Answering complex, list and context questions with LCC's Question-Answering Server

Sanda Harabagiu*, Dan Moldovan*,
Marius Paşca*, Mihai Surdeanu*, Rada Mihalcea, Roxana Gîrju, Vasile Rus,
Finley Lăcătuşu, Paul Morărescu and Răzvan Bunescu
Language Computer Corporation*
Dallas, TX 75206
{sanda,moldovan}@languagecomputer.com

Abstract

This paper presents the architecture of the Question-Answering Server (QAS) developed at the Language Computer Corporation (LCC) and used in the TREC-10 evaluations. LCC's QAS^{TM} extracts answers for (a) factual questions of vairable degree of difficulty; (b) questions that expect lists of answers; and (c) questions posed in the context of previous questions and answers. One of the major novelties is the implementation of bridging inference mechanisms that guide the search for answers to complex questions. Additionally, LCC's QAS^{TM} encodes an efficient way of modeling context via reference resolution. In TREC-10, this system generated an RAR of 0.58 on the main task and 0.78 on the context task.

Introduction

Systems providing question-answering services need to process questions of variable degrees of complexity, ranging from inquiries about definitions of concepts, e.g. *"What is semolina?"* to details about attributes of events or entities, e.g. *"For how long is an elephant pregnant?"*. Finding the answer to questions often involves various degrees of *bridging inference*, depending on the formulation of the question and the actual expression of the answer extracted from the underlying collection of documents. For example, the question *"How do you measure earthquakes?"* is answered by the following text snippet extracted from the TREC collection: *" Richter scale that measures earthquakes"* because the required inference is very simple: a measuring scalar, i.e. *Richer scale*, has a relative adjunct introduced by the same verb as in the question, having the same object of measurement. Yet a different, more complex form of inference is imposed by questions like *"What is done with worn and outdated flags?"*.

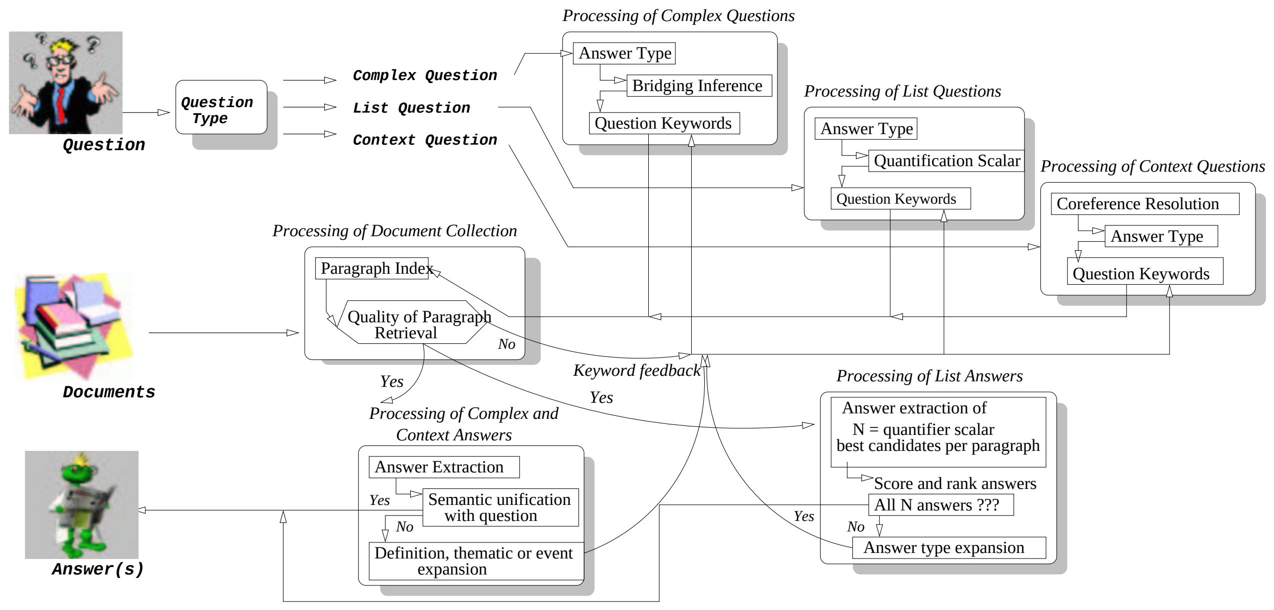
The Question-Answering Server (QAS^{TM}) developed at the Language Computer Corporation (LCC) encodes methods of performing several different bridging inferences that recognize the answer to questions of variable degree of complexity. The pragmatic knowledge required by different forms of inference is distributed along the three main modules of LCC's QAS^{TM}: the *Question Processing* module, the *Document Processing* module and the *Answer Processing* module. Some of the inference forms enabled by LCC's QAS^{TM} determine the answer fusion mechanisms that assemble list-answers expected by questions like *"Name 20 countries that produce coffee."*

