

Indexing on Semantic Roles for Question Answering

Luiz Augusto Pizzato

Centre for Language Technology
Macquarie University
Sydney, Australia
pizzato@ics.mq.edu.au

Diego Mollá

Centre for Language Technology
Macquarie University
Sydney, Australia
diego@ics.mq.edu.au

Abstract

Semantic Role Labeling (SRL) has been used successfully in several stages of automated Question Answering (QA) systems but its inherent slow procedures make it difficult to use at the indexing stage of the document retrieval component. In this paper we confirm the intuition that SRL at indexing stage improves the performance of QA and propose a simplified technique named the Question Prediction Language Model (QPLM), which provides similar information with a much lower cost. The methods were tested on four different QA systems and the results suggest that QPLM can be used as a good compromise between speed and accuracy.

1 Introduction

Semantic Role Labeling (SRL) has been implemented or suggested as a means to aid several Natural Language Processing (NLP) tasks such as information extraction (Kogan et al., 2005), multi-document summarization (Barzilay et al., 1999) and machine translation (Quantz and Schmitz, 1994). Question Answering (QA) is one task that takes advantage of SRL, and in fact much of the research about the application of SRL to NLP is related to QA. Thus, Narayanan and Harabagiu (2004) apply the argument-predicate relationship from PropBank (Palmer et al., 2005) together with the semantic frames from FrameNet (Baker et al., 1998) to create an inference mechanism to improve QA. Kaisser and Webber (2007) apply semantic

relational information in order to transform questions into information retrieval queries and further analyze the results to find the answers for natural language questions. Sun et al. (2005) use a shallow semantic parser to create semantic roles in order to match questions and answers. Shen and Lapata (2007) developed an answer extraction module that incorporates FrameNet style semantic role information. They deal with the semantic role assignment as a optimization problem in a bipartite graph and the answer extraction as a graph matching over the semantic relations.

Most of the studies that use SRL or similar techniques to QA apply semantic relation tools on the input or output of the Information Retrieval phase of their system. Our paper investigates the use of semantic information for *indexing* documents. Our hypothesis is that allowing Semantic Role information at the indexing stage the question analyzer and subsequent stages of the QA system can obtain higher accuracy by providing an implicit query analyzer as well as more precise retrieval. Theoretically, the inclusion of this information at indexing time can also speed up the overall QA process since syntactic rephrasing or re-ranking of documents based on semantic roles would not be necessary. However, SRL techniques are still highly complex and they demand a computational power that is not yet available to most research groups when working with large corpora. In our experience the annotation of a 3GB corpus, such as the AQUAINT (Graff, 2002), using a semantic role labeler, for instance SwiRL from Surdeanu and Turmo (2005) can take more than one year using a standard PC configuration¹.

In order to efficiently process a corpus with se-

© 2008. Licensed under the *Creative Commons Attribution-Noncommercial-Share Alike 3.0 Unported* license (<http://creativecommons.org/licenses/by-nc-sa/3.0/>). Some rights reserved.

¹Intel(R) Pentium(R) 4 HT 2.80GHz with 2.0 GB RAM

semantic relations, we have developed an alternative annotation strategy based on word-to-word relations instead of noun phrase-to-predicate relations. We define semantic triples based on syntactic clues; this approach was also studied by Litkowski (1999) but some major differences with our work are that we use automatically learned rules to generate the semantic relations, and that we use different semantic labels than those defined by Litkowski, some more specific and some more general. Our annotation scheme is named the Question Prediction Language Model (QPLM) and represents relations between pairs of words using labels such as *Who* and *When*, according to how one word complements the other.

In the following section we provide an overview of the proposed semantic annotation module. Then in Section 3 we detail the information retrieval framework used that allows the indexing and retrieval of semantic information. Section 4 describes the experimental setup and presents the results. Finally, Section 5 presents the concluding remarks and some discussion of further work.

