Natural Language Based Reformulation Resource and Web Exploitation for Question Answering

Ulf Hermjakob, Abdessamad Echihabi, Daniel Marcu

Information Sciences Institute
University of Southern California
{ulf, echihabi,marcu}@isi.edu

Abstract

We describe and evaluate how a generalized natural language based reformulation resource in our TextMap question answering system improves web exploitation and answer pinpointing. The reformulation resource, which can be viewed as a clausal extension of WordNet, supports high-precision syntactic and semantic reformulations of questions and other sentences, as well as inferencing and answer generation. The paper shows in some detail how these reformulations can be used to overcome challenges and benefit from the advantages of using the Web.

1 Introduction

Question answering can be easy and it can be very hard. The degree of difficulty doesn't primarily depend on the question *per se*, but rather on how closely a given corpus matches the question.

Q:	Who discovered America?
S1:	Columbus discovered America.
S2:	Columbus Day celebrates the Italian navigator who first landed in the New World on Oct. 12, 1492.

The question above can be answered more easily from sentence S1 than sentence S2 because the string Q is "closer" to string S1 than string S2. Brill et al. (2001) have already demonstrated that a good methodology for increasing the performance of a QA system is that of using the Web as an additional textual resource. When a QA system looks for answers within a relatively small textual collection, the chance of finding strings/sentences that closely match the question string is small. However, when a QA system looks for strings/sentences that closely match the question string on the web, the chance of finding correct answers is much higher. And once a QA system knows the appropriate answer, it is much easier to find support for it in the original collection.

The approach proposed by Brill et al. reduces the gap between questions and answers by searching a larger pool of textual material. In this paper, we propose an alternative approach to reducing the gap between questions and potential answers. In our approach, we create semantically equivalent paraphrases of the question given as input and we assume that a correct answer is found whenever an answer sentence/string matches any of the paraphrases that are equivalent to the question given as input.

For example, the question "How did Mahatma Ghandi die?" is automatically paraphrased by our TextMap question answering system into 30 variants, some of which are shown below, along with paraphrases of two other questions.

1446.0:	How did Mahatma Gandhi die?	1439.0:	How deep is Crater Lake?
1446.1:	Mahatma Gandhi died <how>	1439.1:	Crater Lake is <what distance> deep
1446.2:	Mahatma Gandhi died of <what>	1439.2:	depth of Crater Lake is <what distance>
1446.3:	Mahatma Gandhi died from <what>	1439.3:	Crater Lake has a depth of <what distance>
1446.4:	Mahatma Gandhi's death from <what>	1439.4:	<what distance> deep Crater Lake
1446.5:	Mahatma Gandhi drowned		
1446.6:	Mahatma Gandhi suffocated	1772.0:	Who invented the cotton gin?
1446.7:	Mahatma Gandhi froze to death	1772.1:	<who> invented the cotton gin
1446.8:	<who> killed Mahatma Gandhi	1772.2:	<who> was the inventor of the cotton gin
1446.9:	<who> assassinated Mahatma Gandhi	1772.3:	<who>'s invention of the cotton gin
...	...	1772.4:	<who> was the father of the cotton gin
1446.30:	Mahatma Gandhi was killed	1772.5:	<who> received a patent for the cotton gin

As one can see, the paraphrases cover a fairly wide range of reformulations, from syntactic, such as reformulations 1446.1 and 1439.1, to semantic, such as reformulations 1439.2 and 1772.2. Some reformulations, such as 1446.8 and 1772.5 can be even interpreted as a rudimentary example of natural language inferences.

Our results show that the ability to paraphrase questions is useful with respect to two question answering subprocesses.

- First, question paraphrases can be used in conjunction with a retrieval engine in order to find documents that are more likely to contain correct answers than documents that are retrieved using standard query formulation techniques.
- Second, question paraphrases can be used in order to rank and select better answers than those that are selected by a system that does not have paraphrasing capabilities.

In this paper, we first describe briefly (Section 2) the techniques we use in order to generate paraphrases/reformulations. In Section 3, we present experiments that underscore the utility of question paraphrases in the context of retrieving documents that contain correct answers to questions. In Section 4, we evaluate the utility of our paraphrasing capability in the context of a complete question answering system.

