Answer Extraction

Steven Abney Michael Collins Amit Singhal
AT&T Shannon Laboratory
180 Park Ave.
Florham Park, NJ 07932
{abney,mcollins,singhal}@research.att.com

Abstract

Information retrieval systems have typically concentrated on retrieving a set of *documents* which are relevant to a user's query. This paper describes a system that attempts to retrieve a much smaller section of text, namely, a direct *answer* to a user's question. The SMART IR system is used to extract a ranked set of passages that are relevant to the query. Entities are extracted from these passages as potential answers to the question, and ranked for plausibility according to how well their type matches the query, and according to their frequency and position in the passages. The system was evaluated at the TREC-8 question answering track: we give results and error analysis on these queries.

1 Introduction

In this paper, we describe and evaluate a question-answering system based on passage retrieval and entity-extraction technology.

There has long been a concensus in the Information Retrieval (IR) community that natural language processing has little to offer for retrieval systems. Plausibly, this is creditable to the preeminence of ad hoc document retrieval as the task of interest in IR. However, there is a growing recognition of the limitations of ad hoc retrieval, both in the sense that current systems have reached the limit of achievable performance, and in the sense that users' information needs are often not well characterized by document retrieval.

In many cases, a user has a question with a specific answer, such as *What city is it where the European Parliament meets?* or *Who discovered Pluto?* In such cases, ranked answers with links to supporting documentation are much more useful than the ranked list of documents that standard retrieval engines produce.

The ability to answer specific questions also provides a foundation for addressing quantitative inquiries such as *How many times has the Fed raised interest rates this year?* which can be interpreted as the cardinality of the set of answers to a specific question that happens to have multiple correct answers, like *On what date did the Fed raise interest rates this year?*

We describe a system that extracts specific answers from a document collection. The system's performance was evaluated in the question-answering track that has been introduced this year at the TREC information-retrieval conference. The major points of interest are the following.

- Comparison of the system's performance to a system that uses the same passage retrieval component, but no natural language processing, shows that NLP provides significant performance improvements on the question-answering task.
- The system is designed to build on the strengths of both IR and NLP technologies. This makes for much more robustness than a pure NLP system would have, while affording much greater precision than a pure IR system would have.
- The task is broken into subtasks that admit of independent development and evaluation. Passage retrieval and entity extraction are both recognized independent tasks. Other subtasks are entity classification and query classification—both being classification tasks that use features obtained by parsing—and entity ranking.

In the following section, we describe the question-answering system, and in section 3, we quantify its performance and give an error analysis.

2 The Question-Answering System

The system takes a natural-language query as input and produces a list of answers ranked in order of confidence. The top five answers were submitted to the TREC evaluation.

Queries are processed in two stages. In the information retrieval stage, the most promising passages of the most promising documents are retrieved. In the linguistic processing stage, potential answers are extracted from these passages and ranked.

The system can be divided into five main components. The information retrieval stage consists of a single component, passage retrieval, and the linguistic processing stage circumscribes four components: entity extraction, entity classification, query classification, and entity ranking.

Passage Retrieval Identify relevant documents, and within relevant documents, identify the passages most likely to contain the answer to the question.

Entity Extraction Extract a candidate set of possible answers from the passages.

Entity Classification The candidate set is a list of entities falling into a number of categories, including people, locations, organizations, quantities, dates, and linear measures. In some cases (dates, quantities, linear measures), entity classification is a side effect of entity extraction, but in other cases (proper nouns, which may be people, locations, or organizations), there is a separate classification step after extraction.

Query Classification Determine what category of entity the question is asking for. For example, if the query is

> Who is the author of the book, The Iron Lady: A Biography of Margaret Thatcher?

the answer should be an entity of type `Person`.

Entity Ranking Assign scores to entities, representing roughly belief that the entity is the correct answer. There are two components of the score. The most-significant bit is whether or not the category of the entity (as determined by entity classification) matches the category that the question is seeking (as determined by query classification). A finer-grained ranking is imposed on entities with the correct category, through the use of frequency and other information.

The following sections describe these five components in detail.

2.1 Passage Retrieval

The first step is to find passages likely to contain the answer to the query. We use a modified version of the SMART information retrieval system (Buckley and Lewit, 1985; Salton, 1971) to recover a set of documents which are relevant to the question. We define passages as overlapping sets consisting of a sentence and its two immediate neighbors. (Passages are in one-one correspondence with with sentences, and adjacent passages have two sentences in common.) The score for passage i was calculated as

$$\tfrac{1}{4}S_{i-1} + \tfrac{1}{2}S_i + \tfrac{1}{4}S_{i+1} \qquad (1)$$

where S_j, the score for sentence j, is the sum of IDF weights of non-stop terms that it shares with the query, plus an additional bonus for pairs of words (bigrams) that the sentence and query have in common.