2 Question Prediction Language Model

QPLM, as described in Pizzato and Mollá (2007), represents sentences by specifying the semantic relationship among its components using question words. In this way, we focus on dividing the problem of representing a large sentence into small questions that could be asked about its components. QPLM is expressed by triples $\theta(\omega) \rightarrow \alpha$ where θ is a question word, ω is the word that concerns the question word θ and α is the word that answers the relation θ about ω . For instance the relation $Who(eat) \rightarrow Jack$ tells us that the person who eats is Jack. The representation of our semantic relations as triples $\theta(\omega) \rightarrow \alpha$ is important because it allows the representation of sentences as directed graphs of semantic relations. This representation has the capacity of generating questions about the sentence being analyzed. Figure 1 shows such a representation of the sentence: “*John asked that a flag be placed in every school*”.

Having the sentence of Figure 1 and removing a possible answer α from any relation triple, it is possible to formulate a complete question about this sentence that would require α as an answer. For instance, we can observe that removing the node *John* we obtain the question “*Who asked for a flag to be placed in every*

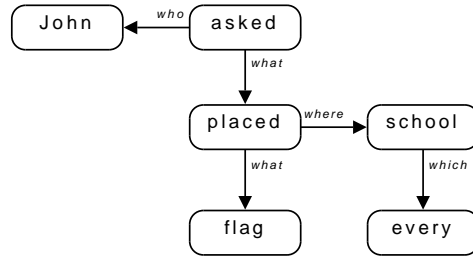


Figure 1: Graph Representation

school?” where *Who* was extracted from the triple $Who(ask) \rightarrow John$. The same is valid for other relations, such as removing the word *school* to obtain the question “*Where did John ask for a flag to be placed?*”. The name Question Prediction for this model is due to its capability of generating questions regarding the sentence that has been modeled.

We have developed a process to automatically annotate QPLM information, the process is rule based where the rules are automatically learned from a corpus obtained from mapping PropBank into QPLM instances. The mapping between semantic roles and QPLM is not one-to-one, which reduces the accuracy of the training corpus. A sample of 40 randomly selected documents was manually evaluated showing that nearly 90% of the QPLM triples obtained were correctly converted from the PropBank mapping. PropBank does not give us some relations that we wish to include such as ownership ($Whose(car) \rightarrow Maria$) or quantity ($HowMany(country) \rightarrow twenty$), but it does give us the benefits of a large training set covering a variety of different predicates.

Our QPLM annotation tool, like most SRL tools, makes use of a syntactic parser and a named-entity (NE) recognizer. We are currently using Connexor² for syntactic parsing and LingPipe³ for named-entity recognition.

An evaluation of our QPLM annotation has shown a reasonable precision (50%) with a low recall (24%). Both precision and recall seem to be connected with the choice of training corpus. The high precision is influenced by the large training set and the different variety of predicates. The low recall is due to the low amount of connections that can be mapped from one sentence in PropBank to QPLM. As we will present in Section 4, QPLM helps to improve results for QA even when

²<http://www.connexor.com>

³<http://alias-i.com/lingpipe/>

this training corpus is not optimal. This suggests that if a more suitable corpus is used to create the QPLM rules then we can improve the already positive results. An ideal training corpus would contain all QPLM pairs; not only verbs and head of noun phrases but also connections among all relevant words in a sentence.

3 Indexing and Retrieving Semantic Information

A document index that contains information about semantic relations provides a way of finding documents on the basis of meaningful relations among words instead of simply their co-occurrence or proximity to each other. A semantic relation index allows the retrieval of the same piece of information when queried using syntactic variations of the same query such as: “*Bill kicked the ball*” or “*The ball was kicked by Bill*”.

Several strategies can be used to build the indexing structure that includes relational information. The task of IR requires fast indexing and retrieval of information regardless of the amount of data stored and how it is going to be retrieved. From our experience, the use of relational databases is acceptable only if the amount of documents and speed of indexing and retrieval is not a concern. When database systems are used on large IR systems there is always a trade off between the speed of indexing and the speed of retrieval as well speed and storage efficiency.

The best approach for IR has always been a custom built inverted file structure. In the semantic role/QPLM case it is important to develop an indexing structure that can maintain the annotation information. Because it is important to allow different types of information to be indexed, we implemented a framework for information retrieval that easily incorporates different linguistic information. The framework allows fast indexing and retrieval and the implementation of different ranking strategies.