2 Reformulations

To address the problem of mismatches going beyond relatively simple variations such as synonyms and alternative spellings, we are building a collection of phrasal synonyms in extended natural language format. Here are some examples of our reformulations:

```
:anchor-pattern "SOMEBODY_1 is the spouse of SOMEBODY_2." :reflexive t
:is-equivalent-to "SOMEBODY_1 is married to SOMEBODY_2." :reflexive t
:can-be-inferred-from "wedding of SOMEBODY_1 and SOMEBODY_2." :reflexive t
:can-be-inferred-from "SOMEBODY_1 married SOMEBODY_2." :reflexive t
:rebutted-by "SOMEBODY_1 and SOMEBODY_2 divorced." :reflexive t

:anchor-pattern "SOMETHING_1 costs MONETARY_QUANTITY_2."
:is-equivalent-to "the price of SOMETHING_1 is MONETARY_QUANTITY_2."
:is-equivalent-to "SOMETHING_1 is on sale for MONETARY_QUANTITY_2."
:can-be-inferred-from "to buy SOMETHING_1 for MONETARY_QUANTITY_2."

:anchor-pattern "SOMEBODY_1 sells SOMETHING_3 to SOMEBODY_2."
:is-equivalent-to "SOMEBODY_2 buys SOMETHING_3 from SOMEBODY_1."

:anchor-pattern "SOMEBODY_1 died of SOMETHING_2."
:is-equivalent-to "SOMEBODY_1 died from SOMETHING_2."
:is-equivalent-to "SOMEBODY_1's death from SOMETHING_2."
:answers "How did SOMEBODY_1 die?" :answer SOMETHING_2

:anchor-pattern "SOMEBODY_1 died from a specific cause." :intermediate-only
:can-be-inferred-from "SOMEBODY_1 drowned."
:can-be-inferred-from "SOMEBODY_1 suffocated."
:can-be-inferred-from "SOMEBODY_1 froze to death."
:answers "How did SOMEBODY_1 die?" :answer :full-pattern

:anchor-pattern "SOMEBODY_2 killed SOMEBODY_1."
:can-be-inferred-from "SOMEBODY_2 assassinated SOMEBODY_1."
:answers "How did SOMEBODY_1 die?" :answer :full-pattern :passive-answer
```

Our system first parses a given question and then identifies its answer type (Hovy 2001). The question reformulation module then uses parsed versions of reformulation patterns as shown above to produce high-precision meaning-preserving variants of the question. The objective of these reformulations is to increase the likelihood of finding correct answers in texts.

The number of reformulations produced by our current system varies from one reformulation (which might just rephrase a question into a declarative format) to about 30 reformulations, with an average of currently 3.14. The reformulation collection currently contains 420 assertions grouped into about 100 equivalence blocks. Many of them are manual generalizations of automatically derived patterns generated by (Ravichandran, 2002).

The principal advantage of using an extended natural language to express phrasal synonyms is that the format is very intuitive for humans. Phrasal synonyms are easy to write, or, when they are automatically generated, easily checked and filtered.

The expressiveness and focus of the patterns is greatly enhanced by variables that can carry syntactic or semantic restrictions. Compared to automatically generated patterns such as (Ravichandran, 2002) and (Lin, 2001), there is also no limit on the number of variables per reformulation and, since all patterns have been checked by hand, only few misreformulations.

In order to fully exploit the power of phrasal synonyms, our TextMap system parses all reformulation patterns and then performs the actual reformulation and matching at the parse tree level. This allows sentences and patterns to match even if the word order differs, as is often the case when sentences differ in mode (interrogative/declarative), voice (active/passive), or the presence of intervening constituents such as relative clauses or prepositional phrases.

Reformulations are useful even when they don't match any sentence at the surface level. Example:

Question: How deep is Crate Lake?
Reformulation: Crater Lake has a depth of <what distance>
Text: Crater Lake, with a depth of 1,932 feet, ...
Answer: 1,932 feet

While the text might not contain the exact phrasing of the reformulation, it does provide 'depth' as a valuable search term for the document retrieval engine and also allows the matcher to identify the correct answer with a higher confidence.

Our TextMap system uses the collection for the reformulation of questions, but the phrasal synonyms are not inherently question-oriented. The assertion

```
:anchor-pattern "SOMEBODY_1 sells SOMETHING_3 to SOMEBODY_2."
:is-equivalent-to "SOMEBODY_2 buys SOMETHING_3 from SOMEBODY_1."
```

for example also allows the reformulation from one declarative sentence to another:

- John sold the laptop to Mary.
- Mary bought the laptop from John.

