A Semantic Approach to Boost Passage Retrieval Effectiveness for Question Answering

Bahadorreza Ofoghi
John Yearwood
Ranadhir Ghosh

Centre for Informatics and Applied Optimization
School of Information Technology and Mathematical Sciences
University of Ballarat
PO Box 663, Ballarat, Victoria 3353, Australia

bofoghi@students.ballarat.edu.au
{j.yearwood, r.ghosh}@ballarat.edu.au

Abstract

In the current state of the rapid growth of information resources and the huge number of requests submitted by users to existing information retrieval systems; recently, Question Answering systems have attracted more attention to meet information needs providing users with more precise and focused retrieval units. As one of the most challenging and important processes of such systems is to retrieve the best related text excerpts with regard to the questions, we propose a novel approach to exploit not only the syntax of the natural language of the questions and texts, but also the semantics relayed beneath them via a semantic question rewriting and passage retrieval task. The semantic structure used to address the surface mismatch of the semantically related passages and queries is FrameNet which is a lexical resource for English constituted based on frame semantics. We have run our proposed approach on a subset of the TREC 2004 factoid questions to retrieve passages containing correct answers from the AQUAINT collection and we have obtained promising results.

Keywords: Passage Retrieval, FrameNet, Question Answering, Semantic Boosting.

1 Introduction

In recent years, Question Answering (QA) systems have evolved out of the field of Information Retrieval (IR) to better understand and more precisely cope with information requests. Unlike simple and popular keyword-based information retrieval systems (e.g. Web search engines), QA systems aim to communicate directly with users through a natural language which brings more convenience and comprehension to users who submit their information needs. Having received natural language questions, such systems perform various processes to return actual direct answers to the requests eliminating the burden of query formulation and reading lots of irrelevant documents to reach the desired answer by users. This is due to the fact that a user usually wants not whole documents but brief answers to the specific questions like: *"How old is the President? Who was the second person on the moon? When was the storming of the Bastille?"* (Hovy, Gerber et al. 2001).

In a typical architecture of a question answering system, there are four main procedures; i) question analysis and query formulation, ii) document retrieval, iii) passage retrieval, and iv) answer extraction. The task of analysis of a question contains different sub-procedures based on the general view of the question answering system. In an ontology-based system, this consists of finding related ontology nodes for the submitted question in order to carry out further related processes (Hejazi, Mirian et al. 2003), while in most other systems the procedure of question analysis tries to find named entities and/or to recognize the answer category of the question (Moschitti and Harabagiu 2004), to take into account the temporal issues of the question (Saquete, Martinez-Barco et al. 2004), and to formulate the best representative keyword-based query to boost the retrieval precision in the tasks of document and passage retrieval (Brill, Dumais et al. 2002). Obviously, none of these goals could be achieved before precise and sophisticated natural language processing on the question. In the next step the question answering system is supposed to find the best textual documents from inside the collection which is the answer resource of the system. Such documents should contain passages relevant to the topic of the question. The task of document retrieval, which could be automated using the best known search engines, is bypassed in some question answering systems as they retrieve best passages directly from inside the whole collection. However, the main idea of retrieving the most relevant text snippets to the question is commonly accepted by all question answering systems, When it comes to answering specific information needs of users, the successful extraction of candidate and actual answers could be achieved only on the part of the text which is most similar to the queries formulated based on the original questions. The idea of how to find candidate and actual answers of a question is mostly dependent on the syntactic or semantic structure that is used by the question answering system. START tries to extract such short amounts of information based on ternary expressions matching (Katz 1997). There is a proposed idea for modelling documents based on recognizing Named Entities (Pérez-Coutiño, Solorio et al. 2004) which leads to finding corresponding named entities already recognized inside the text using the SUMO ontology. One of the sophisticated approaches to extract answers has been developed based on frame semantics and sentence annotation using the English lexicon resource, FrameNet, which performs frame and frame element matching and makes inferences inside the related parts of the conceptual graph of FrameNet (Narayanan and Harabagiu 2004).

