

# A Quick Introduction to Question Answering

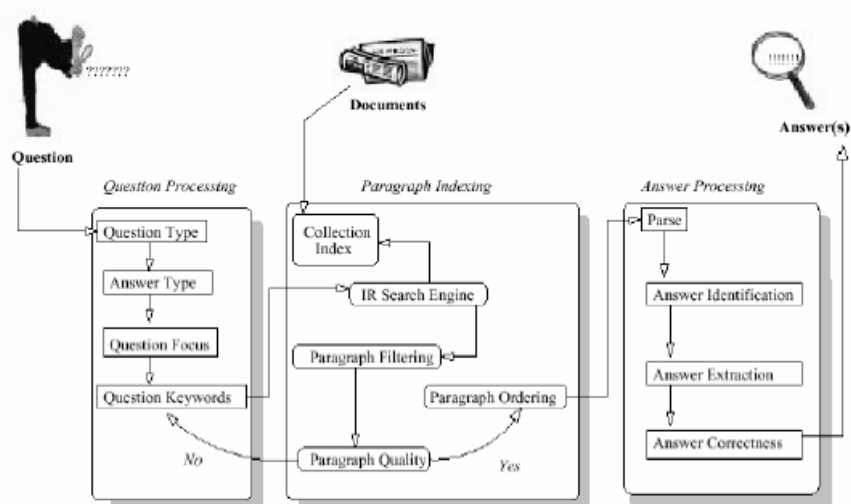
Author: Andrew Lampert (Andrew.Lampert@csiro.au)

Date: December 2004

Question Answering is a specialised form of information retrieval. Given a collection of documents, a Question Answering system attempts to retrieve correct answers to questions posed in natural language.

Open-domain question answering requires question answering systems to be able to answer questions about any conceivable topic. Such systems cannot, therefore, rely on hand crafted domain specific knowledge to find and extract the correct answers.

**Figure 1: Architecture of an Open Domain QA System (from lecture notes for COMP248 subject at Macquarie University)**



Above is shown a possible architecture of an open-domain question answering system. The architecture described below is based in part on one developed by Moldovan et al. [4] for the TREC QA track, but is also inspired with additional design ideas and concepts from a wide range of question answering systems.

## Question Processing Module

Given a natural language question as input, the overall function of the question processing module is to process and analyse the question, and to create some representation of the information requested. Creating this representation requires the question processing module to determine:

- The question type, usually based on a taxonomy of possible questions already coded into the system;
- The expected answer type, through some shallow semantic processing of the question; and
- The question focus, which represents the main information that is required to answer the user's question.

These steps allow the question processing module to finally pass a set of query terms to the Paragraph Indexing module, which uses them to perform the information retrieval.

## ***Question Type Classification***

In order to find a correct answer to a user's question, we need to first know what to look for in our large collection of documents. The type of answer required is related to the form of the question, so knowing the type of a question can provide constraints on what constitutes relevant data, which helps other modules to correctly locate and verify an answer.

The question type classification component is therefore a useful, if not essential component in a QA system, as it provides significant guidance about the nature of the required answer.

While the TREC Question Answering track involves questions about open domain topics, the question types themselves are a fairly closed set. In general, questions conform to predictable language patterns. This makes question type classification somewhat simpler.

Many researchers have proposed various different taxonomies for question classification. Wendy Lehnert, for example, proposed a conceptual taxonomy with 13 conceptual classes [1] back in 1986. More recently, Li and Roth propose a multi-layered taxonomy [2], which has 6 coarse classes (ABBREVIATION, ENTITY, DESCRIPTION, HUMAN, LOCATION and NUMERIC VALUE), and 50 fine classes. Moldovan et al. [4] provide another set of question classes and subclasses along with corresponding answer types, based on the 200 question used in TREC 8. This table is shown in Figure 2.