By expressing reformulations in natural language, for both the reformulation patterns in the internal reformulation file and the reformulation output for a given question, the reformulation module is kept highly independent of other parts of a QA system, making it easily reusable for other QA architectures and other natural language applications.

2.1 Syntactic Reformulations

Even when reformulations are only fairly shallow (syntactic), the benefits turn out to be significant. All the reformulation mechanism might do is turn a question in interrogative word order such as "When did the Titanic sink?" into the corresponding declarative form: "The Titanic sank <when>". As described in much more detail in the next section of this paper, this can be greatly exploited with search engines that allow multi-word search strings.

Here are some more examples of relatively simple reformulations:

1439.0: How deep is Crater Lake?
1439.1: Crater Lake is <what distance> deep
1439.2: <what distance> deep Crater Lake
1439.3: Crater Lake has a depth of <what distance>
1439.4: depth of Crater Lake is <what distance>e

1782.0: Who was the first woman to run for president?
1782.1: <who> was the first woman to run for president
1782.2: the first woman to run for president was <who>
1782.3: <who>, the first woman to run for president

2.2 Advanced forms of reformulation

Reformulation patterns are general enough to support inferencing, reformulation chains, and generation.

2.2.1 Inference

While some assertions are truly equivalent, reformulation is often only a one-way street:

```
:anchor-pattern "SOMEBODY_1 invented SOMETHING_2."
:can-be-inferred-from "SOMEBODY_1 received a patent for SOMETHING_2." :weight 0.8
```

Question: Who invented the telephone?
Reformulation: <who> received a patent for the telephone
Text: Alexander Graham Bell received a patent for the telephone.
Answer: Alexander Graham Bell

2.2.2 Reformulation Chains

Reformulations can be chained together, as the following example shows:

```
:anchor-pattern "SOMEBODY_1 is a student at COLLEGE_2."
:answers "Where does SOMEBODY_1 go to college?" :answer COLLEGE_2

:anchor-pattern "SOMEBODY_1 was a student at COLLEGE_2."
:can-be-inferred-from "SOMEBODY_1 dropped out of COLLEGE_2."

:anchor-pattern "SOMEBODY_1 dropped out of COLLEGE_2."
:is-equivalent-to "SOMEBODY_1 is a COLLEGE_2 dropout."
```

Question: Where did Bill Gates go to college?
Reformulation: Bill Gates was a student at <which college>
Reformulation: Bill Gates dropped out of <which college>
Reformulation: Bill Gates is a <which college> dropout.
Text: Bill Gates is a Harvard dropout.
Answer: Harvard

2.2.3 Generation

Reformulations can also be used to generate answers that are more natural than the actual words in the underlying text:

```
:anchor-pattern "PERSON_1 invented SOMETHING_2."
:is-equivalent-to "PERSON_1 was the inventor of SOMETHING_2."
:is-equivalent-to "PERSON_1's invention of SOMETHING_2"
:answers "Who is PERSON_1 ?" :answer "the inventor of SOMETHING_2"
```

Question: Who was Johan Vaaler?
Reformulation: Johan Vaaler's invention of <what>
Text: ... Johan Vaaler's invention of the paper clip ...
Answer: the inventor of the paper clip

Question: Who was Johan Vaaler?
Reformulation: Johan Vaaler invented <what>
Text: ... the paper clip, invented by Johan Vaaler, ...
Answer: the inventor of the paper clip

2.3 Reformulation Patterns as a Qualitative Extension of WordNet

WordNet(Miller/Fellbaum, 1998) has proven to be a valuable resource for question answering and other natural language applications. Reformulation can be viewed as a qualitative extension of WordNet:

Word synonym	Phrasal synonym
astronaut	SOMETHING_1 is DISTANCE_QUANTITY_2 long.
cosmonaut	length of SOMETHING_1 is DISTANCE_QUANTITY_2.

Generalization	Phrasal inference
lawyer	SOMETHING_1 is the capital of LOCATION_2.
professional person	SOMETHING_1 is in LOCATION_2.

3 Information Retrieval and the Web

We have extended our system to make use of both the TREC collection and the Web. When querying the TREC collection, we use the IR system developed for Webclopedia (Hovy et al., 2001), which uses MG (Witten et al., 1994) as a search engine. When querying the Web we use a new Web-based IR system developed specifically to overcome the challenges and benefit from the advantages of using the Web. The main components of the Web-based IR consist of a Query Reformulation module, a Web search-engine, and a Sentence Ranking module.