While working on a question answering architecture, we realized that the precision of best known passage retrieval algorithms could not go higher than a low pick due to some inconsistencies between the questions and the contents of the documents. Having considered that the passage retrieval task is one of the necessary sub-processes in a question answering system (Clarke and Terra 2003), it is worthy to work more on this step to boost the current state-of-the-art of the existing best-known passage retrieval algorithms. Hence, we propose and explore a novel approach on boosting the effectiveness of the passage retrieval task in the context of question answering in a large collection of text so that the system could cope with different types of syntactical mismatch between formulated queries and the texts. We justify our approach based on the results we obtained for a subset of the TREC 2004 factoid questions and the AQUAINT collection using the MultiText (Clarke, Cormack et al. 1997) passage retrieval algorithm and Lemur's passage retrieval engine. Our idea, which exploits Intra-Frame relations between different English terms inside the frames of FrameNet (Baker, Fillmore et al. 1998), has been developed on the basis of poor coverage of the two above-mentioned passage retrieval techniques on the answers of the questions. It has shown impressive results, even though the idea requires that the question (rewritten question) be submitted to the passage retrieval engine more than once.

This paper is organised as follows. Section 2 describes what we mean by passage retrieval for question answering, and also introduces the two passage retrieval algorithms that we have used. In section 3 the main idea of Intra-Frame analysis in FrameNet in order to rewrite the questions and retrieve semantically related passages, as well as the methodology of judging the passages, are described. Section 4 explains the experimental issues and finally, in section 5 we conclude the paper.

2 Passage Retrieval for Question Answering

There are different reasons for a question answering system to perform either well or poorly on the basis of the precision of the answers it provides to submitted questions. We are convinced that in order to find candidate answers that can be used to decide about the actual answer, such systems should be provided with one or more text snippets each of which may contain one or more sentences. This is a crucial sub-process of an end-to-end question answering system. It is also clear that in case there is no candidate or actual answer present inside retrieved passages, then there is no chance for the system to return a correct answer.

There have been many efforts on different passage retrieval algorithms (Tellex, Katz et al. 2003) for dissimilar purposes with diverse points of view on the definition of the word "passage". As mentioned in (Callan 1994) and (Kaszkiel and Zobel 1997) and also referred to in (Kaszkiel, Zobel et al. 1999), the most effective and reliable definition of passage is what includes a fixed-length sequence of words starting and ending anywhere in the document. However, it is not clear that they have tried all well-known algorithms including MultiText algorithm (Clarke, Cormack et al. 1997) which, in our experiments, outperforms Lemur's passage retrieval engine (using its best retrieval model) that will be discussed further later. All of the Lemur's passage retrieval models take into account fixed-size passages to be indexed and retrieved.

The output of the passage retrieval task is very dependent on the query formulation of the original question, and certainly, the query formulation process could not be established before accurate knowledge about the index structure (e.g. if phrase indexing is supported, and if stemmed terms are indexed) of the texts inside the collection. In the next sections, we explore the two passage retrieval methods that we used as well as the specific settings necessary for each. The selection of these two passage retrieval algorithms is strongly based on the fact they cover both fixed-size and dynamic-size passages which is of important characteristics of such algorithms.

2.1 MultiText Algorithm

One of the best-known passage retrieval algorithms is the MultiText algorithm exploited for document ranking and retrieval purposes as well. This algorithm interprets all documents as a series of continuous words and also interprets passages as any number of words starting and ending anywhere inside the documents of a collection (Clarke, Cormack et al. 1997). These passages, which are initially identified by covers, start with one of the query keywords and end with another one, not overlapping the boundaries of documents which constitute the unique string of the words. Experiments performed in (Tellex, Katz et al. 2003) show that this algorithm has shown quite high performance; the third highest MRR (Main Reciprocal Rank) in documents retrieved by the PRISE search engine and the highest MRR in those retrieved by the Lucene search engine. The results are obtained among the eight passage retrieval algorithms investigated by the authors. This high performance, as well as the frequent participation of MultiText in TREC (Clarke, Cormak et al. 2000), were the main reasons for choosing MultiText as one of our passage retrieval algorithms.