Rarely questions are asked in isolation. When satisfied by the answer, a user may have follow-up questions, requiring additional information. If the answer is not satisfactory, a new question may clarify the user's intentions, thus enabling a better disambiguation of the question. LCC's QAS^{TM} is capable of answering questions in context, thus exploiting the common ground generated between the answer of questions like *"Which museum in Florence was damaged by a major bomb explosion in 1993?"* and its follow-up questions *"On what day did this happen?"* or *"Which galleries were involved?"*. These new capabilities of (a) answering more complex questions than those evaluated in TREC-8 and TREC-9; (b) detecting when a question does not have an answer in the collection; (c) fusing several answers that provide partial information for questions expecting list-answers; and (d) answering questions in context - stem from a new architecture, that enhances the three-module streamlined operation used in the previous TREC Q/A evaluations[1].

The architecture of LCC's QAS^{TM}

The architecture of LCC's QAS^{TM} used in the TREC-10 evaluations is illustrated in Figure 1. Three dif-

[1]In TREC-8, the Q/A evaluations showed that the best performing systems exploited the combination of Named Entity semantics with the semantic of question stems. In TREC-9, two trends could be observed: (1) systems that used advanced pragmatic and semantic knowledge in the processing of questions and answers, and (2) systems that improved on new ways of indexing and retrieving the paragraphs were the answers may lie.



Question
Question Type
Complex Question
List Question
Context Question
Processing of Complex Questions
Answer Type
Bridging Inference
Question Keywords
Processing of List Questions
Answer Type
Quantification Scalar
Question Keywords
Processing of Context Questions
Coreference Resolution
Answer Type
Question Keywords
Processing of Document Collection
Paragraph Index
Quality of Paragraph Retrieval
No
Yes
Keyword feedback
Yes
Documents
Processing of Complex and Context Answers
Answer Extraction
Yes
Semantic unification with question
No
Definition, thematic or event expansion
Answer(s)
Processing of List Answers
Answer extraction of N = quantifier scalar best candidates per paragraph
Score and rank answers
All N answers ???
Yes
No
Answer type expansion

Figure 1: Architecture of LCC's QAS^{TM}

ferent kinds of questions were evaluated: (1) *complex questions*, that expect an answer from the text collections without knowing if such an answer exists; (2) *list questions*, requiring a list of answers; and (3) *context questions*, in which the question was considered in the context of the previous questions and answers processed by the system. Three distinct evaluations were conducted, but a single question-answering architecture handled all three cases.

The *Question Processing* is different for each of the three kinds of questions that were evaluated. For complex, factual questions like *"Q1147: What is the Statue of Liberty made of?"*, the processing involves at first the recognition of the *expected answer type* from an off-line taxonomy of semantic types. In TREC-10, the factual questions were far more complex than those evaluated in TREC-9 and TREC-8 because frequently the expected answer type could not be easily identified. For example, in the case of the question Q1147 virtually anything could be a criterion. To help narrow down the search for the expected answer type and to generate robust processing at the same time, a set of bridging inference procedures were encoded. For example, in the case of the question Q1147, the bridging inference between the question and the expected answer type encode several meronymy relations between different materials and the Statue of Liberty. Instead of searching for the expected answer type in each retrieved paragraph LCC's QAS^{TM} looks for meronymy relations involving any of the keywords used in the query.

