Extracting Exact Answers using a Meta Question Answering System

Luiz Augusto Sangoi Pizzato and **Diego Mollá-Aliod**
Centre for Language Technology
Macquarie University
2109 Sydney, Australia
{pizzato, diego}@ics.mq.edu.au
http://www.clt.mq.edu.au/

Abstract

This work concerns a question answering tool that uses multiple Web search engines and Web question answering systems to retrieve snippets of text that may contain an exact answer for a natural language question. The method described here treats each Web information retrieval system in a unique manner in order to extract the best results they can provide. The results obtained suggest that our method is comparable with some of today's state-of-the-art systems.

1 Introduction

Text-based Question Answering (QA) focuses on finding answers for natural language questions by searching collections of textual documents. This area of research has become especially active after the introduction of a question answering task in TREC-8 (Voorhees, 1999), which was based on open-domain question answering. The result of this research is a number of systems and QA methodologies not only for generic domains (Moldovan et al., 2003), but also for restricted domains (Mollá et al., 2003) and Web-based systems (Zheng, 2002).

Each type of QA system has specific issues and methodologies. Thus, open-domain QA can rely on generic tools and resources such as parsers, named-entity recognisers, and lexical resources like WordNet (Fellbaum, 1998). This can be seen in recent TREC conferences (Voorhees, 2004b) where some of the participants used readily available third-party resources to quickly build systems that obtained satisfactory results for the amount of effort invested.

On the other hand, restricted domain QA can take advantage of deep knowledge of the covered area by using resources that are specific to the domain such as terminology lists and ontologies, for example in the domain of biomedicine (Zweigenbaum, 2003).

Finally, web-based QA can take advantage of the enormous amount of data available on the World Wide Web and use data-intensive approaches that exploit the inherent redundancy to find answers (Brill et al., 2001). Our system belongs to this category.

The satisfaction of the user with a certain answer will depend on various factors. For instance, someone who wants to find some specific fact would be satisfied with a short and brief answer while someone else may require a more detailed answer. These kind of differences between casual users of generic domain QA systems make the establishment of personalized answer models difficult.

One way that may satisfy both types of users is by providing an exact answer while at the same time showing a snippet of the original text from where the answer was extracted. This kind of response was required from the participant systems of the main task of the QA-track of TREC-2003 (Voorhees, 2004a). We have adopted this approach by providing the exact answer, a summary, and a link to the source document.

According to Voorhees (2003), an exact answer is defined as a string that does not contain any extraneous information but the answer in it. For instance *Brasilia* is the answer for *What is the capital of Brazil?*, but *the city of Brasilia* or *Brazilian capital Brasilia* are not.

In order to find the exact amount of text containing an answer, we used an approach that combines the results of several Web search engines and Web QA systems. Our system works in a similar way of those known as metasearch engines (Metacrawler[1], Mamma[2] and Profusion[3] just to name a few), however we do differentiate between the search engines used

[1] `http://www.metacrawler.com`
[2] `http://www.mamma.com`
[3] `http://www.profusion.com`

Proceedings of the Australasian Language Technology Workshop 2005, pages 105–112,
Sydney, Australia, December 2005.

in order to extract the best information they may provide. Although finding more information for a question helps to retrieve their answers, we believe that the assistance of several search engines can cause improvement when the best information of each one are extracted and weighted.

The common framework for question answering systems consists of three main phases:

Question Analysis: The question is classified into several types, possibly forming a classification hierarchy such as (Moldovan et al., 1999). The question type is typically related to the type of the expected answer, which in turn is typically related to the named-entity types available to the system. The question classification can be based on regular expressions (Mollá-Aliod, 2004; Chen et al., 2002; Hovy et al., 2000) or machine learning (Li and Roth, 2002; Zhang and Lee, 2003). Apart from the question type and expected answer type, this phase may return the question focus and other important words or concepts found in the question.

Information Retrieval: The question and/or the question features obtained by the question analysis are fed to an information retrieval system that returns the documents or document fragments that may contain the answer. Typically a generic document retrieval system is used or even a web search engine, though there are suggestions that the type of information retrieval required for this phase is different from the generic one. This phase is crucial, since relevant documents that fail to be retrieved will be ignored in the next phase.

Answer Extraction: The retrieved documents or passages are analysed in order to find the exact answers. Techniques to find the answer range from the selection of named-entities that are compatible with the expected answer type to the use of logical methods to find and/or validate the answer.