Ultimately, I would select some classification taxonomy, probably close to that used by Moldovan et al. . Unfortunately, there are also issues with boundaries between question classes, making classification an ambiguous task. In order to resolve some of this ambiguity, I would allow for multiple class labels to be added to a single question. To perform the actual classification, I would use Support Vector Machines (SVM) to classify questions based on feature sets, using either surface text features (e.g., by treating the question as a bag of words or n-grams), or more semantic features (e.g., identifying named entities). SVMs with both surface text features and shallow semantic features have been used quite successfully to classify questions [Zhang and Lee – SIGIR 2003] [Suzuki et al – ACL 2003].

Figure 2: Types of questions, corresponding answer types, and statistics from the set of TREC 8 questions (From Moldovan et al. [4])

Q-class	Q-subclass	Nr.Q	Nr. Q answ.	Answer type	Example of question	Focus
what		64	54			
	basic what	40	34	MONEY/NUMBER/ DEFINITION/TITLE/ NNP/UNDEFINED	What was the monetary value of the Nobel Peace Prize in 1989?	monetary value
	what-who	7	7	PERSON/ ORGANIZATION	What costume designer decided that Michael Jackson should only wear one glove?	costume designer
	what-when	3	2	DATE	In what year did Ireland elect its first woman president?	year
	what-where	14	12	LOCATION	What is the capital of Uruguay?	capital
who		47	37	PERSON/ ORGANIZATION	Who is the author of the book "The Iron Lady: A Biography of Margaret Thatcher"?	author
how		31	21			
	basic how	1	0	MANNER	How did Socrates die?	Socrates
	how-many	18	13	NUMBER	How many people died when the Estonia sank in 1994?	people
	how-long	2	2	TIME/DISTANCE	How long does it take to travel from Tokyo to Niigata?	-
	how-much	3	2	MONEY/PRICE	How much did Mercury spend on advertising in 1993?	Mercury
	how-much- <modifier>	1	0	UNDEFINED	How much stronger is the new vitreous carbon material invented by the Tokyo Institute of Technology compared with the material made from cellulose?	new vitreous carbon material
	how-far	1	1	DISTANCE	How far is Yaroslavl from Moscow?	Yaroslavl
	how-tall	3	3	NUMBER	How tall is Mt. Everest?	Mt. Everest
	how-rich	1	0	UNDEFINED	How rich is Bill Gates?	Bill Gates
	how-large	1	0	NUMBER	How large is the Arctic refuge to preserve unique wildlife and wilderness value on Alaska's north coast?	Arctic refuge
where		22	16	LOCATION	Where is Taj Mahal?	Taj Mahal
when		19	13	DATE	When did the Jurassic Period end?	Jurassic Period
which		10	8			
	which-who	1	1	PERSON	Which former Klu Klux Klan member won an elected office in the U.S.?	Klu Klux Klan member
	which-where	4	3	LOCATION	Which city has the oldest relationship as sister-city with Los Angeles?	city
	which-when	1	1	DATE	In which year was New Zealand excluded from the ANZUS alliance?	year
	which-what	4	3	NNP/ ORGANIZATION	Which Japanese car maker had its biggest percentage of sale in the domestic market?	Japanese car maker
name		4	4			
	name-who	2	2	PERSON/ ORGANIZATION	Name the designer of the show that spawned millions of plastic imitations, known as "jellies"?	designer
	name-where	1	1	LOCATION	Name a country that is developing a magnetic levitation railway system?	country
	name-what	1	1	TITLE/NNP	Name a film that has won the Golden Bear in the Berlin Film Festival?	film
why		2	0	REASON	Why did David Koresh ask for a word processor?	David Koresh
whom		1	0	PERSON/ ORGANIZATION	Whom did the Chicago Bulls beat in the 1993 championship?	Chicago Bulls
<b>Total</b>		200	153 77%			

## Answer Type Classification

Answer type classification could be based on a mapping of the question classification. Once a question has been classified, a simple rule based mapping would be used to determine the potential answer types. Again, because question classification can be ambiguous, the system would need to allow for multiple answer types.