2.2 Lemur's Retrieval Engine

Lemur is a toolkit designed to facilitate research in language modelling and information retrieval[2]. It includes a well-designed and supported implementation of different functionalities for text parsing, indexing, retrieval, summarization, and clustering. We have used the indexing and passage retrieval functions of Lemur. Focusing on passage retrieval, Lemur has seven retrieval models each of which could be applied for both document and passage retrieval tasks; i) the tf/idf model, ii) the Okapi bm25 model, iii) KL-divergence language model based method, iv) the CORI model, v) CORI collection selection model, vi) Cosine similarity model, and vii) Indri structured query language. After comparing the retrieval efficiency of these different models, the CORI collection selection model showed the best performance in retrieving the most related passages for the TREC 2004 factoid questions in the AQUAINT collection. The task of passage retrieval is performed based on fixed-size passages inside the documents, while passages have overlaps equal to half of the size of the passages.

3 Exploiting Intra-Frame Term-Level Relations inside FrameNet

As most passage retrieval algorithms are dependent on the occurrences of exact matches of surface features inside the queries and textual documents, even their state-of-the-art precision of retrieval could not go beyond the limitations which are formed by mentioned syntactic structures. In other words, there is little chance for any such passage retrieval algorithm to return a passage which contains the word "spot" in response to a query containing the keyword "discover", for instance. This is because of the fact that there could not exist any type of syntactic similarity between the two words, though they share similar meaning. The problem could be still more complex to solve, in a state of common concepts rather than meanings. For example, in a scenario of a passage where the word "son" is mentioned, there is no syntactic clue to relate any query containing the word "father" to the passage. Such types of mismatch between query keywords and those which may occur inside the texts lead us to resolve the issue by moving towards the semantics underlying the text. Initially, we have found a solution to this sort of query and passage mismatch by using FrameNet data in a Question Rewriting and re-retrieval of passages inside the collection.

3.1 FrameNet Lexicon Resource

FrameNet is a lexicon resource for English (Baker, Fillmore et al. 1998) whose infrastructure is based on Frame Semantics (Lowe, Baker et al. 1997) which is different with Marvin Minsky's frames. FrameNet contains two main entities to completely model and conceptualize the scenarios and the target words which could be realized in the scenarios. Frames, in the highest level of abstraction within FrameNet, encode the base definitions necessary to understand the semantics and the scene of each contained word. In other words, real-world knowledge about the scenarios and their related properties are encoded inside the Frames (Lowe, Baker et al. 1997). To address this, each Frame contains some Frame Elements as representatives of the different semantic and syntactic roles (valences) regarding a target word inside the Frame. The semantic roles are usually common among all of the words that are inherited from a Frame. This ensures a suitable generalization over the English words which either have similar meanings or share the context and/or the scenario in which they could occur in the sentences of the language.

FrameNet is different from WordNet as it contains not only words with similar meanings, but also higher level concepts of similar scenarios of usage in the real-world. On the other hand, these scenarios are related to each other to model an end-to-end scenario containing some smaller sub-scenarios. The different relation types existing between Frames cover this overview of the different events all of which could be realized by FrameNet.

In addition, FrameNet has more than what are formulated by the Predicate-Argument Structure (Surdeanu, Harabagiu et al. 2003), considering the fact that predicates in the Predicate-Argument structure normally are the verbs of the language and the arguments are formed based on dissimilar roles that the predicate could play in the sentences of the language. Target words of FrameNet are nouns, verbs, and even adjectives of the language.

Given the above considerations, FrameNet is well suited for our proposed idea on the resolution of the passage-query syntactical mismatches.

3.2 Passage-Query Mismatch Resolution

The generalization over conceptual scenarios and their related properties is the main characteristic of FrameNet that we have been interested in for resolving the problem of poor passage retrieval performance in the context of question answering due to the syntactic mismatch between the words inside the collection and the keywords of the queries formulated based on original questions. The semantic generalization applied by FrameNet is playing the role of the lost chain for retrieving semantically related passages in response to the queries.