As it can be observed in Figure 1, our system structure is very similar to the common framework, however the approaches for performing each of the tasks are different. The question analysis is performed using the Trie-based question classifier (Pizzato, 2004; Zaanen et al., 2005) trained over the set of near 5500 questions prepared by Li and Roth (2002). As already stated, the information retrieval stage is a combination of several Web search engine results, and the answer extraction combines named-entity, n-grams and lexico-semantic information from WordNet (Fellbaum, 1998).

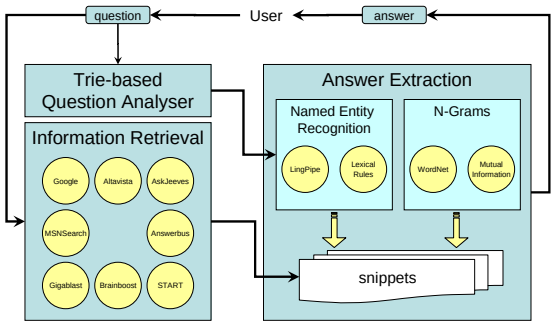

Figure 1: Overview of the system's architecture

In the next section we show some of the characteristics of the Web search engine we explored. Then, in Section 3, we address our method of combining the results and how we used named-entities and n-grams to pinpoint the answer location. In Section 4 we show an evaluation of our technique, while in the last section we present the concluding remarks and future work.

2 Web search results combined

According to Oztekin et al. (2002), the combination of search engine results is not a new approach for improving information retrieval. Many meta Web search engines provide a better retrieval by combining results of several search engines and re-ranking their results according to techniques such as the linear combination of scores described by Vogt and Cottrell (1998). However it seems that most approaches do not consider the differences between search engines. In this work, we take into account the best of each search engine used and, since our goal is to find exact answers to a question, we explored the characteristics of these search engines in order to answer questions.

Because of their availability on the Web, we also used the results of three Web QA systems

(Start[4], Answerbus[5] and Brainboost[6]). These systems perform their jobs using very different approaches and they do not provide exact answers (in the same sense of Voorhees (2003)), but only snippets of text where the answers are expected to be.

As stated we extracted the best of several search engines. Let's list some of their characteristics.

Start: Combines predefined Web databases to provide answers to Geography; Science and Reference; Arts and Entertainment; and History and Culture. The answers are normally structured as tables, sentences or even images and graphics.

Answerbus: Question Answering system that provide answers in a snippet-like format.

Brainboost: Provides answers to natural language questions in a similar way to Answerbus.

Altavista[7]: Well established search engine with a large amount of indexed Web pages.

AskJeeves[8]: It provides very useful information regarding specific questions on famous people, movies, definitions of words, and current weather.

Gigablast[9]: It has the feature, referred to as Gigabits, that presents related concepts to the search results. There is no disclosure on how this information is calculated (we would guess n-grams computation from every search result), but it is possible to notice that the answer for a question is likely to appear in the list of Gigabits.

MSN Search[10]: The Microsoft search engine has the ability to answer some NL questions by using encyclopedia information, as well as providing definitions for words, and a way to make measurement conversion.

Google[11]: It is considered one of the best Web search engines available. It also provides some information on definitions questions like: *What is a platypus?* or *define:platypus*. Following MSN Search, Google has recently acquired the ability to answer encyclopedia questions. We understand that this is a good feature to be used in our system, and we are planning to incorporate this. However the version and results we describe in this paper does not yet consider the QA ability for Google.

The results obtained from the search engine were combined into four different sets:

1. Answers from MSN Search;
2. Answer summaries from Start, Answerbus and Brainboost;
3. Definitions from Google, MSN Search and AskJeeves;
4. Summaries of the results from every Web search/QA system used;

The exact answer was extracted using these sets in a slightly different manner. For instance we observed that, the encyclopedia answers from MSN Search are of a high quality and they are easily pinpointed due to the fixed format and the short size of the passage used. The not-yet incorporated QA feature of Google will fit into this first set when implemented.

The answer snippets from Start, Answerbus and Brainboost do not have the high quality of MSN Search, but they normally contain the right answer within their results.

For definition questions we checked the definition results of Google, MSN Search and AskJeeves. If they are not present and the question asks for a definition, we rephrase the question to the search engine submitting a query in the format *define [question_focus]* or *define:[question_focus]* (on Google) in order to obtain a definition if available.

