

# A Review Paper: Question and Answering System

Himanshu Goyal  
AITEM  
AUUP, Noida  
Uttar Pradesh

Atul Kumar Srivastava  
AITEM  
AUUP, Noida  
Uttar Pradesh

## ABSTRACT

It is examined that the power of ontology based for open and closed domain question and its answering systems in this paper. In the order of obtaining an optimal database for this system, it has studied the method for linking the different phrases of different web links. The tagged corpus is built from an Internet in the bootstrapping process by providing some of the hand-crafted examples of each question and their types. And then the patterns are automatically extracted from the returned documents and formatted answers are provided according to the entities and keywords provided. The precision of each entity has been calculated, and the each question type's average. This ontology is then applied to find the answers of new questions which are about to ask.

## KEYWORDS

Anchor, Corpus, database, multilingual, Ontology and Pattern

## 1. INTRODUCTION

Question answering is the special task of finding natural language answers to the natural language questions [1]. Such systems have become one of the active topics in natural language processing over the past few years. Its popularity stems from the fact that a user receives an exact answer to his questions rather than being overwhelmed with a large number of retrieved documents, which he must then sort through to find the desired answer.

Need of the automatic Question answering system becomes more urgent, as the user struggle to navigate the wealth of on-line information now available [2]. We need a system that allows the user to ask the questions in the everyday language and to receive an answer quickly and succinctly, with sufficient context to validate the answer. Current search engines can return ranked lists of documents, but they do not deliver answers to the user.

Question answering systems address this problem [2]. Recent successes have been reported in a series of question-answering evaluations that started in 1999 as a part of the Text Retrieval Conference (TREC). The best systems are now able to answer more than two thirds of factual questions in this evaluation. The combination of user demand and promising results have stimulated international interest and activity in question answering. This special issue arises from an invitation to the research community to discuss the performance, requirements,

uses, and challenges of question answering systems.

To answer a question, Q/A system must analyse the question firstly, perhaps in context of some current occurring interaction; it must be able to find more than one answers by consulting the on-line resourced database; and it must presents the answers to the user in some appropriate form, perhaps associated with some sort of justification or some of the supporting materials regarding the topic.

Large amounts of data in many disciplines are continuously being added to semantic repositories as a result of continuing

research in different scientific fields, and it is becoming an increasing challenge for researchers to use these repositories efficiently and at the same time cope with this fast pace of the introduction of new knowledge. For example, the National Library of Medicine's MeSH (Medical Subject Heading) vocabulary is used for annotation of scientific literature. Efforts in industry as well as those by scientific communities (e.g., Open Biological Ontologies), which lists well over eighty ontologies) have demonstrated capabilities for building large populated ontologies[3]. Additionally, extraction and the annotation of metadata in the web pages have been addressed earlier and proven to be scalable. Although, publishers of such ontologies try to keep up with the pace of that knowledge expansion, it will be difficult for these semantic repositories to always contain the up-to-date knowledge that exists, for example, in published journal articles or online repositories before these ontologies get updated with the new knowledge.

It has been noted in the several QA systems that certain types of answer are expressed using characteristic phrases. For example, for BIRTHDATEs (with questions like "When was X born?"), typical answers are

"Mozart was born in 1756."  
"Gandhi (1869–1948)..."

These examples suggest that phrases like

"<NAME> was born in <BIRTHDATE>"  
"<NAME> (<BIRTHDATE>..."

When formulated as regular expressions, then it can be used for locating the correct answer[4].

In this paper an approach has been presented. Our method uses the machine learning technique of bootstrapping to build a large tagged corpus starting with only a few examples of QA pairs. Similar techniques have been investigated extensively. These techniques are greatly aided by the fact that there is no need to hand-tag a corpus, while the abundance of data on the web makes it easier to determine reliable statistical estimates.

Our system assumes each of the sentences to be a simple sequence of words and searches for repeated word orderings as evidence for useful answer phrases. The suffix trees for extracting substrings of optimal lengths are used.

## 2. FINDING ANSWERS

Using the patterns to answer a new question following algorithm is been employed:

1. Determine the type of question of the new question.
2. The question term in the question is identified, also using our existing system.
3. Create a query from the question term and perform IR (by using a given answer document corpus collection or web search otherwise) [4].
4. Then segment the document obtained into a sentences and smooth out white space variations and html and other tags, as before.
5. Replace the question term in each sentence by the question tag.