3.1 Query Reformulation

One of the main challenges of using the Web as a resource for QA is the generation of Web search-engine queries from a natural language question. A couple of previous systems addressed this challenge. Simple but exhaustive string-based manipulations were used by Brill et al. (2001), Transformational Grammar was used by Kwok et al. (2001) and even learning algorithms were developed by Agichtein et al. (2001) and Radev et al. (2001) to find the best query reformulations.

Our approach was first to analyze how people will naturally form queries to find the answer to a given question. For this purpose, we carried out the following experiment: We randomly selected 50 TREC8 questions and, for each question, we manually produced the simplest queries that yielded the most Web pages containing the answer.

We analyzed the manually produced queries and identified seven "natural" techniques that we used to go from a natural language question to a Web search engine query. We enumerate these techniques below and provide for each an example of natural language question and corresponding Web query.

1. We quoted some question terms and we preserved already quoted terms:
   Name a film that has won the Golden Bear in the Berlin Film Festival?
   Film AND won AND "Golden Bear" AND "Berlin Film Festival"
2. We introduced the appropriate unit if the answer is a quantity:
   How tall is Mt. Everest?
   "Mt. Everest" AND feet
3. We quoted and used parts of the declarative form of the question:
   What is the name of the highest mountain in Africa?
   "the highest mountain in Africa is"
4. We reformulated the declarative form of the question, quoted and used parts of it:
   Where is the Taj Mahal?
   "the Taj Mahal is located"
5. We expanded quoted terms with appropriate prepositions:
   When did Jaco Pastorius die?
   "Jaco Pastorius died on" OR "Jaco Pastorius died in"
6. We used different spellings of some question terms:
   Who was the 16th President of the United States?
   "was the 16th President of the United States" OR "was the 16th President of the U.S."
7. We restricted the Web search to specific Web sites based on the question type:
   Who was the lead actress in the movie "Sleepless in Seattle"?
   site:www.imdb.com actress AND "Sleepless in Seattle"

We, then, derived algorithms that replicate each of the first six observed techniques. We left the last technique for future work. We have also implemented the standard query expansion by morphological variants and synonyms. Table 1 summarizes the different query reformulation techniques used by our Web-IR.

3.2 Sentence Ranking

Using all the query reformulation techniques described earlier, we produce, for each question, a list of Boolean queries (an average of 8 queries per question on TREC-2002). Then, using a Web search-engine, we retrieve the top 10 results (URLs + snippets) for each query. After filtering duplicate URLs, we retrieve the documents, strip HTML, and segment text into sentences.

Every sentence is then scored according to two different schemas:

Name	*Description*	*Sample System Output*
Simple	Preserve quoted terms and quote the smallest NPs.	**What is the longest river in the United States?** -"longest river" AND "United States"
Units	Expand the query with potential units in answer based on answer type and question context.	**How tall is Mt. Everest?** -"Mt. Everest" AND "tall" AND ("foot" OR "feet" OR "miles")
Morpho	Expand the query using morphological variants produced by the Contex parser.	**When did the Titanic sink?** -"Titanic" AND ("sink" OR "sank" OR "did sink")
Synonym	Expand the query using WordNet synonyms.	**What is the length of border between Ukraine and Russia?** -("length" OR "distance") AND ("border" OR "surround") AND ("Ukraine" OR "Ukrayina") AND ("Russia" OR "Soviet Union") AND ("between" OR "betwixt")
Spelling	Expand the query using different spellings of question terms.	**Where is Al-Qaeda located?** -("Al-Qaeda" OR "Al Qaida") AND "located"
Rephrase	Using Contex reformulations, add quoted rephrases of the question's declarative form.	**What is an atom?** -"is an atom" -"an atom is" -"an atom is one of" -"an atom is defined to be" -"an atom is defined as" -"such as an atom" -"called an atom"
Cuephrase	Expand the rephrases with prepositions and/or discourse cues based on answer type.	**When did Abraham Lincoln die?** -"Abraham Lincoln died" -"Abraham Lincoln died on" -"Abraham Lincoln died in"

Table 1: Query Reformulation Techniques

- Scoring with respect to the question/queries terms:
  - Each word in a query is assigned a weight using an information content measure.
  - Each quoted term in a query is given a weight, which is the sum of the weight of its words.
  - Sentences are then scored according to their weighted overlap with the question/queries terms.
- Scoring with respect to the answer:
  - Sentences are tagged using the BBN's IdentiFinder (Bikel et al. 1999).
  - Sentences are then scored according to their overlap with the answer type. The overlap is measured by checking the answer type against IdentiFinder semantic entities in the sentence candidates.