Because it is not possible to delimit an exact answer in definitions, we state that for these type of questions an exact answer is a brief description of the question subject (focus).

The last set involves all the snippets of documents obtained from the Web search engines. We used the top-50 documents provided by every search engine appended to each other. We did not merge common documents, since we believe that the process of finding the correct answer will take advantage of the several instances of the same information.

[4] http://www.ai.mit.edu/projects/infolab
[5] http://www.answerbus.com
[6] http://www.brainboost.com
[7] http://www.altavista.com
[8] http://www.ask.com
[9] http://www.gigablast.com
[10] http://search.msn.com
[11] http://www.google.com

3 Exact answer extraction

A good approach for answering questions is to provide an exact answer combined with a snippet of text supporting the answer. This may boost the satisfaction of the users of a QA system since the validation of the answers is fast and straightforward. The approach used for pinpointing the exact answer location uses named-entity recognition combined with n-grams extraction and word overlap. We also make use of the semantic classification of terms in WordNet (Fellbaum, 1998).

We first established some priorities in the sets of answers retrieved. If the answer requires a definition, the set of definition answers is evaluated, if this set is empty we try to rephrase the question forcing the search engines to provide a definition if one exists. In case a definition could still not be found, we give up this approach since the information may not be available in a dictionary or even the question analyser may have made a mistake when defining the expected answer category. Giving up the approach means that we will try to find the answer as if the question did not require a definition.

For exact answers, we found that in the rare cases when MSN Search answers questions, they are normally correct. Because of this, we first consider the summary of MSN Search answers if present to extract the exact answer. Otherwise we evaluate the set of answers from the QA systems. If no answer could be found, the set of all search engine responses is analysed.

If still no answer can be found, we relax the expected answer category to all the fine grained categories that the question classification returned. If by this time still no answers are found, the coarse grained categories are used.

3.1 Pinpointing an exact answer

Given the preferences explained above, we define the exact answer by extracting all the named-entities that match the expected answer category provided by our question analyser. The answer categories follow Li and Roth (2002) classification. They are divided into coarse and fine grained categories as shown on Table 1.

We used a large collection of gazetteer files, involving most types of named-entities, along with the LingPipe named-entity recognizer[12] for the definition of persons, organization and location names. In the spirit of Mikheev et al. (1999), we developed a set of internal and external lexical patterns in order to define the remaining types of named-entities.

Table 1: Answer classification and examples

Coarse	Fine	Example
HUM	IND	Who killed JFK?
HUM	GR	What ISPs exist in the NYC?
LOC	CITY	What is the capital of Brazil?
NUM	SPEED	How fast is light?
NUM	MONEY	How much does the President get paid?
DESC	DEF	What is ethology?
ENTY	ANIMAL	What is a female rabbit called?
ENTY	FOOD	What was the first Lifesaver flavor?
ENTY	SUBSTANCE	What is a golf ball made of?
ENTY	DISMED	What is a fear of disease?
ENTY	TERMEQ	What is the name of the Jewish alphabet?

HUM (human)	IND (individual)	GR (group)
LOC (location)	NUM (number)	DESC (description)
DEF (definition)	ENTY (entity)	DISMED (disease)
TERMEQ (equivalent term)		

For every named-entity found we calculate a score according to their average distance from all question words. Consider $F = \{f_1, f_2, \ldots, f_n\}$ to be the sequence of words in the question focus, and $\delta(a, b)$ the distance in words between two terms a and b in the summaries retrieved by the search engines. The score $S(E)$ of a named-entity E is computed as follows:

$$S(E) = \sum_{i=1}^{n} \frac{\delta(E, f_i)^{-1}}{n}$$

The $S(E)$ scoring assumes that possible answers are more likely to be close to the question focus words (Pasca, 2003; Kwok et al., 2001).

Although this provides a measure showing if a named-entity is likely to be the answer in a certain piece of text, we consider that the presence of the same answer string in different passages provides more hints that the answer string is the answer. In order to take advantage of the redundancy the Web provides (Brill et al., 2001), we sum the scores that a named-entity receives for every passage found.

We also have two extra processes that help to improve the answer extraction. The first one uses the Gigabits from Gigablast search engine. If an identified named-entity is in the Gigabits set, the score $S(E)$ is summed to a percentage value given by Gigablast.