## Question Focus

Unfortunately, knowing the question type alone is not sufficient for finding answers [4] to all questions. In particular, *what* questions can be quite ambiguous in terms of

the information asked by the question. In order to address this ambiguity, an additional component which extracts the question focus is necessary.

The *focus* of a question has been defined by Moldovan et al [4] to be a word or sequence of words which indicate what information is being asked for in the question. As an example, the question “*What is the longest river in New South Wales?*” has the focus “*longest river*”. If both the question type (from the question classification component) and the focus are known, the system is able to more easily determine the type of answer required.

Identifying the focus can be done either using pattern matching rules, based on the question type classification, or using a statistical approach. A statistical approach might again make use of n-grams to identify likely focus words of questions. Such an approach would require a training corpus of questions with known question foci to be developed, which may be prohibitively expensive in terms of time and effort.

### **Question Keywords**

Once determined, the question focus is also used to determine the list of keywords to be passed to the information retrieval component. The process of extracting keywords could be performed with the aid of standard techniques such as named entity recognition, stop-word lists, and part-of-speech taggers, along with a set of ordered heuristics, such as those described in [4]. Based on the work in [4], all words satisfying any of the following 8 heuristics would be chosen as keywords:

1. For each quoted expression in a question, all non-stop words in the quotation.
2. Words recognized as proper nouns (using named-entity recognition).
3. Complex nominals and their adjective modifiers.
4. All other complex nominals
5. All nouns and their adjectival modifiers
6. All other nouns
7. All verbs
8. The question focus

Each heuristic would return a set of keywords that would be added in order to the set of question keywords. Usually, only the first 6 heuristics are used – the final two offer further keywords if required after paragraph indexing (i.e., if there are insufficient candidate paragraphs after extraction and filtering).

Other methods of expanding the set of question keywords could include using an online resource such as WordNet. The synonym sets in WordNet could be used to expand the set of question keywords with semantically related words that might also occur in documents containing the correct question answer.

The set of question keywords is sorted by priority, so if too many keywords are extracted from the question, only the first N words are passed onto the next module. N would be a configurable value that could be tuned, based on an evaluation of performance with different numbers of keywords for information retrieval. The number of keywords passed can also be controlled by the paragraph indexing module,

if the paragraph quality component determines that a different set of question keywords is required.

## **Paragraph Indexing Module**

The Paragraph Indexing module is often also referred to as a Document Processing module in many Question Answering systems. Usually, this module relies on one or several separate information retrieval systems to gather information from a collection of document corpora (which in the case of open domain Question Answering, almost always involves the World Wide Web as at least one of those corpora).

Results from the information retrieval systems (search engines) are generally filtered to remove paragraphs that do not contain all the keywords of the question being answered. This allows for a paragraph index to be generated. After assessing the quality of indexed paragraphs, this module then orders the extracted paragraphs, according to how plausible the contained answer is. If there are too many or too few paragraphs with plausible answers, then new queries can be constructed, either with more or less question keywords, and sent again to the information retrieval system. This ensures that a reasonable number of paragraphs (i.e., not too many, and not too few) are passed on to the Answer Processing module.

The motivation for distilling documents down to paragraphs before processing them in detail is to require less content to be analysed in detail, making for a faster system. The response time of a QA system is very important due to the interactive nature of question answering.

## ***Information Retrieval***

I would choose to have Information retrieval (IR) performed using standard IR technologies and techniques, such as existing web search engines (Google, AltaVista etc.). One thing to be aware of when using these systems is that the fairly standard IR approach of using a cosine vector space model of measuring similarity between documents and queries is not desirable for IR in question answering. This is mainly because a QA system usually wants documents to be retrieved only when all keywords are present in the document. This is because the keywords have been carefully selected by the Question Processing module as being the most representative words in the question. Cosine similarity based IR systems often return documents even if not all keywords are present.

Information retrieval systems are usually evaluated based on two metrics – precision and recall. Precision refers to the ratio of relevant (or correct) documents returned to the total number of documents returned. Recall refers to the number of relevant documents returned out of the total number of relevant documents available in the document collection being searched. In general, the aim for information retrieval systems is to optimise both precision and recall. For question answering, however, the focus is subtly different.