3.3 Web-IR Evaluation:

We have evaluated our Web-IR on the first 200 questions from TREC-2002. We started with the simplest IR system (Simple). And then we added to the system increasingly sophisticated query expansion techniques. The system label Morpho, for example, uses the query expansion techniques that were labeled above as Simple, Unit, and Morpho. Figure 1 shows the percentage of questions that can be answered correctly if one manually selects a sentence from the top 300 or top 10 sentences as ranked by our retrieval component. The results in figure 1 show that more sophisticated query reformulation techniques lead to better candidates for the answer pinpointing module. The most significant improvement occurs when the query rephrasing is added to the pool of query expansion techniques. The clear difference between the percentage of answers found within the top 300 sentences and within the top 10 sentences indicates that further improvement of the sentence ranking module is required.

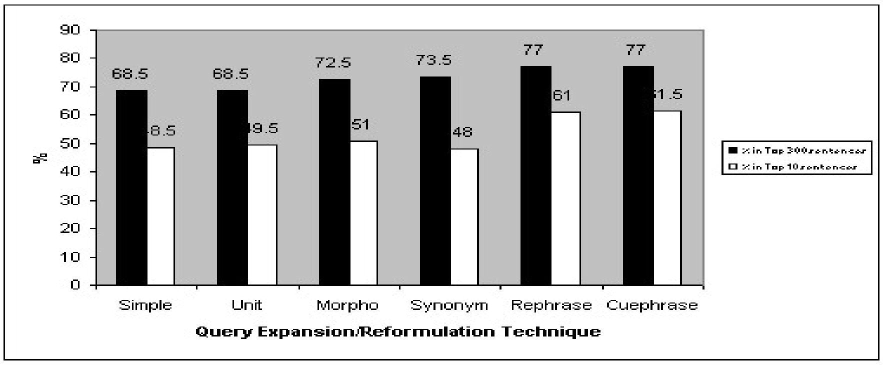

Figure 1: Percentage of answers found in top 300 and top 10 sentences using AltaVista with different query reformulation techniques.

Figure 2 shows how the query reformulation techniques affect the answer redundancy. Intuitively, the more redundant an answer is, the more likely a question answering system is able to find it. Once again, the significant improvement is recorded when the rephrase technique was introduced.

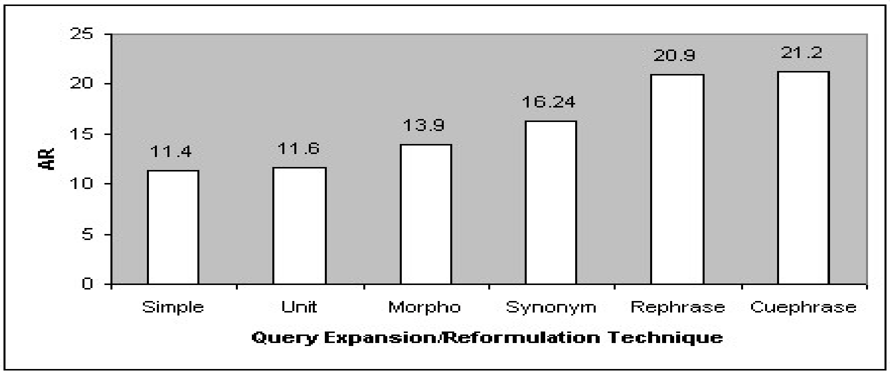

Figure 2: Average Answer Redundancy using AltaVista with different query reformulation techniques.

Finally, figure 3 shows the impact of the search engine choice on the performance of our Web-IR.

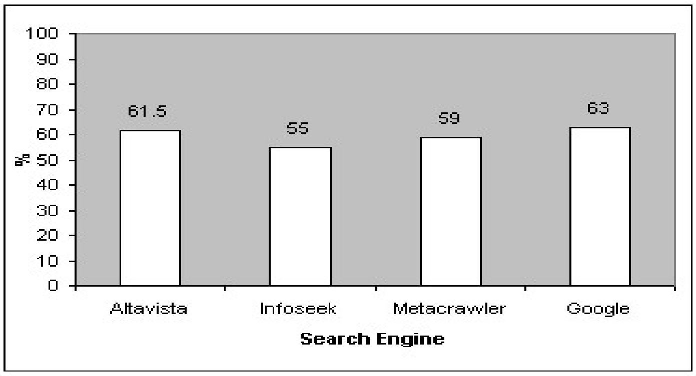

Figure 3: Percentage of answers found in top 10 sentences using different search engines.