The top 50 passages are passed on as input to linguistic processing.

2.2 Entity Extraction

Entity extraction is done using the Cass partial parser (Abney, 1996). From the Cass output, we take dates, durations, linear measures, and quantities.

In addition, we constructed specialized code for extracting proper names. The proper-name extractor essentially classifies capitalized words as intrinsically capitalized or not, where the alternatives to intrinsic capitalization are sentence-initial capitalization or capitalization in titles and headings. The extractor uses various heuristics, including whether the words under consideration appear unambiguously capitalized elsewhere in the document.

2.3 Entity Classification

The following types of entities were extracted as potential answers to queries.

Person, Location, Organization, Other Proper names were classified into these categories using a classifier built using the method described in (Collins and Singer, 1999).[1] This is the only place where entity classification was actually done as a separate step from entity extraction.

Dates Four-digit numbers starting with `1...` or `20..` were taken to be years. Cass was used to extract more complex date expressions (such as *Saturday, January 1st, 2000*).

Quantities Quantities include bare numbers and numeric expressions like *The Three Stooges*, *4 1/2 quarts*, *27%*. The head word of complex numeric expressions was identified (*stooges*, *quarts* or *percent*); these entities could then be later identified as good answers to *How many* questions such as *How many stooges were there?*

Durations, Linear Measures Durations and linear measures are essentially special cases of quantities, in which the head word is a time unit or a unit of linear measure. Examples of durations are *three years*, *6 1/2 hours*. Examples of linear measures are *140 million miles*, *about 12 feet*.

We should note that this list does not exhaust the space of useful categories. Monetary amounts (e.g.,

[1] The classifier makes a three way distinction between `Person`, `Location` and `Organization`; names where the classifier makes no decision were classified as `Other Named Entity`.

$25 million) were added to the system shortly after the Trec run, but other gaps in coverage remain. We discuss this further in section 3.

2.4 Query Classification

This step involves processing the query to identify the category of answer the user is seeking. We parse the query, then use the following rules to determine the category of the desired answer:

- *Who, Whom* → `Person`.
- *Where, Whence, Whither* → `Location`.
- *When* → `Date`.
- *How few, great, little, many, much* → `Quantity`. We also extract the head word of the *How* expression (e.g., *stooges* in *how many stooges*) for later comparison to the head word of candidate answers.
- *How long* → `Duration` or `Linear Measure`. *How tall, wide, high, big, far* → `Linear Measure`.
- The wh-words *Which* or *What* typically appear with a head noun that describes the category of entity involved. These questions fall into two formats: *What X* where *X* is the noun involved, and *What is the ... X*. Here are a couple of examples:

  What company is the largest Japanese ship builder?

  What is the largest city in Germany?

  For these queries the head noun (e.g., *company* or *city*) is extracted, and a lexicon mapping nouns to categories is used to identify the category of the query. The lexicon was partly hand-built (including some common cases such as *number* → `Quantity` or *year* → `Date`). A large list of nouns indicating `Person`, `Location` or `Organization` categories was automatically taken from the contextual (appositive) cues learned in the named entity classifier described in (Collins and Singer, 1999).
- In queries containing no wh-word (e.g., *Name the largest city in Germany*), the first noun phrase that is an immediate constituent of the matrix sentence is extracted, and its head is used to determine query category, as for *What X* questions.
- Otherwise, the category is the wildcard `Any`.

2.5 Entity Ranking

Entity scores have two components. The first, most-significant, component is whether or not the entity's category matches the query's category. (If the query category is `Any`, all entities match it.)

In most cases, the matching is boolean: either an entity has the correct category or not. However, there are a couple of special cases where finer distinctions are made. If a question is of the `Date` type, and the query contains one of the words *day* or *month*, then "full" dates are ranked above years. Conversely, if the query contains the word *year*, then years are ranked above full dates. In *How many X* questions (where *X* is a noun), quantified phrases whose head noun is also *X* are ranked above bare numbers or other quantified phrases: for example, in the query *How many lives were lost in the Lockerbie air crash*, entities such as *270 lives* or *almost 300 lives* would be ranked above entities such as *200 pumpkins* or *150*.[2]

The second component of the entity score is based on the frequency and position of occurrences of a given entity within the retrieved passages. Each occurrence of an entity in a top-ranked passage counts 10 points, and each occurrence of an entity in any other passage counts 1 point. ("Top-ranked passage" means the passage or passages that received the maximal score from the passage retrieval component.) This score component is used as a secondary sort key, to impose a ranking on entities that are not distinguished by the first score component.