6. Use the pattern table search for presence of the each pattern.
7. Select words matching the tag “<ANSWER>” as the answer.
8. Sort-out these answers by their pattern’s precision scores. Discard duplicates (by elementary string comparisons)

### 3. PATTERN SET ACCURACY

**Table 1: Evaluation of the individual pattern sets.**

Pattern Sets	% Correc t	% Wrong	% Unanswe r
What is abbreviation for X?	78	7	15
When was X born?	60	8	32
What is the capital of X?	29	61	10
What country is X the capital of?	55	40	5
When did X die?	54	1	45
What does X stands for?	21	30	49
<b>Average</b>	<b>49.5%</b>	<b>24.5%</b>	<b>26.0%</b>

Each analysed pattern set was evaluated over one hundred unseen examples with relevant documents being downloaded from the Internet via Google, Yahoo, Bing and many more. The results of this evaluation, which can be seen in Table 1, show that although some of the pattern sets perform well returning very few wrong answers others, such as those for answering “What does X stand for?”, perform poorly.

One of the reasons for the poor performance of some pattern sets is the number of very general rules which are acquired by this method. For instance the pattern set acquired for questions of the form “What is the capital of X?” include the very general pattern: of AnCHoR, AnSWeR. Clearly such a pattern will match against a large number of sentences, not all of which will extract a country or a state or even a more general location instance. This will be illustrated by the fact that the precision of this pattern is only 18%. While one approach would be to insist that the text extracted as answer had the correct semantic type this is not always feasible. For instance what is the semantic type of an abbreviation? How would a system know that NASA is an abbreviation and not a normal word? It may be that using more example question-answer pairs during the learning process would produce more high quality patterns which would be used first when answering questions.

### 4. COMPARISONS OF WEB BASED QA SYSTEMS

From a user’s point of view the Power Answer system is similar to Answer Finder, in the fact that the full question is given to the system and then answers are displayed. The difference is that the answers are very much what you would expect from a search engine in that each answer is a sentence and no attempt is made to cluster (or remove) sentences which contain the same answer. This means that the user still has to read the sentences to locate the answer to their question. This is strangely given that fact that underlying technology have shown to be highly accurate (approximately 85%) even when returning only to a single exact answer.

The system is most comparable with the Answer Finder, from a user’s perspective, as it accepts unstructured natural language questions and returns exact answers and supporting snippets is IONAUT. IONAUT uses its own crawler to index the web with specific focus on entities and the relationships between them in order to provide a richer base for answering questions than the unstructured documents returned by standard search engines. The system returns both exact answers and snippets. Unfortunately the exact answers are not tied to a specific snippet, so it is not immediately clear which snippet supports which answer [5].