The last ranking process uses n-grams information. Unigrams, bigrams and trigrams are extracted from all the responses from the search engines and the mutual information of the n-grams are extracted. The mutual information

[12] http://www.alias-i.com/lingpipe/

$I(a,b)$ is calculated as follows:

$$I(a,b) = \log \frac{P(a,b)}{P(a)P(b)}$$

Observe that $P(a,b)$ is the probability of occurrence of the unigram a followed by unigram b (bigram (a,b)) and $P(x)$ is the probability of occurrence of unigram x.

Since there is no mutual information for unigrams, we calculated a similar measure by the natural logarithm of the product of the probability of finding a certain unigram ($P(u) = freq(u)$/corpus size in unigrams) by the number of different unigrams in the collection of all retrieved summaries.

Those n-grams representing question words and stopwords are discarded and the values from the mutual information that are larger than one[13] are tested on the upper hypernym hierarchy of WordNet. We assumed that if an n-gram is a hyponym of a question word, it may increase the chance that this hyponym is the answer of a question. This helps to answer questions like *Which breed of dog has a blue tongue?*. By using this technique we increase the score of any breed of dog found within the search engine results.

We may still use n-grams that are not in WordNet, however we assign a very low score to them. If the n-gram was also identified as a named-entity, their scores are summed, otherwise it becomes the score for the n-gram alone. The score used by the n-grams is only a fraction of the mutual information calculation. We empirically defined the added value for n-grams found in WordNet hyponyms as a tenth of the mutual information value, while the added value for non-WordNet n-grams was defined as a 20th part.

This seems useful in two distinct cases. First, when the question analysis module fails we are still able to retrieve the correct answer; Second, it gives perspectives on answering multilingual questions. The second case is feasible, however it needs a list of the language stopwords and if possible a lexico-semantic database like WordNet.

After these scoring procedures take place, we collapse and sum scores of different numeric named-entities if their values are the same. For instance 'Two million' is the same as '2 million' or even '2,000,000'. However, though this idea is promising we haven't had the time to implement more information clusters. In the same manner of Kwok at al. (2001), information cluster would help to improve the system precision by grouping similar strings of text. Other information cluster could also include measurement conversions (i.e. 1 km = 1000 meters) and synonyms.

[13] A mutual information value larger than one means that the n-gram occurs more often than its probability of random occurrence.

```
<RANKED_ANSWERS>
 <QUESTION str="What is the capital of Brazil?"/>
  <ANSWER str="Brasilia" id="1" score="3.22">
   <DOC url="http://www.brazzil.com/p35nov95.htm">
    <TITLE>BRAZZIL - News from Brazil - FOOD -
           BRASILIA'S RECIPES
    </TITLE>
    <SUMMARY>
      Brasilia, the capital of Brazil, is better
      known for its prize-winning ultramodern
      design and for the unfriendliness of the
      city to the people who live there
    </SUMMARY>
   </DOC>
   <DOC url="http://gosouthamerica.about.com/b/a/
   065069.htm">
    <TITLE>Brasilia, Capital of Brazil</TITLE>
    <SUMMARY>
       Brasilia, Capital of Brazil. South America
       for Visitors Blog. ... Brasilia, Capital
       of Brazil Brasilia is a monument to what
       Brazilians can do and have done.
    </SUMMARY>
   </DOC>
  </ANSWER>
</RANKED_ANSWERS>
```

Figure 2: System current output

With this final score, every exact answer is ranked and then presented to the user with their source passages. Current results are shown into a XML-like structure containing all the answers and their passages. The idea is to develop a Web interface that will allow users to find answers with a minimum effort and also to provide feedback on the answer quality. An example of the current system output is shown in Figure 2.

The evaluation of the method was performed in a similar way as the main task of the QA track of TREC-2003 (Voorhees, 2004a) as we describe in details in Section 4.

4 Evaluation

The QA track of TREC conferences (Voorhees, 2004a) provides a common environment where QA systems can be tested and evaluated under the same measures. The main task for QA required exact answers from the participant's systems. They were required to provide the answer

and indicate one document in the AQUAINT corpus supporting that answer. The systems' results were manually evaluated from NIST personnel and the answers/systems scores were calculated in a different way for factoid, list and definition questions.