It should be noticed that, in the context of question answering, not all types of semantic query expansion is of interest regarding the fact that a question answering system has to be capable of answering exact questions with actual direct answers. For instance, it is not realistic to change the original query, formed based on the original question, using WordNet semantic relations which has performed well for document retrieval tasks (Voorhees 1994). It causes the retrieval of more indirectly related passages to the question leading to extracting answers which may not be of interest. This argument does not include the systems which try to identify online relations between concepts of different abstraction levels (e.g. (Moldovan, Harabagiu et al. 2002)) that may result in a beneficial semantic matching of the text of the questions

[2] http://www.lemurproject.org/lemur/overview.html

and passages. On the other hand, applying ontology relations between entities or using fuzzy inclusion relations (Akrivas, Wallace et al. 2002) could result in irrelevant passages to come up in the final ranking of retrieved passages. We argue that these methods are not suitable for answering direct factoid questions; however, they have performed well in different contexts.

In what is called generalization over conceptual scenarios and their related properties, the actual procedure of our proposed idea contains a joint generalization-specialization action which evokes a Frame and then considers one of the related terms that is inherited from the Frame. This generalization-specialization method guarantees the query remains at the same semantic abstraction level of the original question.

While these sorts of passages either could not be retrieved or have a very low similarity measure with the query, the way to boost the performance of the retrieval is to substitute the target word of the question with semantically related ones. This is what we call Intra-Frame Term-Level relation, as the substitution is performed based on the target Frame inside FrameNet and the lexical units (terms) covered by the Frame. Figure 1 depicts what happens in a cycle of boosting the passage retrieval effectiveness via question rewriting in the context of question answering.

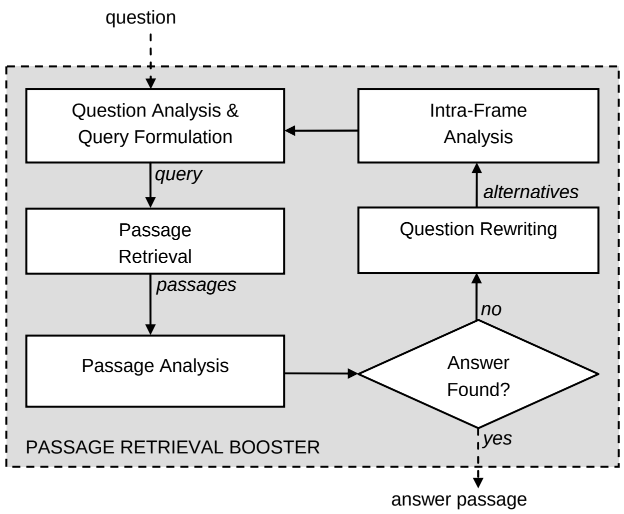

Figure 1: The main cycle of boosting passage retrieval effectiveness in the context of question answering

It should be noticed that the passage retrieval algorithm that is mainly used in this architecture is a modified version of MultiText passage retrieval algorithm whose modifications will be discussed further in the next sections.

The cycle of passage retrieval starts with submitting a question to the system already developed for this purpose. Initially, the question is subject to natural language processing in order that the main keywords to formulate the representative query are known and some other information related to other tasks of question answering is extracted. Then, the query will be sent to the passage retrieval engine to find the best match text excerpts. If the top-ranked passages, based on the manual analysis performed by the passage analysis module, contain the real answer, then no further process is performed at this stage; otherwise, the system tries to identify semantically related text snippets, which are missed due to a syntactic mismatch, after the Intra-Frame analysis on the Frame from which the current target word inherits. The alternative word is one which is also inherited from the evoked Frame by the initial target word and in addition, it has the same part-of-speech (e.g. verb) as that of the initial target. In order to better explain the idea, we consider Example 1.

Example 1: A question from the question list of TREC 2004 is considered (the question id is 3.1 and the target id is 3). The question is fed to the system and the retrieval cycle is as follows;

> Question “When was the comet discovered?” (TREC Target: Hale Bopp comet) → Query “comet discover Hale Bopp” → No Answer in Retrieved Passages → Corresponding Frame Call Evokes the Frame “BECOMING_AWARE” → Intra-Frame Analysis and Alternative Predicate Finding “Spot” → Question Rewriting Using Alternative Predicate “ When was the comet spotted” (TREC Target: Hale Bopp comet) → Query “comet spot Hale Bopp” → Answer Found in the Second Passage.