For questions expecting a list of answers, the quantification scalar, defining the size of the list, is identified at the time of question processing and used when the answers are extracted and fused together. For example, in the case of question *"Name 15 religious cults."* the expected answer type is ORGANIZATION of the type *religious cult* and the quantifier is 15. Sometimes, the expected answer type have multiple attributes, e.g. *"Name 4 people from Massachusetts who were candidates for vice-president."* Such attributes are translated into keywords that retrieve the relevant document passages or paragraphs.

If the question needs to be processed in the context of the previous questions and answers, a coreference resolution process takes place prior to the recognition of the expected answer type. For example, the pronoun *this* from the question *"On what day did this happen?"* is resolved as the event mentioned in its preceding question, i.e. *"Which museum in Florence was damaged by a major bomb explosion in 1993?"*. The reference resolution entails the usage of the keywords defining the antecedent along with the keywords extracted from the current question.

The *Document Processing* module uses a paragraph index to retrieve document passages that (a) contain the keywords from the query, and (b) contain either a concept of the expected answer type or a relation indicated by the bridging inference mechanisms. However, if insufficient evidence of the paragraph relevance exists, pragmatic information is passed back to the feedback loop that reformulates the query searching for the complex answer. When the most relevant paragraphs are retrieved, the answers are processed.

When answers to complex, factual questions are extracted, their validity is granted by semantic unifications with the question. If the question was asked in context, the unifications of previous questions and answers are also used to grant the validity of the answer of the current question. When unifications are not possible, several expansions that use the gloss definitions of the WordNet concepts are considered. The question is ruled not to have an answer when none of the expansions generate unifications. The processing of list answers is performed differently because LCC's QAS^{TM} extracts for each question all N best candidate answers, where N is the quantifier scalar. Additional answers are sought only if we could not find all N answers and if variations of the keywords defining the same answer type are possible.

Processing questions from the main task

Two main trends have characterized the main task in TREC-10. First, the percentage of questions that ask for definitions of concepts, e.g. *"What are capers?"* or *"What is an antigen?"* represented 25% of the questions from the main task, an increase from a mere 9% in TREC-9 and 1% in TREC-8 respectively. The definition questions normaly require an increase in the sophistication of the question-answering system. Second, in general, the questions had an increased level of difficulty. Questions like *What is the esophagus used for?"* or *"Why is the sun yellow?"* are difficult to process because the answer relies on expert knowledge, from medicine in the former example, and from physics in the latter one. Nevertheless, if a lexical dictionary that explains the definitions of concepts is available, some supporting knowledge can be mined. For example, by inspecting WordNet (Miller 1995), in the case of *esophagus* we can find that it is *"the passage between the pharynx and the stomach"*. Moreover, WordNet encodes several relations, like *meronymy*, showing that the esophagus is part of the *digestive tube* or *gastrointestinal tract.* The glossed definition of the *digestive tube* shows that one of its function is the *digestion.*

The information mined from WordNet guides several processes of bridging inference between the question and the expected answer. First the definition of the concept defined by the WordNet synonym set {*esophagus, gorge, gullet*} indicates its usage as a *passage* between two other body parts: the pharynx and the stomach. Thus the query *" esophagus AND pharynx AND stomach"* retrieves all paragraphs containing relevant connections between the three concepts, including other possible functions of the esophagus. When the query does not retrieve relevant paragraphs, new queries combining esophagus and its holonyms (i.e. gastrointestinal tract) or functions of the holonyms (i.e. digestion) retrieve the paragraphs that may contain the answer. To extract the correct answer, the question and the answer need to be semantically unified.

Q912: What is epilepsy?
Q1273: What is an annuity?
Q1022: What is Wimbledon?
Q1152: What is Mardi Gras?
Q1160: What is dianetics?
Q1280: What is Muscular Distrophy?