With the inclusion of relational information, the framework provides a way to retrieve documents according to a query of semantically connected words. This feature is best used when queries are formed as sentences in natural language. A simplified representation of the framework index is shown in Figure 2 for a QPLM annotated sentence. Figure 2 shows that the relation of words are represented by a common relation identifier and a re-

QPLM representation for “*Bill kicked the ball*”:

ID	Relation
11	<i>Who(kick) → Bill</i>
12	<i>What(kick) → ball</i>

Inverted file representation:

Term	Document	Rel. ID	Rel. Type	Role
Bill	1	11	Who	Arg
kick	1	11	Who	Pred
		12	What	Pred
ball	1	12	What	Arg

Figure 2: Simplified representation of the indexing of QPLM relations

Query	Returns documents that
<i>*(kick) → *</i>	contain the word kick
<i>Who(kick) → *</i>	inform that someone kicks
<i>Who(*) → Bill</i>	inform that Bill does an action
<i>Who(kick) → Bill</i>	inform that Bill kicks

Figure 3: QPLM Queries (asterisk symbol is used to represent a wildcard)

lation type. The roles that each word plays in a relation is also included within the same record. The IF is optimized so that redundant information is not represented, as illustrated by the record of the word *kick* and the single document number.

The framework also provides a way to include words that have not been explicitly related to other words in the text just in the same way as a standard bag-of-words (BoW) approach. This feature is important even when the text is fully semantically or syntactically parsed. Many words may not be associated with the others in a sentence because of different reasons such as errors in the parser. Therefore, even if the query presented to the retrieval component is not a proper natural language sentence or it fails to be analyzed, the system will perform as a normal BoW system.

Once the retrieval query is analyzed, it is possible to perform queries that focus on retrieving all documents where a certain relation occurs as well as all documents where a certain word plays a specific role. The example in Figure 3 demonstrates some queries and what documents or sentences they return.

A document containing the sentence “*Bill kicked the ball*” would be retrieved for all the queries in Figure 3. The framework also allows the formulation of more complex queries such as:

$$(Who(kick) \rightarrow *) \wedge (What(kick) \rightarrow ball)$$

Each token is indexed by itself (i.e not together with the related words) including the information from the relations it is part of. This is done with no overhead or redundant information being stored. This approach makes it possible to keep the standard models for document ranking. A normal calculation of Term Frequency (TF) and Inverted Document Frequency (IDF) is performed when taking the terms individually or as BoW, while only a minimal modification of TF/IDF is required when a more complex retrieval strategy is needed.

The ranking strategy is based on a vector space model. Documents and queries are represented as three different vectors: bag-of-words (BoW-V), partial relation (PR-V) and full relation (FR-V). The weights of the vector tokens are calculated using the weights of their individual tokens in the context of the vector being analyzed. In BoW-V, weights are calculated based on words; PR-V uses individual words and their relation types; FR-V uses the association of a specific word with another word. Figure 4 illustrates the contents of these vectors for the sentence “*John loves Mary, but Mary likes Brad*” when used as a query:

BoW-V:	\langle [John:1], [loves:1], [Mary:2], [likes:1], [Brad:1] \rangle
PR-V:	\langle [John:ARG0:1], [loves:PRED:1], [Mary:ARG1:1], [Mary:ARG0:1], [likes:PRED:1], [Brad:ARG1:1] \rangle
FR-V:	\langle [John:ARG0:loves:1], [Mary:ARG1:loves:1], [Mary:ARG0:likes:1], [Brad:ARG1:likes:1] \rangle

Figure 4: Vectors used for document ranking

The tokens of the above example would have different weights if the same sentence appeared in a document with additional sentences. Because of their lower frequency, it is expected that the components of FR-V and, in a lesser extent, of PR-V to have a stronger impact on the calculation of similarity than the components of BoW-V. With this approach, for queries with relations that are not indexed, the method is equivalent to a traditional BoW approach.

4 Experiments and Evaluation

We have performed a series of experiments using the techniques described on Section 3 in order to verify the usefulness of QPLM in comparison to

SRL based on PropBank. We compared both semantic annotations by using it with IR and under QA evaluation methods.