Inside the AQUAINT collection for TREC 2004, there are some passages containing similar passages to the original question 3.1; however, none of them contains the answer. The top-most passage which is returned by the modified MultiText at the first cycle is:

> <PASSAGE no=1 score= 1.0>
> *Hale-Bopp, a newly-discovered extraordinarily large comet in the solar system, has been recently observed for the first time in China.*
> </PASSAGE>

which is very similar to the query formulated as mentioned above. However, because of the fact that the real answer has not been mentioned using the same predicate “discover”, the passage retrieval algorithm could not either bring the container of the real answer to the top ranks or even retrieve it, as it is the case in this example.

After finding the alternative semantically related predicate “Spot” from inside the corresponding Frame “BECOMING_AWARE”, the rewrite question and the respective query will come up with a passage like below at the second rank;

> <PASSAGE no=2 score= 0.96209>
> *The comet, one of the brightest comets this century, was first spotted by Hale and Bopp, both astronomers in the United States, on July 23, 1995.*
> </PASSAGE>

which contains the correct answer to the question, although it still needs some context resolution and actual answer extraction processes to be performed.

This example clearly shows what happens in the passage retrieval process for question answering systems which could not extract correct answers for those questions which have not a syntactically direct match inside the collection. In contrast, the proposed idea for re-submitting rewrite questions based on Intra-Frame Term-

Level analysis shows promising resolution over the problem.

3.3 Evaluating Passages

As discussed in (Kaszkiel, Zobel et al. 1999) there are usually two ways to measure the retrieval performance of a text retrieval system (e.g. a passage retrieval system). The first way is to measure the *efficiency* which is based on the usage of the resources like disk, time, and memory. In the second manner, the *effectiveness* of the system is measured with regard to the value of satisfaction of users by retrieved texts.

In the context of the question answering systems, the effectiveness of the passages are more important especially to the extent that they potentially deliver correct actual answers to the question submitted by a user.

In focussing on a QA task and using the TREC QA track, our judgment of the passages is based on whether the retrieved passages satisfy the reported correct answer patterns by TREC for each question. In standard passage retrieval, passages are judged for relevance or `aboutness' but in this instance we are assessing passages on whether or not they contain the correct answers. This is a more stringent requirement than relevance. Consequently many highly similar passages, in this context, will not have the actual answer.

The justification of the passages in passage analysis module of the boosting cycle, in further experiments, is to be based on complicated judgements on the candidate answers in the context of a question answering system, although in our first experiments, as mentioned earlier, this has been done manually with regard to the answer patterns reported by TREC. The manual justification of the passages is subject to further study and work with respect to the features of the answer extraction process in an end-to-end question answering system.

In addition, we are concerned about a reasonable method that could extract such answers from inside the potentially correct passages. We do not cover these issues here as they are part of our work in the question answering architecture and the subject of our current and next study.

4 Experimental Issues

We discuss our experimental results with regard to the three aspects of the research study that is being undertaken for semantically answering factoid questions.

4.1 Data

The dataset that has been used for this research study is the TREC 2004 question list and its corresponding text collection of AQUAINT[3]. This collection contains news articles from New York Times News Service (1998-2000), Xinhua News Service (1996-2000), and Associated Press Worldstream News Service (1998-2000).

The question list contains 65 targets and 230 factoid questions (the total number of all type of the questions is 351). We have tried our proposed idea on a subset of this track which contains 20 targets out of 65 and 65 factoid questions out of 230 which is equal to 28.26% of the total number of factoid questions in the TREC 2004 QA track. However, there are 5 questions out of these 65 factoid questions for which no answer could be found in the AQUAINT collection, as a subset of NIL answers reported by TREC (Voorhees 2004). Therefore, we consider a total of 60 questions in our experiments.

4.2 Procedure

In order that a passage retrieval task is performed, in most question answering systems, there is a document retrieval process prior to the passage retrieval task, as mentioned earlier. This should be the case, especially when manipulating a huge-sized collection of text on which a direct passage retrieval task is very complex and time-consuming. Therefore, we used the top-ranked documents reported by TREC for each target[4] to escape the need of the implementation of a document retrieval engine. This ensures that we are convinced of the necessity of a document retrieval stage, although we have not implemented it and benefited from the results from the PRISE information retrieval system via the TREC reports.