In counting occurrences of entities, it is necessary to decide whether or not two occurrences are tokens of the same entity or different entities. To this end, we do some normalization of entities. Dates are mapped to the format year-month-day: that is, *last Tuesday, November 9, 1999* and *11/9/99* are both mapped to the normal form *1999 Nov 9* before frequencies are counted. Person names are aliased based on the final word they contain. For example, *Jackson* and *Michael Jackson* are both mapped to the normal form *Jackson*.[3]

3 Evaluation

3.1 Results on the TREC-8 Evaluation

The system was evaluated in the TREC-8 question-answering track. TREC provided 198 questions as a blind test set: systems were required to provide five potential answers for each question, ranked in order of plausibility. The output from each system was then scored by hand by evaluators at NIST, each answer being marked as either correct or incorrect. The system's score on a particular question is a function of whether it got a correct answer in the five ranked answers, with higher scores for the answer appearing higher in the ranking. The system receives a score of 1, 1/2, 1/3, 1/4, 1/5, or 0, re-

[2]Perhaps less desirably, *people* would not be recognized as a synonym of *lives* in this example: *200 people* would be indistinguishable from *200 pumpkins*.

[3]This does introduce occasional errors, when two people with the same last name appear in retrieved passages.

System	Mean Ans Len	Answer in Top 5	Mean Score
Entity	10.5 B	46%	0.356
Passage 50	50 B	38.9%	0.261
Passage 250	250 B	68%	0.545

Figure 1: Results on the TREC-8 Evaluation

spectively, according as the correct answer is ranked 1st, 2nd, 3rd, 4th, 5th, or lower in the system output. The final score for a system is calculated as its mean score on the 198 questions.

The TREC evaluation considered two question-answering scenarios: one where answers were limited to be less than 250 bytes in length, the other where the limit was 50 bytes. The output from the passage retrieval component (section 2.1), with some trimming of passages to ensure they were less than 250 bytes, was submitted to the 250 byte scenario. The output of the full entity-based system was submitted to the 50 byte track. For comparison, we also submitted the output of a 50-byte system based on IR techniques alone. In this system single-sentence passages were retrieved as potential answers, their score being calculated using conventional IR methods. Some trimming of sentences so that they were less than 50 bytes in length was performed.

Figure 1 shows results on the TREC-8 evaluation. The 250-byte passage-based system found a correct answer somewhere in the top five answers on 68% of the questions, with a final score of 0.545. The 50-byte passage-based system found a correct answer on 38.9% of all questions, with an average score of 0.261. The reduction in accuracy when moving from the 250-byte limit to the 50-byte limit is expected, because much higher precision is required; the 50-byte limit allows much less extraneous material to be included with the answer. The benefit of the including less extraneous material is that the user can interpret the output with much less effort.

Our entity-based system found a correct answer in the top five answers on 46% of the questions, with a final score of 0.356. The performance is not as good as that of the 250-byte passage-based system. But when less extraneous material is permitted, the entity-based system outperforms the passage-based approach. The accuracy of the entity-based system is significantly better than that of the 50-byte passage-based system, and it returns virtually no extraneous material, as reflected in the average answer length of only 10.5 bytes. The implication is that NLP techniques become increasingly useful when short answers are required.

3.2 Error Analysis of the Entity-Based System

3.2.1 Ranking of Answers

As a first point, we looked at the performance of the entity-based system, considering the queries where the correct answer was found somewhere in the top 5 answers (46% of the 198 questions). We found that on these questions, the percentage of answers ranked 1, 2, 3, 4, and 5 was 66%, 14%, 11%, 4%, and 4% respectively. This distribution is by no means uniform; it is clear that when the answer is somewhere in the top five, it is very likely to be ranked 1st or 2nd. The system's performance is quite bimodal: it either completely fails to get the answer, or else recovers it with a high ranking.

3.2.2 Accuracy on Different Categories

Figure 2 shows the distribution of question types in the TREC-8 test set ("Percentage of Q's"), and the performance of the entity-based system by question type ("System Accuracy"). We categorized the questions by hand, using the eight categories described in section 2.3, plus two categories that essentially represent types that were not handled by the system at the time of the TREC competition: `Monetary Amount` and `Miscellaneous`.

"System Accuracy" means the percentage of questions for which the correct answer was in the top five returned by the system. There is a sharp division in the performance on different question types. The categories `Person`, `Location`, `Date` and `Quantity` are handled fairly well, with the correct answer appearing in the top five 60% of the time. These four categories make up 67% of all questions. In contrast, the other question types, accounting for 33% of the questions, are handled with only 15% accuracy.

Unsurprisingly, the `Miscellaneous` and `Other Named Entity` categories are problematic; unfortunately, they are also rather frequent. Figure 3 shows some examples of these queries. They include a large tail of questions seeking other entity types (mountain ranges, growth rates, films, etc.) and questions whose answer is not even an entity (e.g., "Why did David Koresh ask the FBI for a word processor?")