Because a QA system performs post processing on the documents returned, the recall of the IR system is significantly more important than its precision. A QA system filters out irrelevant documents or sections as part of the paragraph filtering component. In doing so, it raises the precision of information, compensating for lower precision in the document set returned by the IR system. Lower recall in the IR systems means the answer is less likely to be in the returned documents. Where the answer is not present, the QA system must first recognise this (which is difficult and error prone), and then re-select question keywords to send to the information retrieval engine(s). If rephrasing the query still does not cause the IR system to include the correct answer in its output, even the best QA system is unable to successfully answer the posed question.

Rather than attempting to construct my own document collection and index, I would use one or several web search engines, such as Google. Search engines such as Google already employ some common techniques of increasing recall such as stemming. And besides, the recall of a search engine that has indexed more than 8 billion documents is hard to overlook! To ensure that only documents containing all keywords are returned, Boolean 'AND' operators can be placed between each question keyword.

In addition to Google, other more specialised resources could also be used. These resources could be linked to specific types of questions and keywords. Examples of specialised information retrieval services include databases like Amazon.com for books, IMDB for movies, and the CIA World Fact Book. Such specialised resources are likely to have much higher precision and recall than open-domain web search engines for questions in their specific domain. Of course, this higher recall and precision is over a much smaller corpus, but it is likely that the bulk of these specialised resources are not included in the general web search corpora. This stems from the fact that many of these specialised databases are part of the so-called 'dark web', which is yet to be indexed by current state-of-the-art web search engines.

The other advantage of making use of separate knowledge sources are that they offer the opportunity for sanity checking candidate answers. More on answer checking will be discussed when we look in detail at the Answer Processing module.

Other approaches to improve information retrieval include predictive annotation of documents. Predictive annotation allows a document to be indexed or marked-up with concepts or features that are expected to be useful in answering a reasonable number of questions. While this makes the indexing process more computationally expensive and can make the index size an order of magnitude larger, indexing of document collections is performed offline, so from a QA point of view, it is a worthwhile trade-off.

As already noted in the description of the question processing module, it is often possible to determine the answer type required, based on analysis of the question. The concepts that could be marked during predictive annotation could therefore use the same taxonomy of concepts as is used for answer type classification. This would allow the QA system to directly and easily exploit its answer type classification to constrain the information retrieval search to content that matches not only the

keywords, but keywords of the specified type (eg. Bush as a <personName> rather than any of its other senses).

## ***Paragraph Filtering***

As already hinted at, the number of documents returned by the information retrieval system may be very large. Paragraph filtering can be used to reduce the number of candidate documents, and to reduce the amount of candidate text from each document.

The concept of paragraph filtering is based on the principle that the most relevant documents should contain the question keywords in a few neighbouring paragraphs, rather than dispersed over the entire document. To exploit this idea, the location of the set of question keywords in each document is examined. If the keywords are all found in some set of N consecutive paragraphs, then that set of paragraphs will be returned, otherwise, the document is discarded from further processing. N is again a configurable number that could be tuned based on an evaluation of system performance under different tolerances of keyword distance in documents.

## ***Paragraph Quality***

The paragraph quality component is responsible for evaluating the quality of the selected paragraphs. If the quality of paragraphs is deemed to be insufficient, then the system returns to the question keyword extraction module, and alters the heuristics for extracting keywords from the question. This new set of keywords is then used to perform the information retrieval from scratch.

Most commonly, the cause of re-determining question keywords stems from having either too many or too few candidate paragraphs after paragraph filtering. In either case, new queries for the information retrieval system are created by revisiting the question keywords component, and either adding or dropping keywords. This feedback loop provides some form of retrieval context that ensures that only a 'reasonable' number of paragraphs are passed onto the Answer Processing module. Like many other parameters, exactly how many paragraphs constitute a 'reasonable' number should be configured, based on performance testing.