Table 1: Examples of definition questions.

The difficulty stands in resolving the level of precision required by the unification. Currently, LCC's QAS^{TM} considers an acceptable unification when (a) a textual relation can be established between the elements of the query matched in the answer (e.g. esophagus and gastrointestinal tract); and (b) the textual relation is either a syntactic dependency generated by a parser, a reference relation or it is induced by matching against a predefined pattern. For example, the pattern "X, particularly Y" accounts for such a relation, granting the validity of the answer *"the upper gastrointestinal tract, particularly the esophagus"*. However, we are aware that such patterns generate multiple false positive results, degrading the performance of the question-answering system.

Definition pattern	*Phrase to be defined (QP)*	*Candidate answer phrase (AP)*
<*AP*> such as <*QP*>	What is autism?	developmental disorders such as autism
<*AP*> (also called <*QP*>)	What is bipolar disorder?	manic-dipressive illness (also called bipolar disorder)
<*QP*> is an <*AP*>	What is caffeine?	caffeine is an alkaloid

Table 2: Identifying candidate answers with pattern matching.

Predefined patterns are also important for processing definition questions, similar to those listed in Table 1. Table 2 lists several patterns and their components: the question phrase (QP) that requires a definition, and the candidate answer phrase (AP) providing the definition. To process definition questions in a more robust manner, the search space is enlarged, allowing the substitution of the phrase to be defined QP with the immediate hypernym of its head. Table 3 illustrates several examples of definition questions that are resolved by the substitution of the QP with its hypernym. The usage of WordNet hypernyms builds on the conjecture that any concept encoded in a dictionary like WordNet is defined by a *genus* and a *differentia*. Thus when asking about the definition of a concept, retrieving the genus is sufficient evidence of the explanation of the definition, especially when the genus is identical with the hypernym.

Phrase to be defined (QP)	*Hypernym from* WordNet	*Candidate answer phrase (AP)*
What is a <u>shaman</u>?	{priest, non-Christian priest}	Mathews is the <u>priest</u> or shaman
What is a <u>nematode</u>?	{worm}	nematodes, tiny <u>worms</u> in soil.
What is <u>anise</u>?	{herb, herbaceous plant}	aloe, anise, rhubarb and other <u>herbs</u>

Table 3: WordNet information employed for detecting answers of defintion questions.

In WordNet only one third of the glosses use a hypernym of the concept being defined as the genus of the gloss. Therefore, the genus, processes as the head of the first NP from the gloss, can also be used to substitute the QP of the definition question. For example, the processing of question Q1273 from Table 1 relies on the substitution of *annuity* with *income* the genus of its WordNet gloss, rather than its hypernym, the concept *regular payment*. The availability of the genus or hypernym helps also the processing of definition questions in which the QP is a named entity, as it is in the case of questions Q1022 and Q1152 from Table 1. In this way *Wimbledon* is replaced with *a suburb of London* and *Mardi Gras* with *holiday*. The processing of definition question is however hindered by the absence of the QP head from the WordNet database. For example, both *diametics* from Q1160 and *Muscular Distrophy* from Q1280 are not encoded in WordNet.

Processing list questions

Unlike complex, factual questions, list questions expect a list of answers. The length of the list is specified by a *quantification scalar* that has to be identified in the natural language question. All the elements of the list must be valid answers to the question and, in addition, the list cannot have duplicate items.

The extraction of the answers depends in large measure on the recognition of the expected answer type. 21 out of 25 test questions (84%) asked about categories that are easily matched by Named Entity recognizers, e.g. countries, cities, people, organizations and currencies. LCC's QAS^{TM} uses a robust answer extraction technology that enables it to extract candidate answers even when the expected answer is unknown. A combination for the keyword features (e.g. the distance between the keywords in the paragraph) enables the server to pinpoint possible answers. However, for processing list answers two additional enhancements had to be encoded.