For reference, figure 4 gives an impression of the sorts of questions that the system does well on (correct answer in top five).

3.2.3 Errors by Component

Finally, we performed an analysis to gauge which components represent performance bottlenecks in the current system. We examined system logs for a 50-question sample, and made a judgment of what caused the error, when there was an error. Figure 5 gives the breakdown. Each question was assigned to exactly one line of the table.

The largest body of errors, accounting for 18% of the questions, are those that are due to unhandled

Question	Rank	Output from System
Who is the author of the book, The Iron Lady: A Biography of Margaret Thatcher?	2	Hugo Young
What is the name of the managing director of Apricot Computer?	1	Dr Peter Horne
What country is the biggest producer of tungsten?	1	China
Who was the first Taiwanese President?	1	Taiwanese President Li Teng hui
When did Nixon visit China?	1	1972
How many calories are there in a Big Mac?	4	562 calories
What is the acronym for the rating system for air conditioner efficiency?	1	EER

Figure 4: A few TREC questions answered correctly by the system.

Type	Percent of Q's	System Accuracy
Person	28	62.5
Location	18.5	67.6
Date	11	45.5
Quantity	9.5	52.7
TOTAL	67	60
Other Named Ent	14.5	31
Miscellaneous	8.5	5.9
Linear Measure	3.5	0
Monetary Amt	3	0
Organization	2	0
Duration	1.5	0
TOTAL	33	15

Figure 2: Performance of the entity-based system on different question types. "System Accuracy" means percent of questions for which the correct answer was in the top five returned by the system. "Good" types are in the upper block, "Bad" types are in the lower block.

What does the Peugeot company manufacture?
Why did David Koresh ask the FBI for a word processor?
What are the Valdez Principles?
What was the target rate for M3 growth in 1992?
What does El Nino mean in spanish?

Figure 3: Examples of "Other Named Entity" and "Miscellaneous" questions.

Errors	
Passage retrieval failed	16%
Answer is not an entity	4%
Answer of unhandled type: money	10%
Answer of unhandled type: misc	8%
Entity extraction failed	2%
Entity classification failed	4%
Query classification failed	4%
Entity ranking failed	4%
Successes	
Answer at Rank 2-5	16%
Answer at Rank 1	32%
TOTAL	100%

Figure 5: Breakdown of questions by error type, in particular, by component responsible. Numbers are percent of questions in a 50-question sample.

types, of which half are monetary amounts. (Questions with non-entity answers account for another 4%.) Another large block (16%) is due to the passage retrieval component: the correct answer was not present in the retrieved passages. The linguistic components together account for the remaining 14% of error, spread evenly among them.

The cases in which the correct answer is in the top five, but not at rank one, are almost all due to failures of entity ranking.[4] Various factors contributing to misrankings are the heavy weighting assigned to answers in the top-ranked passage, the failure to adjust frequencies by "complexity" (e.g., it is significant if *22.5 million* occurs several times, but not if *3* occurs several times), and the failure of the system to consider the linguistic context in which entities appear.

4 Conclusions and Future Work

We have described a system that handles arbitrary questions, producing a candidate list of answers ranked by their plausibility. Evaluation on the TREC question-answering track showed that the correct answer to queries appeared in the top five answers 46% of the time, with a mean score of 0.356. The average length of answers produced by the system was 10.5 bytes.

[4] The sole exception was a query misclassification caused by a parse failure—miraculously, the correct answer made it to rank five despite being of the "wrong" type.

There are several possible areas for future work. There may be potential for improved performance through more sophisticated use of NLP techniques. In particular, the syntactic context in which a particular entity appears may provide important information, but it is not currently used by the system.

Another area of future work is to extend the entity-extraction component of the system to handle arbitrary types (mountain ranges, films etc.). The error analysis in section 3.2.2 showed that these question types cause particular difficulties for the system.

The system is largely hand-built. It is likely that as more features are added a trainable statistical or machine learning approach to the problem will become increasingly desirable. This entails developing a training set of question-answer pairs, raising the question of how a relatively large corpus of questions can be gathered and annotated.

References

Steven Abney. 1996. Partial parsing via finite-state cascades. *J. Natural Language Engineering*, 2(4):337–344, December.

C. Buckley and A.F. Lewit. 1985. Optimization of inverted vector searches. In *Proc. Eighth International ACM SIGIR Conference*, pages 97–110.

Michael Collins and Yoram Singer. 1999. Unsupervised models for named entity classification. In *EMNLP*.

G. Salton, editor. 1971. *The Smart Retrieval System – Experiments in Automatic Document Processing*. Prentice-Hall, Inc., Englewood Cliffs, NJ.