4.1 Configuration of experiments

We performed experiments using data resources from the QA track of the TREC conferences (Voorhees and Dang, 2006) and the evaluation scripts available at their TREC website of years 2004, 2005 and 2006. The retrieval experiments were carried out using only a reduced set of documents from the AQUAINT corpus because the semantic role labelers tested were not able to parse the full set, unlike QPLM which parsed all documents successfully.

The SRL tool SwiRL (Surdeanu and Turmo, 2005) has a good precision and coverage, however it is slow and quite unstable when parsing large amounts of data. We have assembled a cluster of computers in order to speed up the corpus annotation, but even when having around ten dedicated computers the estimated completion time was larger than one year. The lack of semantic annotators that can quickly evaluate large amount of data gave us the stimulus needed to use a simplified and quicker technique. We used the QPLM annotation tool which takes less than 3 weeks to fully annotate the 3GB of data from the AQUAINT corpus using a single machine.

Since we wanted to determinate how QPLM compares to SRL, particularly on the basis of its usage for IR and for QA, we performed some tests using the available amount of data annotated with semantic roles, and the same documents with QPLM. The part of the AQUAINT corpus annotated includes the first 41,116 documents, in chronological order, from the New York Times (NYT) newspaper. We used the 1,448 questions from the QA track of 2004, 2005 and 2006 from the TREC competition. Since these questions are not always self contained and in some cases (OTHER-type questions) not even a proper natural language sentence, we performed some question modification so that the entire topic text could be included. These modifications include substitution of key pronouns as well as the inclusion of the whole topic text when shorter representations were found. In some extreme cases when no substitution was possible and the question did not mention the topic, we added a phrase containing the topic at the start of the question. Some examples are presented

Topic:	Gordon Gekko
Question:	What year was the movie released?
Modification:	Regarding Gordon Gekko, what year was the movie released?
Question:	What was Gekko’s profession?
Modification:	What was Gordon Gekko’s profession?
Question:	Other
Modification:	Tell me more about Gordon Gekko.

Figure 5: Modifications applied to TREC questions

in Figure 5.

Using these questions as queries for our IR framework, we retrieved a set of 50 documents for every question. We analyzed the impact of the semantic annotation when used on document indices by checking the presence of the answer string in the documents returned. We also obtained a list of 50 documents using solely the BoW approach in order to compare what is the gain over standard retrieval.

4.2 Evaluation of retrieval sets

Table 1 presents the results of the retrieval set using TREC’s QA track from 2004, 2005 and 2006 using the BoW, the SRL and the QPLM approaches. Because we performed the evaluation of these documents automatically, we consider a document relevant on the only basis of the presence of the required answer string. We adopted the evaluation metrics for QA documents sets proposed by Roberts and Gaizauskas (2004). We used the following metrics: $p@n$ as the precision at n documents or percentage of documents containing an answer when retrieving at most n documents; $c@n$ as the coverage at n documents or percentage of questions that can be answered using up to n documents for each question; and $r@n$ as the redundancy at n document or the average number of answers found in the first n documents per question.

As observed in Table 1, the SRL approach gives the best results for all question sets on all evaluation metrics, with the exception of $c@50$ on the 2006 question set. In most other retrieval sets the baseline performs worse than both QPLM and SRL, however for 2004 questions it performed better than QPLM on $p@50$ and $r@50$. It is interesting to observe that the QPLM results for the same year on $c@50$ are better than the BoW approach indicating that a larger amount of questions can potentially be answered by QPLM.

2004	p@50	c@50	r@50
BoW	5.85%	33.33%	2.92
SRL	6.40%	35.33%	3.20
QPLM	5.58%	34.47%	2.79
2005	p@50	c@50	r@50
BoW	10.03%	41.13%	5.02
SRL	11.00%	43.77%	5.50
QPLM	10.58%	42.08%	5.29
2006	p@50	c@50	r@50
BoW	7.30%	34.57%	3.65
SRL	8.73%	36.33%	4.37
QPLM	8.31%	38.45%	4.16

Table 1: Experimental results of index approaches on TREC questions

4.3 Experiments on QA systems

To better understand the relation between the retrieved document sets and question answering we applied the retrieval sets to four question answering systems:

- Aranea: Developed by Lin (2007), the Aranea system utilizes the redundancy from the World Wide Web using different Web Search Engines. The system relies on the text snippets to generate candidate answers. It applies filtering techniques based on intuitive rules, as well as the expected answer classes with named-entities recognition defined by regular expressions and a fixed list for some special cases.
- OpenEphyra: Developed by Schlaefel et al. (2007), the OpenEphyra framework attempts to be a test bench for question answering techniques. The system approaches QA in a fairly standard way. Using a three-stage QA architecture (Question Analysis, Information Retrieval, Answer Extraction), it performed reasonably well at the QA Track at TREC 2007 by using Web Search engines on its IR stage and mapping the answers back into the TREC corpus.
- MetaQA System: Similar to the Aranea QA system, MetaQA (Pizzato and Molla, 2005) makes heavy use of redundancy and the information provided by Web Search Engines. However it goes a step further by combining different classes of Web Search engines (including Web Question Answering Systems) and assigning different confidence scores to each of the classes.

- AnswerFinder: Developed by Mollá and Van Zaanen (2006), the AnswerFinder QA system unique feature is the use of QA graph rules learned automatically from a small training corpus. These graph rules are based on the maximum common subgraph between the deep syntactic representation of a question and a candidate answer sentence. The graphs were derived from the output of the Connexor dependency-based parser.

For most of these systems some modifications of the standard system configuration were required. All the systems used, with the exception of AnswerFinder, make heavy use of web search engines and the redundancy obtained to find their answers. For our experiments we had to turn the Web search off, causing a significant drop in performance when compared to the reported results in the literature. Because AnswerFinder’s IR component is performed offline, the integration is seamless and only required providing the system with a list of documents in the same format as TREC distributes the ranked list of files per topic. The OpenEphyra framework is well designed and implemented, however the interaction between its components still depended on the overall system architecture, which makes the implementation of new modules for the system quite difficult.

With the exception of AnswerFinder, all the QA systems received a retrieval set as a collection of snippets. This was based on the fact that these systems are based on Web Retrieval and they expect to receive documents in this format. We extracted for every document the 255 character window where more question words (non-stopwords) were found. The implementation of different ranking strategies for passage retrieval such as those described by Tellex et al. (2003) could improve the results for individual QA systems. However, a preliminary evaluation of the passage retrieval have shown us that the 255 character window with the current snippet construction method was enough to achieve near optimal performance on the document set used.

The results obtained by the QA systems were processed using the answer regular expressions distributed by TREC. The numbers described in this study show the factoid score for correct answers. We have not used the exact answer because it required some cleaning of the answer log files and some modification of some QA systems.

	2004	2005	2006
BoW	5.00%	2.30%	2.10%
SRL	6.10%	3.50%	2.70%
QPLM	5.00%	2.50%	3.50%

Table 2: Factoid results for $C@1$ on the Aranea system

	2004	2005	2006
BoW	2.50%	5.10%	3.00%
SRL	3.30%	7.00%	4.40%
QPLM	2.80%	6.20%	4.20%

Table 3: Factoid results for $C@1$ on the OpenEphyra system

Therefore, the results shown on Tables 2, 3 and 5 are product of the same retrieval set and result of the same evaluation procedure. Results of the MetaQA system at Table 4 are presented as coverage at answer 10 ($C@10$) since this system has a non standard approach for QA that is invalidated by the methodology of this test. The results in the other tables could be understood as either precision or coverage at answer 1, we will refer to them as $C@1$.

We observed that the results from the QA system are consistent with the findings from the results of the retrieval system. The Aranea QA system results on Table 2 show an average improvement for the SRL approach. QPLM has similar performance to BoW for 2004 and 2005 questions but outperforms both techniques on 2006 questions.

The results shown by OpenEphyra in Table 3 also demonstrate that semantic annotation can help question answering when used in the IR stages of a QA system. The best results were observed when SRL was applied. QPLM followed SRL and outperformed BoW on three tests. It is important to point out that results for the retrieval set alone in Table 1 showed BoW outperforming QPLM for 2004 questions on both redundancy and precision metrics. This might be an indication that OpenEphyra answer extraction modules are more precise than the other QA systems and do not heavily rely on redundancy as do the Aranea and the MetaQA systems.