We performed the evaluation of our method only for factoid questions. The system was evaluated using the Mean Reciprocal Rank (MRR) measure of previous TREC QA tracks (Voorhees, 2002). This measure is the average of the precision for every question on it's first correct answer. The MRR is calculated as $\sum_{q=1}^{k} \frac{1}{r_1}$, where r is the ranking of the first correct answer and k is the number of questions.

To obtain the accuracy of our system for fact-based questions we ran the 413 questions of this type from the QA track of TREC-2003 using our system and performed an automatic evaluation using the answer patterns provided by NIST.

As expected these answers did not reflect every answer found in the Internet. Many questions had a correct answer that could not be identified by the patterns. The reasons for this may vary from the lack of answers representations to different or updated answers. For instance our system identified *6000 degrees Celsius* as the answer for *How hot is the sun?*, but the automatic evaluation could only validate answers that follow the patterns of Table 2.

Table 2: TREC-2003 patterns for automatic evaluation of answers.

6500 Celsius
6,?000 degrees Centigrade
two million degrees centigrade

Since the patterns are based on the answers supported by the AQUAINT corpus, some may contain outdated information, for instance *How many NFL teams are there?* requires *31* as its answer according to the patterns, however today's NFL is played with *32* teams.

Because of the reasons listed above, we pre-evaluated our system using the TREC answer pattern in order to adjust some of its parameters, and then we performed a manual evaluation in a slightly different way than the NIST guidelines. Due to time limitations we could not verify all the information sources for unsupported answers. Therefore the answers were assigned only as right or wrong.

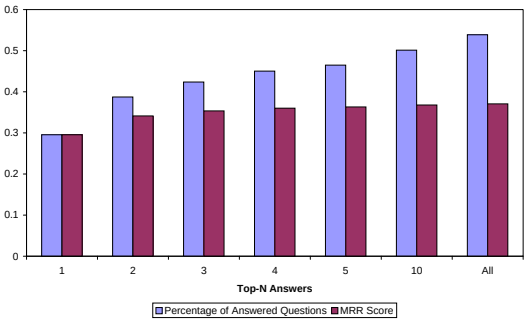

Figure 3: Results for the Top-n answers returned

As observed in Figure 3 using this approach for factoid questions we obtained an accuracy of 30% for the first answer. This result is reasonable considering that it is of the same value of the average results of TREC-2003 systems. This result places our system among the top-5 competitors of the main task for factoid questions.

We should stress that our system does not provides answers using the AQUAINT corpus, nor indicates a document to support the answers in that corpus. We also did not compute the NIL recall and precision since some NIL answers (answer that AQUAINT corpus did not provide an answer) could be found by using the Internet.

We can observe from these results that the exact answer could be found almost half of the times by considering up to 5 answers for every question, giving a reasonable MRR score of 0.36. With these results we may say that the performance obtained by our system could be compared with some of the best systems in TREC.

5 Concluding Remarks

In this work, we develop a meta-QA system that combines the results of different Web search/QA systems in order to provide exact answers for natural language questions. By using a trie-based question analysis, named-entity recognition, n-gram computation and lexico-semantic information from WordNet, we were able to achieve results comparable to some best state-of-the-art QA systems.

Even though our system regards heavily in third-part systems for information retrieval, we showed that it is possible to use and combine the

results from these systems in order to extract exact answers.

Since the developed system is highly modularized, it is possible to remove and add search engines, making the use of those here cited just the first trial for this approach. Further work is needed in order to identify the gain in performance by adding, replacing, removing and promoting search engines. There is also a need for the evaluation of the best weights for the features used to pinpoint the location of the answers, and the feasibility of using language independent methods such as n-grams and mutual information to perform a multilingual QA.

Other interesting aspect of this approach is the capacity of taking advantage of certain features provided by search engines. For instance, by restricting the search domain by Web site, language, country or even neighborhoods, it is possible to restrict the QA domain as well. We already performed some minor tests asking questions in the Macquarie University Web site showing promising results.

We may say that the success of a Web question answering system may not only depend on the precision of its answer. We believe that an effort has to be made in the user interface allowing them to easily verify the answer provided. Further work is needed to be done on developing such a user interface.

References

Eric Brill, Jimmy Lin, Michele Banko, Susan Dumais, and Andrew Ng. 2001. Data-intensive question answering. In Ellen M. Voorhees and Donna K. Harman, editors, *Proc. TREC 2001*, number 500-250 in NIST Special Publication. NIST.