First we had to allow the extraction of multiple candidate answers from each paragraph. Table 4 illustrates two different paragraphs containing multiple elements of the answer list expected by two distinct questions. The first paragraph contains two elements of the answer list whereas the second paragraph contains three elements of the answer list. Each paragraph has a relevance score associated with it, enabling an ordering of the answers based on the relevance score of its original paragraph. The answer list is assembled by collecting the first N ranked answers, where N is the quantification scalar identified in the question.

Question: Name 20 countries that produce coffee.
FT944-7920: It would also co-ordinate assistance for countries such as <u>*Angola*</u> and <u>*Rwanda*</u>, whose coffee sectors have been badly damaged by war or climatic disasters.
Question: Name 10 countries that banned beef imports from Britain in the 1990s.
AP900601-0140: <u>*West Germany*</u> and <u>*Luxembourg*</u> joined <u>*France*</u> on Friday in banning British beef imports because of "mad cow" disease.

Table 4: Paragraphs containing multiple candidate answers.

Second, we had to discard duplicate candidate answers from the answer list. This operation improves the recall of the answer lists. To this end we implemented an *answer normalization* procedure, encoding two functions: (1) name alias recognition, identifying *United States of America*, *USA*, *U.S.* and *US* as the same entity; and (2) distinguish separate entities bearing the same name, e.g. *Paris, France* and *Paris, TX*. Table 5 illustrates several text snippets that contain duplicate candidate answers for the same question.

Context questions

Processing a sequence of questions posed in the same context requires the resolution of several forms of reference. A question may use:

1. *demonstrative pronouns*, like this, these or there; (e.g. "On what day did <u>this</u> happen?", or "Where were <u>these</u> people located?" or "Name a company that flies <u>there</u>?")
2. *third person pronouns*, like he or it; (e.g. "What California winery does <u>he</u> own?" or "In what facility was <u>it</u> constructed?")

Question: Name 20 countries that produce coffee.
First instance non-duplicate (FT944-2823): in *Brazil*. Mr. Jorge Cardenas, head of the
Duplicate (FT933-1482): *Brazil*, intends to cover exporters' costs
Duplicate (WSJ911203-0140): in *Brazil*, said producers told her there has
Duplicate (AP900718-0272): notably *Brazil*, the world's largest producer
Duplicate (WSJ870602-0079): said *Brazil*, the world's largest coffee

Table 5: 50-byte text snippets containing duplicate candidate answers.

3. *possessive pronouns*, like his or its; (e.g. "What was his first radio song?")
4. *definite nominals*, in which the definite article or the demonstrative pronoun indicate that the concept was already introduced by a previous question or answer; (e.g. "What executive from the company was a member of the Supreme Council in 1994?" or "This city's name was later changed to what?")
5. *nominalizations* of verbs used in previous questions; (e.g. "When was construction begun?" following "In what facility was it constructed?")
6. *elliptical* reference, in which the expected answer type is inherited from the previous question; (e.g. "How many are poisonous to humans?" following "How many species of spiders are there?")
7. *causal-effect* reference; e.g. explosive from "How much explosive was used?" is the cause of explosion from its preceding question "Which museum in Florence was damaged by a major bomb explosion in 1993?"
8. *meronymic* reference, e.g. galleries from "Which galleries were involved?" are referenced as a part of the *museum* from the preceding question "Which museum in Florence was damaged by a major bomb explosion in 1993?".

The resolution of all the forms of reference is performed by identifying the antecedent of the anaphora in (1) a previous question; or (2) the answer to a previous question; or (3) an anaphor used in a previous question. Before applying the reference resolution algorithm, the pleonastic usage of pronouns is identified, ruling out the resolution of pronouns like *there* in "How many species of spiders are there?"