Because of the high dependency on Web sources, the MetaQA system performed rather poorly. As explained earlier, the results were measured using $C@10$ instead of $C@1$. The reason for this is that the MetaQA system is meant to be an aggregator of information sources and its ranking

	2004	2005	2006
BoW	0.87%	3.31%	1.24%
SRL	2.61%	3.87%	1.99%
QPLM	0.43%	3.31%	1.24%

Table 4: Factoid results for $C@10$ on the MetaQA system

	2004	2005	2006
BoW	1.10%	2.50%	1.20%
SRL	1.80%	2.60%	2.20%
QPLM	1.80%	2.70%	2.00%

Table 5: Factoid results for $C@1$ on the AnswerFinder system

mechanisms only work when sufficient evidence is given for certain entities. Not only was the system not designed for the single-source setup, but it was not designed to provide a single answer. Nevertheless, even with the non-conformity of the system, it appears to support that semantic markup can enhance the IR results for QA. Not surprisingly the extra redundancy presented in the 2004 BoW retrieval contributed to better results in this redundancy based QA system.

Results in Table 5 show that AnswerFinder correctly answered only a few questions for the given question set. On the other hand, it provided some consistent results such that the improvements were due to additional correct answers and not to a larger but different set of correct answers. The AnswerFinder QA system showed a similar performance for both semantic-based strategies and both outperformed the BoW strategy.

In this section we have shown an evaluation of different retrieval sets of documents using four distinct QA systems. We have observed that semantic strategies not only assist the retrieval of better documents, but also help in finding answers for questions when used with QA systems.

5 Concluding Remarks

In this work we propose the use of semantic relation in QA. We also present QPLM as an alternative to SRL. QPLM is a simpler approach to semantic annotation based on relations between pairs of words, which gives a large advantage in speed performance over SRL. We show some comparison of retrieval sets using the questions from the QA track of TREC and conclude that SRL and QPLM improve the quality of the retrieval set over a standard BoW approach. From these results we also observe that QPLM performance does not fall

much behind SRL.

We performed an evaluation using four QA systems. These systems are conceptually different which gives a broad perspective of the obtained results. The results once again show the effectiveness of semantic annotation. Over QA, SRL has performed better than the other techniques, but was closely followed by QPLM. The results obtained here suggest that QPLM is a cheaper and effective method of semantic annotation that can help in tuning the search component of a QA system to find the correct answers for a question.

The results presented in this work for all QA systems are much lower than those reported in the literature. This is an undesirable but expected problem that occurred not only because of the modifications carried on the QA systems but mainly because of the reduced number of documents used for this evaluation. We are looking into more efficient alternatives for performing the SRL annotation of the AQUAINT corpus.

Only recently we have been able to test Koomen et al. (2005) SRL tool. This SRL tool is the top ranking SRL tool at the CoNLL-2005 Shared Task Evaluation and it seems to be much faster than SwiRL. Preliminary tests suggest that it is able to perform the annotation of AQUAINT in almost one full year using a single computer; however, this tool, like SwiRL, is not very stable, crashing several times during the experiments. As further work, we plan to employ several computers and attempt to parse the whole AQUAINT corpus with this tool.

It is important to point out that although the tool of Koomen et al. seems much faster than SwiRL, QPLM still outperforms both of them on speed by large. QPLM represents word relations that are built using rules from syntactic and NE information. This simpler representation, combined with a smaller number of supporting NLP tools, allow QPLM to be faster than current SRL tools. We plan to carry out further work on the QPLM tool to increase its performance on both speed and accuracy. QPLM's precision and recall figures are going to be improved by using a hand annotated corpus. QPLM's speed suggest that it can be currently used on IR tools as a pre-processing engine. It is understandable that any delay in the IR phases is undesirable when dealing with large amount of data, therefore optimizing the speed of QPLM is one of our priorities.

Acknowledgement

This work was supported by an iMURS scholarship from Macquarie University and the CSIRO.

