Question Answering using the DLT System at TREC 2002

Richard F. E. Sutcliffe

Department of Computer Science
and Information Systems
University of Limerick
Limerick, Ireland

+353 61 202706 Tel
+353 61 202734 Fax
Richard.Sutcliffe@ul.ie Email
www.csis.ul.ie/staff/richard.sutcliffe URL

1. Introduction

This article outlines our participation in the Question Answering Track of the Text REtrieval Conference organised by the National Institute of Standards and Technology. Having not taken part before, our objective was to study the task and build a simple working system capable of answering at least some questions correctly. Only three person weeks was available for the work but this proved sufficient to achieve our goal. The article is structured as follows. Firstly, some preliminaries such as our starting point, tools and strategy are described. After this, the architecture of the Documents and Linguistic Technology Group's DLT system is outlined. Thirdly, the question types analysed by the system are described along with the named entities with which they work. Fourthly, the runs performed are presented together with the results we obtained. Finally, conclusions are drawn based on our findings.

2. Preliminaries

2.1 Previous Work

Our first step was to study previous attempts at the problem. Because of the limited time available, only two articles could be examined in detail. The first was Gaizauskas, Wakao, Humphreys, Cunningham and Wilks (1995). This describes the Sheffield LaSIE system and along the way provides many hints regarding techniques for effective information extraction and general text processing. The second was Rennert (2002). This excellent but unconventional article describes very concisely a number of key techniques used by the author in his 2001 TREC system. These ideas are highly pragmatic in nature and broken down by question type. In building our system we used Rennert's question taxonomy and methods as a starting point though in fact the final system was somewhat different as will be seen later.

2.2 Initial Tools

In computational terms, we started with some modest tools comprising a robust multiple pass parser (Sutcliffe, 2000) which we have used on a number of projects in Japanese (Sutcliffe and Nashimoto, 2000) and Spanish (Ruiz-Cascales, 2002) as well as English, a term recogniser and a general approach to natural language processing. To simplify the task we decided not to index the source documents ourselves but instead to use the TOPDOCS files provided by NIST and comprising the actual text of the top 50 matching documents found by the PRISE information retrieval system for each input query.

2.3 Overall Strategy

Beside the articles mentioned above we also devoted a small amount of time to looking at questions from previous years (mainly 2001) and establishing the relationship between them and answers in the corresponding documents. This led to an approximate strategy which has much in common with other QA systems and can be summarised as follows:

- Identify the type of the question;
- Based on the question type, search for appropriate named entities in the answer texts;
- Try to find a named entity which co-occurs with keywords from the question;
- Return the value of the 'best' such named entity as the answer.

3. Architecture of the System

3.1 Outline

We summarise here the architecture of the DLT system. Firstly, we identify the query type and hence the relevant named entities for which we will be searching. Secondly, we parse the 50 TOPDOCS Files dividing them into textual units using the markup. Thirdly, we search for instances of appropriate named entities in the textual units and mark them. Fourthly, we identify the winning named entity using one of two possible strategies: highest_scoring or most_frequent. We return this as the answer to the query. These stages are now dealt with in more detail.

3.2 Query Type Identification

Having identified the query types to use in the system, we studied questions of each type and developed simple keyword-based heuristics to recognise them. This is a very crude approach adopted due to shortage of time but it was suprisingly effective (Table 2): 425 of the 500 queries were correctly classified.

3.3 Text File Parsing

The text files are in XML-compliant form so it was easy to parse them without a Document Type Definition (DTD). Each document to be analysed was divided into a series of segments corresponding to a short passage of text. As with so many corpora, the level of markup varies from document to document and indeed we can not be sure that it has been used consistently. The strategy adopted was thus as follows: First, text within a HEADLINE tag was extracted. Second text within a TEXT tag was extracted and divided up into separate Ps. Finally a P was divided wherever three contiguous blanks were found. This last stage was to approximate sentence recognition. the resulting textual units were used in subsequent processing.

3.4 Named Entity Recognition

The type of question as identified in the first step determines the type of named entity or entities to be searched for. For example if the question type is what_state the entity is us_state i.e. 'California' etc. Each segment identified in the previous step was therefore inspected and all instances of appropriate named entities were identified.