The reference resolution algorithm employed by LCC's QAS^{TM} is different from reference resolution algorithms used in discourse or dialog processing because the goal is not to resolve the reference, but to identify the question that either contains the antecedent of the reference or expects an answer that contains the antecedent. Consequently, when processing the question Q_1 that contains a reference, by knowing which preceding question Q_0 generates the antecedent, we can combine the keywords of Q_1 with the keywords of Q_0 to retrieve relevant paragraphs. Moreover, since question keywords are extracted in a predefined order in QAS^{TM}, when keywords from two different questions are combined, the keywords from the previous question always preceded the keywords from the current question. This keyword ordering is important for the feedback loops implemented in LCC's QAS^{TM}, illustrated in Figure 1. For example Table 6 illustrates the combination of keywords resulting from the reference resolution within context questions.

Example 1
Question CTX1d: How many people were killed?
Keywords from CTX1d: (k_1=killed)
Reference of question CTX1d = question CTX1a: Which museum in Florence was damaged by a major bomb explosion in 1993?
Keywords from CTX1a: (k_2=Florence, k_3=bomb, k_4=explosion)
Keywords used to process CTX1d: (k_2=Florence, k_3=bomb, k_4=explosion, k_1=killed)
Example 2
Question CTX7g: How wide?
Keywords from CTX7g: (k_1=wide)
Reference of question CTX7g = question CTX7a: What type of vessel was the modern Varyag?
Keywords from CTX7a: (k_2=Varyag)
Keywords used to process CTX7g: (k_2=Varyag, k_1=wide)

Table 6: Keyword extraction for context questions.

The algorithm that performs reference resolution for context questions is:

```
Algorithm Reference Context Resolution(Q)
Input: LQ = precedence-ordered list of previous
    questions asked in the same context + $w_Q$,
    where $w_Q$ = the reference word from Q
    when we have an ellipsis $w_Q = \phi$
if ($w_Q$ repeats in a question Q' from LQ)
      return Q' if it does not contain a reference
      else return Reference Context Resolution(Q')
if ($w_Q$ is a pronoun)
   CASE ($w_Q \in$ {he,his,she,her,they,their})
     return Q', the closest question from LQ that has
       the expected answer type=PERSON or has a
       PERSON mentioned
   CASE ($w_Q \in$ {it,its})
     if $w_Q$ is the subject of one of
       the verbs {happen, occur} return Q' the first
       question that mentions an event
     return Q', the closest question from LQ that has
```

```
      the expected answer type different than PERSON
      or mentions some non-PERSON entity
  CASE ($w_Q$ =there)
    return Q', the closest question from LQ that has
      the expected answer type=LOCATION or has a
      LOCATION mentioned
  CASE ($w_Q$ =this or $w_Q = \phi$)
    return Q', the first question from LQ
      if it does not contain a reference
      else return Reference Context Resolution(Q')
if ($w_Q$ morphological-root ($w_Q$) =
                     morphological-root ($w_{Q'}$))
  -where $w_{Q'}$ is a word from a question Q'$\in$ LQ
  return Q'
if (there is a WordNet semantic relation
        (e.g. meronymy) between $w_Q$ and $w_{Q'}$)
  -where $w_{Q'}$ is a word from a question Q'$\in$ LQ
  return Q'
```

An interesting by-product of this reference resolution algorithm was the way it allowed the modeling of context through the passing of keywords from the antecedent question to the follow-up question. It is interesting to be noted that each time when a follow-up question would be processed, LCC's QAS^{TM} would operate on the same relevant paragraphs as for the antecedent question in 85% of the cases. However, it would extract different answers, since the expected answer type would be different.

Performance evaluation

Table 7 summarizes the scores provided by NIST for our system. At the time of this writing we did not have the results for the list questions.

	NIST score *lenient*	NIST score *strict*
Main Task	58.7%	57.0%
Context Questions	77.8%	77.0%

Table 7: Accuracy performance

The reading of the results from Table 7 may be misleading - one could conclude that it is easier to process questions in context rather than processing them in isolation. There is a quantitative and a qualitative aspect to this conclusion. First, in TREC-10 there were only 31 questions that were processed in the context of another question whereas in the main task, where questions were processed in isolation, we evaluated close to 500 questions. Second, the first questions in each context were much easier to process then most of the questions from the main task (e.g. definition questions). However, the way context was modeled in LCC's QAS^{TM} was quite felicitous.