We ran two passage retrieval algorithms on the dataset; modified MultiText, which we implemented, and Lemur's passage retrieval engine, where we used the APIs.

In modified MultiText, we create a feature vector for both the passage and the query. Afterwards, we use the Cosine similarity function to measure the similarity value between passages and the query. To find the feature values of the feature vector for the passages we use Equation 1 and to measure the similarity value between the two feature vectors of the query and the passage Equation 2 is applied, which is composed of the well-known Cosine Similarity Function and the effect of the term coverage of the passage.

$$vec_i = \frac{tf_i}{\log(passageLength_j) + tf_i} * weight_i \qquad (1)$$

$$Sim(q, p) = \cos(q, p) * \frac{coverage_j}{queryLength} \qquad (2)$$

In Equation 1, tf_i is the raw term frequency of the query term i inside the passage, $weight_i$ is the weight of this term which is assigned based on two considerations; i) the part-of-speech of the term (i.e. the verbs have higher weights than nouns, adjectives, and so on), and ii) the terms which occur inside the TREC target of the question gain a bonus on their weights to increase up to 1.0. The value $coverage_j$, in Equation 2, contains the unique

[3] http://www.ldc.upenn.edu/Catalog/docs/LDC2002T31/

[4] http://trec.nist.gov/data/qa.html

number of the query terms that a passage covers and *queryLength* is the total number of the terms inside the query.

While running Lemur's passage retrieval algorithm, we used the passage size of 160 words. Authors in (Kaszkiel, Zobel et al. 1999) have mentioned that this could be in the optimum value range for the passage retrieval algorithms which take into account a fixed size for the passages to be retrieved. Also, we tried different retrieval methods of Lemur and decided that the CORI-Collection Selection method outperforms the other supported models in the context of our work.

4.3 Results

We developed a software platform to test the two above-mentioned passage retrieval algorithms and also to perceive the increase on the output results based on the evaluation methodology explained at the section 3.3.

As shown in Table 1, the highest retrieval effectiveness for Lemur's retrieval engine, which has been acquired by the CORI-Collection Selection retrieval model, was 58.2%, while this percentage went up to 70% for the same questions using the modified MultiText algorithm.

Retrieval Method	Questions with Answers in Top 10 Passages	No. of Questions
Lemur's PR	%58.3	60
Modified MultiText	%70	60
Modified MultiText along with Semantic Resolution	%75	60

Table 1: Retrieval effectiveness of the three runs of passage retrieval

The results have been obtained by considering the top 10 passages for each retrieval task. Whenever the answer was recognized inside one of the top 10 passages retrieved for any question the score for that question was considered 1; otherwise 0. In the end, the percentage was calculated as the average value of over all scores.

Because of the higher performance of the MultiText algorithm on the dataset that we are working on, we chose to apply the proposed idea of semantic question rewriting and semantic mismatch resolution on the modified version of the MultiText algorithm. We obtained an effectiveness of 75% on the same subset of factoid questions and their representative queries. A promising increase in effectiveness is gained on a subset of the TREC questions. We expect that this performance may go even higher either on a bigger subset or on the total number of the questions in the track.

5 Conclusion

Due to the poor coverage of the best-known passage retrieval algorithms on the actual answers related to a question answering task of TREC 2004, we have developed an idea to retrieve passages which are not syntactically matched to the keywords of representative queries of the original questions. As long as deep semantic relations are not considered by the passage retrieval process, it can not cope with syntactically mismatched passages which at the same time contain semantically related elements to the question. The proposed idea tries to rewrite the questions which come up in such situations using alternative related terms from inside the evoked Frame of FrameNet by the original target predicate. This rewriting and re-submit cycle is protective of the original semantic abstraction level of the questions and does not cause any unnecessary generalization over the concepts which exist in the questions to avoid retrieving irrelevant passages. We have developed our idea on a subset of the TREC 2004 questions and the AQUAINT collection and have achieved impressive improvement on the state-of-the-art of two best-known passage retrieval algorithms.