J. Chen, A.R. Diekema, M.D. Taffet, N. McCracken, N. Ercan Ozgencil, O. Yilmazel, and E.D. Liddy. 2002. Question answering: CNLP at the TREC-10 question answering track. In *Proceedings of TREC-2001*, pages 485–494.

C. Fellbaum, editor. 1998. *WordNet: An electronic Lexical Database*. MIT Press.

E. Hovy, L. Gerber, U. Hermjakob, M. Junk, and C. Y. Lin. 2000. Question answering in webclopedia. In *Proceedings of TREC-9*, pages 655–654.

Cody Kwok, Oren Etzioni, and Daniel S. Weld. 2001. Scaling question answering to the web. *ACM Trans. Inf. Syst.*, 19(3):242–262.

X. Li and D. Roth. 2002. Learning question classifiers. In *Proceedings of the COLING-02*, pages 556–562.

Andrei Mikheev, Marc Moens, and Claire Grover. 1999. Named entity recognition without gazetteers. In *Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics*, pages 1–8. Association for Computational Linguistics.

D. Moldovan, S. Harabagiu, M. Paşca, R. Mihalcea, R. Goodrum, Roxana Gîrju, and Vasile Rus. 1999. LASSO: A tool for surfing the answer net. In Voorhees and Harman (Voorhees and Harman, 1999).

Dan Moldovan, Marius Paşca, Sanda Harabagiu, and Mihai Surdeanu. 2003. Performance issues and error analysis in an open-domain question answering system. *ACM Trans. Inf. Syst.*, 21(2):133–154.

D. Mollá-Aliod. 2004. Answerfinder in TREC 2003. In *Proceedings of TREC-2003*.

D. Mollá, F. Rinaldi, R. Schwitter, J. Dowdall, and M. Hess. 2003. Extrans: Extracting answers from technical texts. *IEEE Intelligent Systems*, 18(4):12–17.

B. Uygar Oztekin, George Karypis, and Vipin Kumar. 2002. Expert agreement and content based reranking in a meta search environment using Mearf. In *WWW '02: Proceedings of the eleventh international conference on World Wide Web*, pages 333–344. ACM Press.

Marius Pasca. 2003. *Open-Domain Question Answering from Large Text Collections*. CSLI Publications, Stanford California, USA.

Luiz Augusto Sangoi Pizzato. 2004. Using a trie-based structure for question analysis. In Ash Asudeh, Cécile Paris, and Stephen Wan, editors, *Proceedings of the Australasian Language Technology Workshop 2004*, pages 25–31, Macquarie University, Sydney, Australia, December. ASSTA. ISBN: 0 9581946 1 0.

Christopher C. Vogt and Garrison W. Cottrell. 1998. Fusion via a linear combination of scores. *Information Retrieval*, 1(3):151–173, October.

Ellen M. Voorhees and Donna K. Harman, editors. 1999. *The Eighth Text REtrieval Conference (TREC-8)*, number 500-246 in NIST Special Publication. NIST.

Ellen M. Voorhees. 1999. The TREC-8 question answering track report. In Voorhees and Harman (Voorhees and Harman, 1999).

Ellen M. Voorhees. 2002. Overview of the

TREC 2001 question answering track. In *Proceedings of The Tenth Text REtrieval Conference (TREC 2001)*.

Ellen M. Voorhees. 2003. Overview of the TREC 2002 question answering track. In *Proceedings of TREC-2002*.

Ellen M. Voorhees. 2004a. Overview of the TREC 2003 question answering track. In *Proceedings of TREC-2003*.

Ellen M. Voorhees. 2004b. Overview of TREC 2003. In *Proceedings of TREC-2003*.

Menno Van Zaanen, Luiz Augusto Pizzato, and Diego Molla. 2005. Question classification by structure induction. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI-2005).*, Edinburgh, Scotland, August.

Dell Zhang and See Sun Lee. 2003. Question classification using support vector machines. In *Proc. SIGIR 03*. ACM.

Zhiping Zheng. 2002. Answerbus question answering system. In *Human Language Technology Conference (HLT 2002)*, San Diego, CA, March 24-27.

Pierre Zweigenbaum. 2003. Question answering in biomedicine. In *Proc. EACL2003, workshop on NLP for Question Answering*, Budapest.