Lessons learned

In TREC-10 we learned again that open-domain resources such as WordNet can be fully exploited to process more and more complex definition questions or for processing questions in context. We also learned that such resources are not exhaustive, thus Q/A systems need to robustly process questions even when lexico-semantic information is not available. We also learned that when questions are classified by very broad, practical criteria, e.g. questions asked in isolation vs. questions asked in context, we need to operate changes in the architecture of the Q/A system, using novel ways of solving reference - by customizing its resolution for the Q/A task rather than using methods of resolving the linguistic phonemenon of reference.

For TREC-10 we introduced new modules at the level of question processing to guide the search and extraction of answers based on several forms of bridging inference, mostly determined by lexico-semantic cues. As context questions will probably become more complex, we hope to enhance our bridging inference procedures by relying more on the semantics of follow-up questions.

References

Sanda Harabagiu and Dan Moldovan. Knowledge Processing on Extended WordNet. In *WordNet: An Electronic Lexical Database and Some of its Applications*, editor Fellbaum, C., MIT Press, Cambridge, MA, 1998.

Sanda Harabagiu and Steven Maiorano. Finding answers in large collections of texts: paragraph indexing + abductive inference. *Working Notes of the Fall AAAI Symposium on Question Answering*, November 1999.

Sanda Harabagiu, Marius Paşca and Steven Maiorano. Experiments with Open-Domain Textual Question Answering. In the *Proceedings of the 18th International Conference on Computational Linguistics (COLING-2000)*, pages 292–298, 2000.

Sanda Harabagiu, Dan Moldovan, Marius Pasca, Rada Mihalcea, Mihai Surdeanu, Razvan Bunescu, Roxana Girju, Vasile Rus and Paul Morarescu. FALCON: Boosting Knowledge for Answer Engines. *Proceedings of the Text Retrieval Conference (TREC-9)*, 2000.

Sanda Harabagiu, Dan Moldovan, Marius Pasca, Rada Mihalcea, Mihai Surdeanu, Razvan Bunescu, Roxana Girju, Vasile Rus and Paul Morarescu. The

Role of Lexico-Semantic Feedback in Open-Domain Textual Question-Answering. *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL-2001)*, Toulouse France, pages 274-281, 2001.

G.A. Miller. WordNet: A Lexical Database. *Communication of the ACM*, vol 38: No11, pages 39–41, November 1995.

Dan Moldovan and Rada Mihalcea. A WordNet-based Interface to Internet Search Engines. In *Proceedings of the FLAIRS-98*, pages 275–279, 1998.

Dan Moldovan, Sanda Harabagiu, Marius Paşca, Rada Mihalcea, Richard Goodrum, Roxana Gîrju and Vasile Rus. LASSO - A tool for Surfing the Answer Net. *Proceedings of the Text Retrieval Conference (TREC-8)*, 1999.

Dan Moldovan, Sanda Harabagiu, Marius Paşca, Rada Mihalcea, Richard Goodrum, Roxana Gîrju and Vasile Rus. The Structure and Performance of an Open-Domain Question Answering System. *Proceedings of the 38th Annual Meeting of the Association for Comoutational Linguistics (ACL-2000)*, pages 563–570, 2000.

Marius Paşca and Sanda Harabagiu. High Performance Question/Answering. *Proceedings of the 24th 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-2001)*, New Orleans LA, pages 366-374, 2001.

Marius Paşca and Sanda Harabagiu. The Informative Role of WordNet in Open-Domain Question Answering. *Proceedings of the NAACL 2001 Workshop on WordNet and Other Lexical Resources: Applications, Extensions and Customizations*, Carnegie Mellon University, Pittsburgh PA, pages 138-143, 2001.

Marius Paşca and Sanda Harabagiu. Answer Mining from On-Line Documents. *Proceedings of the ACL-2001 Workshop on Open-Domain Question Answering*, Toulouse France, pages 38-45, 2001.