4 Evaluation of overall system

We also evaluated the impact of reformulation on overall performance of our system[1]. In the six configurations in table 2, we tested our system on (1) the TREC-2002 collection only, (2) the Web only and (3) the TREC-2002 collection with feedback from the Web, both with (+) and without (-) reformulations.

The results in Table 2 show that our reformulations did not have a significant impact when used in conjunction with a small corpus, the TREC-2002 corpus. The increase in performance for this corpus was small, from 144 to 146 correct answers. However, when used in conjunction with a large corpus, the Web, reformulations enabled us to find many more correct answers (227 compared to 173). Retrofitting the answer to the TREC-2002 collection reduces the performance of our system, but the impact of our reformulation capability remains strong (176 vs. 147) correct answers. These results were achieved with a version of the reformulation module that produced, on average, 1.2 reformulations per question.

Collection	reformulations	**R**	R+X	R+U	R+X+U	W
TREC only	-	**144**	155	148	160	340
Web only	-	**173**	198	173	198	302
Web+TREC	-	**147**	160	163	178	322
TREC only	+	**146**	157	151	163	337
Web only	+	**227**	251	227	251	249
Web+TREC	+	**176**	187	195	208	292

Note: R stands for the number of questions with a fully correct answer, X for correct but inexact, U for correct but unsupported, and W for wrong.

Table 2: Overall System Evaluations (July 2002 system)

Since July, we have expanded on our reformulation module significantly, motivated by questions and answers from TREC-9. Currently, we are averaging about 3.14 reformulation per question. When we re-evaluated the performance of our system on the first 100 questions in the TREC-2002 collection, we observed that this increase in reformulation capability led to a 5% increase in overall system performance (see Table 3 below).

Collection	reformulations	#ref/q	**R**	R+X	R+U	R+X+U	W
Web+TREC	+ (July)	1.2	**34**	38	38	42	58
Web+TREC	+ (Oct.)	3.1	**39**	44	41	46	54

Table 3: Effect of additional reformulations

5 Conclusion

Question reformulations increase the likelihood that correct answers are properly identified by (1) enabling the information retrieval module to produce higher quality answer candidates, particularly if the underlying search engine can handle multi-word strings. (2) Reformulations also increase the scoring precision for answer candidates and improve answer pinpointing in the QA matching phase. (3) Finally, even if the QA system would have selected a correct answer without reformulations, a strong match with a reformulation as opposed to a weaker match with the original question can provide additional confidence in the correctness of the answer.

References

Agichtein, E., S. Lawrence and L. Gravano. Learning Search Engine Specific Query Transformations for Question Answering. WWW10, 2001.

Banko M., E. Brill, S. Dumais, J. Lin 2001. AskMSR: Question Answering Using the Worldwide Web In *Proceedings of the TREC-2001 Conference*, NIST. Gaithersburg, MD

Bikel, D., R. Schwartz, and R. Weischedel. 1999. An Algorithm that Learns What's in a Name.*Machine Learning-Special Issue on NL Learning* 34, 1-3.

[1] TextMap is the successor system to Webclopedia (Hovy 2001).

Fellbaum, C. Editor; G. Miller, preface 1998. MIT Press.
URL: http://www.cogsci.princeton.edu/~wn/

Hovy, E., L. Gerber, U. Hermjakob, C.-Y. Lin, D. Ravichandran 2001. Towards Semantics-Based Answer Pinpointing In *Proceedings of the HLT 2001 Conference*, San Diego

Lin, D. and P. Pantel 2001. Discovery of Inference Rules for Question Answering. In *Journal of Natural Language Engineering 7(4)* pp. 343-360.

Kwok, C., O. Etzioni and D. Weld. Scaling Question Answering to the Web. WWW10, 2001.

Radev, D., H. Qi, Z. Zheng, S. Blair-Goldensohn, Z. Zhang, W. Fan and J. Prager. Mining the Web for Answers to Natural Language Questions. ACM CIKM 2001.

Ravichandran, D. and E. Hovy 2002. Learning Surface Text Patterns for a Question Answering System In *Proceedings of the 40th meeting of the Association for Computational Linguistics*, Philadelphia, PA

Witten, I.H., A. Moffat, and T.C. Bell. 1994. Managing Gigabytes: Compressing and indexing documents and images. New York: Van Nostrand Reinhold.