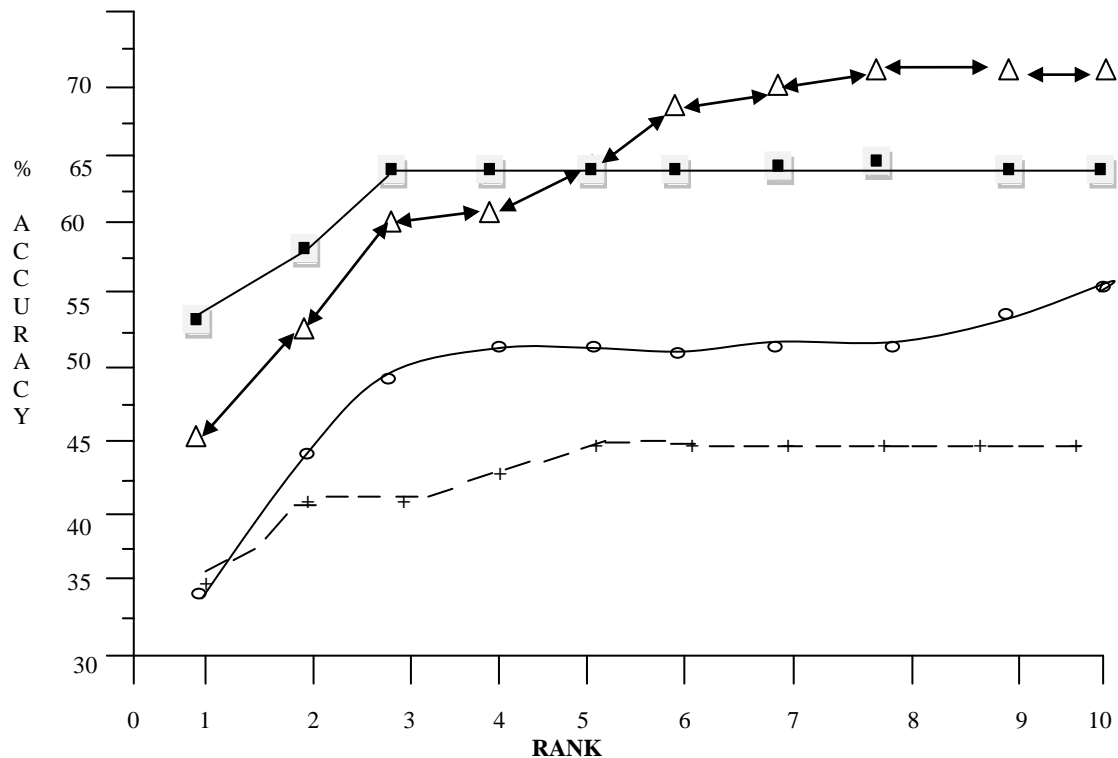
It is believed that Answer Finder, by supplying both exact answers and supporting snippets, is closer to what users would expect of question answering systems than the other web based QA systems that have evaluated – although the actual performance of some of the systems (notably Power Answer) far outstrip that of Answer Finder over the TREC test sets[2].

In a brief experiment to determine the relative performance of the available online QA systems fiftyth questions had been put and used to evaluate web search engines to Answer Bus, Answer Finder, IONAUT and PowerAnswer[5].

Firstly it should be made clear that this comparison is a little unfair, as two of the systems, Answer Finder and IONAUT, return exact answers with supporting snippets while Answer Bus and Power Answer perform the easier task of returning answer bearing sentences. Given this, however, it should be clear from the results in Figure 1 that Answer Finder performs exceptionally well over this small set of questions outperforming all but Power Answer which is only providing relevant sentences. On the other hand this suggests that Answer Bus is actually performing quite badly as it is performing poorly in comparison to Answer Finder (and Power Answer) and is also only attempting the easier task of retrieving relevant sentences.

It should be noted that due to the small number of test questions it is difficult to draw firm conclusions from these experiments. The results are encouraging; however, given that only 22% of the questions were correctly answered at rank one, hence both

Answer Finder and Power Answer have a performance well above average.



**Figure 1: Comparison of Answer Bus O, Answer Finder ■, IONAUT +, and Power Answer △.**

## 5. SHORTCOMING AND EXTENSIONS

No external knowledge need to add to these patterns. The need for matching part of speech and/or semantic types is frequently observed, however. For example, the question: “Where are the Rocky Mountains located?” is answered by “Denver’s new airport, topped with white fibreglass cones in imitation of the Rocky Mountains in the background, continues to lie empty”, because the system picked the answer “the background” using the pattern “the <NAME> in <ANSWER>,”. Using a named entity tagger and ontology would enable the system to use the knowledge that “background” is not a location [4]. DEFINITION questions pose a related problem. Frequently the system’s patterns match a term that is too general, though correct technically. For “what is sonar?” the pattern “<NAME> and related <ANSWER>s” matches “while its sonar and related undersea systems are built” [6].

The patterns cannot handle long-distance dependencies. For example, for “Where is London?” the system cannot locate the answer in “London, which has one of the most busiest airports in the world, lies on the banks of the river Thames” due to the explosive danger of unrestricted wildcard matching, as would be required in the pattern “<QUESTION>, (<any word>)\*, lies on <ANSWER>”. This is one of the reasons why the system performs very well on certain types of questions from the web but performs poorly with documents obtained from the corpus. The abundance and variation of data on the Internet allows the system to find an instance of its patterns without losing answers to long term dependencies. The corpus, on the other hand, typically contains fewer candidate answers for a given question and many of the answers present may match only long-term dependency patterns [6]. More information needs to be added to the text patterns regarding the length of the answer phrase to be expected. The system searches in the range of 50 bytes of the answer phrase to capture the pattern. It fails to perform under certain conditions as exemplified by the question “When was Lyndon B. Johnson born?”. The system selects the sentence “Tower gained national attention in 1960 when he lost to