6 References

Akrivas, G., M. Wallace, et al. (2002): Context-Sensitive *Semantic* Qquery Expansion. *Proceedings of the 2002 IEEE International Conference on Artificial Intelligence Systems (ICAIS'02).*

Baker, C. F., C. J. *Fillmore*, et al. (1998): The Berkeley FrameNet Project. *International Conference On Computational Linguistics* **1**: 86 - 90.

Brill, E., S. Dumais, et *al*. (2002): An Analysis of the AskMSR Question-Answering System. *Proceedings of 2002 conference on empirical methods in natural language processing (EMNLP 2002).*

Callan, J. P. (*1994*): Passage-Level Evidence in Document Retrieval. *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*: 302-310.

Clarke, C., G. *Cormack*, et al. (1997): Relevance Ranking for One to Three Term Queries. *Proceedings of RIAO-97, 5th International Conference ``Recherche d'Information Assistee par Ordinateur''*: 388-400.

Clarke, C. L. A., G. V. Cormak, et al. (2000): Question Answering By Passage Selection (MultiText Experiments for TREC-9). *Ninth Text REtrieval Conference (TREC 9).*

Clarke, C. L. A. and E. L. Terra (2003): Passage Retrieval vs. Document Retrieval for Factoid Question Answering. *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*: 427 - 428.

Hejazi, M. R., M. S. Mirian, et al. (2003): TeLQAS: A Telecommunication Literature Question Answering System Benefits from a Text Categorization Mechanism. *IKE03*.

Hovy, E., L. Gerber, et al. (2001): Question Answering in Webclopedia. Proceedings *of the TREC-9 Conference*.

Kaszkiel, M. and J. Zobel (1997): Passage Retrieval Revisited. *Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval*: 178-185.

Kaszkiel, M., J. Zobel, et al. (1999): Efficient passage ranking for document databases. *ACM Transactions on Information Systems (TOIS)* **17**(4): 406-439.

Katz, B. (1997): Annotating the World Wide Web using Natural Language. *Proceedings of the 5th RIAO Conference on Computer Assisted Information Searching on the Internet (RIAO '97).*

Lowe, J. B., C. F. Baker, et al. (1997): A Frame-Semantic Approach to Semantic Annotation. *SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What, and How?*

Moldovan, D., S. Harabagiu, et al. (2002): LCC Tools for Question Answering. *Proceedings of the eleventh Text REtrieval Conference (TREC 2002).*

Moschitti, A. and S. Harabagiu (2004): A Novel Approach to Focus Identification in Question/Answering Systems. *In proceedings of the Workshop on Pragmatics of Question Answering at HLT-NAACL 2004*.

Narayanan, S. and S. Harabagiu (2004): Question Answering Based on Semantic Structures. *International Conference on Computational Linguistics (COLING 2004).*

Pérez-Coutiño, M., T. Solorio, et al. (2004): Toward a Document Model for Question Answering Systems. *Second International Atlantic Web Intelligence Conference, AWIC 2004*: 145-154.

Saquete, E., P. Martinez-Barco, et al. (2004): Splitting Complex Temporal Questions for Question Answering systems. *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics: ACL 2004*: 566-573.

Surdeanu, M., S. Harabagiu, et al. (2003): Using predicate-argument structures for information extraction. *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics* **1**: 8-15.

Tellex, S., B. Katz, et al. (2003): Quantitative Evaluation of Passage Retrieval Algorithms for Question Answering. Proceedings *of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*: 41 - 47.

Voorhees, E. M. (1994): Query Expansion Using Lexical-Semantic Relations. *Annual ACM Conference on Research and Development in Information Retrieval*: 61-69.

Voorhees, E. M. (2004): Overview of the TREC 2004 Question Answering Track. *The Thirteenth Text REtrieval Conference Proceedings (TREC 2004).*

Fillmore, Charles J. (1976): *Frame* semantics and the nature of language. *In Annals of the New York Academy of Sciences: Conference on the Origin and Development of Language and Speech* **280**: 20-32.