References

- Baker, Collin F., Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley framenet project. In *Proceedings of the 17th international conference on Computational linguistics*, pages 86–90, Morristown, NJ, USA. Association for Computational Linguistics.
- Barzilay, Regina, Kathleen R. McKeown, and Michael Elhadad. 1999. Information fusion in the context of multi-document summarization. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 550–557, Morristown, NJ, USA. Association for Computational Linguistics.
- Graff, David. 2002. The AQUAINT corpus of English news text. CDROM. ISBN: 1-58563-240-6.
- Kaisser, Michael and Bonnie Webber. 2007. Question answering based on semantic roles. In *Proceedings of the ACL 2007 Workshop on Deep Linguistic Processing*, page 4148, Prague, Czech Republic, June. c2007 Association for Computational Linguistics.
- Kogan, Y., N. Collier, S. Pakhomov, and M. Krauthammer. 2005. Towards semantic role labeling & IE in the medical literature. In *American Medical Informatics Association Annual Symposium*, Washington, DC.
- Koomen, P., V. Punyakanok, D. Roth, and W. Yih. 2005. Generalized inference with multiple semantic role labeling systems (shared task paper). In Dagan, Ido and Dan Gildea, editors, *Proc. of the Annual Conference on Computational Natural Language Learning (CoNLL)*, pages 181–184.
- Lin, Jimmy. 2007. An exploration of the principles underlying redundancy-based factoid question answering. *ACM Trans. Inf. Syst.*, 25(2):6.
- Litkowski, K. 1999. Question-answering using semantic relation triples. In *Proceedings of the 8th Text Retrieval Conference (TREC-8)*, pages 349–356.
- Molla, Diego and Menno van Zaanen. 2006. Answerfinder at TREC 2005. In *The Fourteenth Text REtrieval Conference (TREC 2005)*, Gaithersburg, Maryland. National Institute of Standards and Technology.
- Narayanan, Srinu and Sanda Harabagiu. 2004. Question answering based on semantic structures. In *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, page 693, Morristown, NJ, USA. Association for Computational Linguistics.
- Palmer, Martha, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational Linguist.*, 31(1):71–106.
- Pizzato, Luiz Augusto and Diego Molla. 2005. Extracting exact answers using a meta question answering system. In *Proceedings of the Australasian Language Technology Workshop 2005 (ALTA-2005)*, The University of Sydney, Australia, December.
- Pizzato, Luiz Augusto and Diego Mollá. 2007. Question prediction language model. In *Proceedings of the Australasian Language Technology Workshop 2007*, pages 92–99, Melbourne, December.
- Quantz, Joachim and Birte Schmitz. 1994. Knowledge-based disambiguation for machine translation. *Minds and Machines*, 4(1):39–57, February.
- Roberts, Ian and Robert J. Gaizauskas. 2004. Evaluating passage retrieval approaches for question answering. In McDonald, Sharon and John Tait, editors, *Advances in Information Retrieval, ECIR 2004, Sunderland, UK, April 5-7, 2004, Proceedings*, volume 2997 of *Lecture Notes in Computer Science*, pages 72–84. Springer.
- Schlaefler, N., P. Gieselmann, and G. Sautter. 2007. The ephyra qa system at trec 2006 the Ephyra QA system at TREC 2006. In *The Fifteenth Text REtrieval Conference (TREC 2006)*.
- Shen, Dan and Mirella Lapata. 2007. Using semantic roles to improve question answering. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 12–21, Prague, June 2007. Association for Computational Linguistics.
- Sun, R. X., J. J. Jiang, Y. F. Tan, H. Cui, T. S. Chua, and M. Y. Kan. 2005. Using syntactic and semantic relation analysis in question answering. In *Proceedings of the TREC*.
- Surdeanu, Mihai and Jordi Turmo. 2005. Semantic role labeling using complete syntactic analysis. In *Proceedings of CoNLL 2005 Shared Task*, June.
- Tellex, Stefanie, Boris Katz, Jimmy Lin, Aaron Fernandes, and Gregory Marton. 2003. Quantitative evaluation of passage retrieval algorithms for question answering. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 41–47, New York, NY, USA. ACM Press.
- Voorhees, Ellen M. and Hoa Trang Dang. 2006. Overview of the TREC 2005 question answering track. In *Text REtrieval Conference*.