democratic Sen. Lyndon B. Johnson, who ran for both re-election and the vice presidency” using the pattern “<NAME> <ANSWER> –“. The system lacks the information that the <ANSWER> tag should be replaced exactly by one word [6]. A more serious limitation is that the present framework can handle only one anchor point (the question term) in the candidate answer sentence. It cannot work for types of question that require multiple words from the question to be in the answer sentence, possibly apart from each other. For example, in “Which county does the city of Long Beach lie?”, the answer “Long Beach is situated in Los Angeles County” requires the pattern. “<QUESTION\_TERM\_1> situated in <ANSWER> <QUESTION\_TERM\_2>”, where <QUESTION\_TERM\_1> and <QUESTION\_TERM\_2> represent the terms “Long Beach” and “county” respectively. The presence of multiple anchor words would help to eliminate many of the candidate answers by simply using the condition that all the anchor words from the question must be present in the candidate answer sentence [7]. The system does not classify or make any distinction between upper and lower case letters. For example, “What is micron?” is answered by “In Boise, Idaho, a spokesman for Micron, a maker of semiconductors, said Simms is ‘a very high volume product for us’ ”. The answer returned by the system would have been perfect if the word “micron” had been capitalized in the question [7]. Canonicalization of words is also an issue. Say, for BIRTHDATE questions, (for example, Gandhi’s birth date can be written as “1869”, “Oct. 2, 1869”, “2<sup>nd</sup> October 1869”, “October 2, 1869”, and so on). The date tagger could used to cluster all of the variations instead of enlisting all the possibilities and it also tags them with the same term. For smoothing out to the variations in the question terms for names of persons, the same idea could be extended here (Gandhi could be written as “Mahatma Gandhi”, “Mohandas Karamchand Gandhi”, etc.)[7].

## 6. CONCLUSIONS

Since the output from the Web contains many correct answers among the top ones, a simple word count could help in eliminating many unlikely answers. This would work well for question types like BIRTHDATE or LOCATION but is not clear for question types like DEFINITION.

The simplicity of this method makes it perfect for multilingual QA. Many tools required by sophisticated QA systems (name entity taggers, parsers, ontologies, etc.) are language specific and require significant effort to adapt to a new language. Since the answer patterns used in this method are learned using only a small number of manual training terms, one can rapidly learn patterns for new languages, assuming the web search engine is appropriately switched.

## 7. ACKNOWLEDGMENTS

Our sincere thanks to the Mr. Deepak and Mr. Eduards without whose help this paper was hard to make.

## 8. REFERENCES

- [1] Proceedings of the 9th Text Retrieval Conference (TREC 9), NIST, 479–488. Richard J Cooper and Stefan M R., “A Simple Question Answering System”, News article from TIPSTER and TREC CD’s, (Pg 1-7).
- [2] Robert F. Simons, “Natural Language Question- Answering Systems: 1969”, Communication of the ACM, Volume 13, January 1970, (Pg 15-30).
- [3] David Ferrucci, Eric Nyberg, James Allan, Ken Barker, “Towards the Open Advancement of Question Answering Systems”, IBM Research Report, RC24789(W0904-093) April 22,2009, (Pg 1-28).
- [4] Harabagiu, S., D. Moldovan, M. Pasca, R. Mihalcea, M. Surdeanu, R. Buneascu, R. Gîrju, V. Rus and P. Morarescu. 2001. FALCON: Boosting Knowledge for Answer Engines.
- [5] Lotfi A. Zadeh,” From Search Engines to Question Answering Systems – The Problems of World Knowledge, Relevance, Deduction and Precisation”, FUZZY LOGIC AND THE SEMANTIC WEB, 2005, (Pg 163-210).
- [6] Voorhees, E. 2001. Overview of the Question Answering Track. Proceedings of the TREC-10 Conference. NIST, Gaithersburg, MD, 157–165.
- [7] Hovy, E.H., U. Hermjakob, C.-Y. Lin, and D. Ravichandran. 2002b. Using Knowledge to Facilitate Pinpointing of Factoid Answers. Proceedings of the COLING-2002 conference. Taipei, Taiwan.