## ***Paragraph Ordering***

The aim of paragraph ordering is to rank the paragraphs according to a plausibility degree of containing the correct answer.

Paragraph ordering is performed using a standard radix sort algorithm. The radix sort uses three different scores to order paragraphs:

- The number of words from the question that are recognized in the same sequence within the current paragraph window;

- The number of words that separate the most distant keywords in the current paragraph window; and
- The number of unmatched keywords in the current paragraph window.

A paragraph window is defined as the minimal span of text required to capture each maximally inclusive set of question keywords within each paragraph. Radix sorting is performed for each paragraph window across all paragraphs.

Other possible heuristics include weighting the confidence of each paragraph according to its source. One might choose, for example, to give greater weight to local or well known data sources than unknown sources. Similarly, government organisations (identified by the namespace of the source domain name) might be ranked with greater confidence than other sources.

## **Answer Processing**

As the final module in the architecture pipeline, the Answer Processing module is responsible for identifying and extracting answers from the set of ordered paragraphs passed to it from the paragraph indexing module.

### ***Answer Identification***

The answer type (hopefully) determined during question processing, is crucial to guiding this process.

In order to identify paragraphs which contain the required answer type, shallow parsing techniques such as named entity recognition are commonly used. As mentioned previously, if predictive annotation of the document corpus has been performed, such conceptual mark-up can also help the answer identification module.

The use of a part-of-speech tagger (e.g., Brill tagger) can help to enable recognition of answer candidates within identified paragraphs. Answer candidates can be ranked based on measures of distance between keywords, numbers of keywords matched and other similar heuristic metrics.

Commonly, if no match is found, QA systems would fallback to delivering the best ranked paragraph. Unfortunately, given the tightening requirements of the TREC QA track, such behaviour is no longer useful.

### ***Answer Extraction***

Once an answer has been identified, the shallow parsing performed is leveraged to extract only the relevant word or phrase in answer to the question.

## **Answer Correctness**

Confidence in the correctness of an answer can be increased in a number of ways. One way is to use a lexical resource like WordNet to verify that a candidate response was of the correct answer type.

As mentioned earlier, specific knowledge sources can also be used as a 'second opinion' to check answers to questions within specific domains. This allows candidate answers to be sanity checked before being presented to a user.

If a specific knowledge source has been used to actually retrieve the answer, then general web search can also be used to sanity check answers. The principle relied on here is that the number of documents that can be retrieved from the web in which the question and the answer co-occur can be considered a significant clue of the validity of the answer. Several people have investigated using the redundancy of the web to validate answers based on frequency counts of question answer collocation, and found it to be surprisingly effective [3]. Given its simplicity, this makes it an attractive technique.

## **Answer Presentation**

In the original TREC QA tracks, systems could present a list of several answers, and were ranked based on where the correct answer appeared in the list. From 1999-2001, the length of this list was 5. Since 2002, systems have been required to present only a single answer. So, my system would be designed to select and present only the most likely answer.

## **References**

1. Lehnert, W., (1986). "A conceptual theory of question answering". In B. J. Grosz, K. Sparck Jones, and B. L. Webber, editors, *Natural Language Processing*, pages 651–657. Kaufmann, Los Altos, CA.
2. Li, X. and Roth, D. (2002), "Learning Question Classifiers", In *Proceedings of COLING 2002*.
3. Magnini, B., Negri, M., Prevete, R. and Tanev, H., (2002) "Is It the Right Answer? Exploiting Web Redundancy for Answer Validation". In *Proceedings of the 40<sup>th</sup> Annual Meeting of the Association for Computational Linguistics (ACL-2002)*, Philadelphia, PA
4. Moldovan, D., Harabagiu, S., Pasca, M., Mihalcea, R., Girju, R, Goodrum, R, Rus, V., "The Structure and Performance of an Open-Domain Question Answering System", (2000), In *Proceedings of the Conference of the Association for Computational Linguistics (ACL-2000)*, p 563-570.