liu and T. S. Chua. 2001 *Building semantic perceptron net for topic spotting*. In Proceedings of 37th Meeting of Association of Computational Linguistics (ACL 2001),370-377.
- D. I. Moldovan and Vasile Rus. 2001.*Logic Form Transformation of WordNet and its Applicability to Question Answering*. In Proceedings of the ACL 2001 Conference, July 2001.
- M. A. Pasca and S. M. Harabagiu. 2001. *High performance question/answering*. In Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'2001), 366-374.
- J. Prager, E. Brown, A. Coden and D. Radev. 2000. *Question answering by predictive annotation*. In Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'2000), 184-191.
- D. R. Radev, Weiguo Fan, Hong Qi, Harris Wu, and Amardeep Grewal. 2002. *Probabilistic question answering from the web*. In The Eleventh International World Wide Web Conference,2002.
- E.M.Voorhees. 2002. *Overview of the TREC 2001 Question Answering Track*. In Proceedings of the Tenth Text Retrieval Conference (TREC 2001)
- I. Witten, A. Moffat, and T. Bell. 1999. *Managing Gigabytes*. Morgan Kaufmann.
- H. Yang and T. S. Chua. 2003. *The Integration of Lexical Knowledge and External Resources for Question Answering*. In Proceedings of the Tenth Text Retrieval Conference (TREC 2002)