Question Type	Example Question	Named Entities	Candidate Answer
state_bird	1517 What is the state bird of Alaska?	state_bird	Mockingbird
state_flower	1008* What is Hawaii's state flower?	state_flower	Yellow hibiscus
what_city	1520 What is the capital of Kentucky?	us_city, non_us_city	Houston
what_state	1743 Which state has the longest coastline on the Atlantic Ocean?	us_state	Calif.
what_county	1875 What county is St. Paul, Minnesota in?	us_county	Orange County
what_country	1496 What country is Berlin in?	country	Germany
what_continent	1489 What continent is India on?	continent	Asia
where	1500 Where is Georgetown University?	us_city, non_us_city us_state, us_county country	Phoenix, Ariz.
how_many3	1404 How many chromosomes does a human zygote have?	num	two chromosomes
how_much	1571 How much copper is in a penny?	num	20 percent
distance	1792 How far is it from Buffalo, New York to Syracuse, New York?	num, distance	100 miles
speed	1471 How fast does a cheetah run?	num, speed	60 miles an hour
temp	1606 What is the boiling point of water?	temp	212 degrees Fahrenheit
population	1750 What is Mexico's population?	num, population	one hundred million people
who	1395 Who is Tom Cruise married to?	proper_name	Nicole Kidman
when_interval	1056* When is hurricane season in the Caribbean?	date, interval	from June 1 to Nov. 30
length_of_time	1763 How old is the universe?	num, length_of_time	5 billion years
when	1698 When was Julius Caesar born?	date	100 B.C.
colour	1193* What color is a giraffe's tongue?	colour	black
unknown	1641 Where did 'N Sync get their name?	N/A	N/A

Table 1: Question Types used in the DLT system. The second column shows a sample question for each type. All are drawn from this year's data except for those question types which did not occur this year (indicated by an asterisk) where a sample from last year is shown. The third column lists the named entities which are used for answering a question of a particular type. The final column shows sample answers. These are all of appropriate types for the question but are not necessarily correct.

3.5 Answer Entity Selection

We experimented with two methods for selecting an answer which we call highest_scoring and most_frequent. In the first, we returned the named entity occurring in a textual unit which matched the keywords in the query best, chosen from any of the 50 PRISE documents. In the second, we returned the named entity which most frequently occurred in the vicinity of query keywords, observed across all occurrences of the entity in the 50 PRISE documents. Both strategies are unsophisticated but sometimes one or other of them can perform well on a particular query type.

In the next section we briefly outline the query types identified, the characteristics of the associated named entities and any special issues which affected processing for particular query forms.

Query Type	Classif.		Correct Classification								Incorrect Classification							
			Run 1				Run 2				Ru n 1				Run 1			
	C	NC	R	X	U	W	R	X	U	W	R	X	U	W	R	X	U	W
state_bird	1	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0
state_flower	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
what_city	13	0	2	0	1	10	2	0	1	10	0	0	0	0	0	0	0	0
what_state	1	5	0	0	0	1	0	0	0	1	1	0	0	4	1	0	0	4
what_county	3	0	0	0	0	3	0	0	0	3	0	0	0	0	0	0	0	0
what_country	8	1	2	0	0	6	2	0	0	6	0	0	0	1	0	0	0	1
what_continent	2	0	1	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0
where	40	3	11	1	1	27	7	0	1	32	0	0	0	3	0	0	0	3
how_many3	5	0	2	0	0	3	2	0	0	3	0	0	0	0	0	0	0	0
how_much	13	0	0	0	0	13	1	0	1	11	0	0	0	0	0	0	0	0
distance	21	0	4	0	0	17	4	0	0	17	0	0	0	0	0	0	0	0
speed	3	0	0	0	0	3	0	0	0	3	0	0	0	0	0	0	0	0
temp	2	0	1	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0
population	4	0	0	0	0	4	0	0	0	4	0	0	0	0	0	0	0	0
who	54	7	5	0	1	48	5	0	1	48	0	0	0	7	0	0	0	7
when_interval	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
length_of_time	7	0	1	0	0	6	1	0	1	5	0	0	0	0	0	0	0	0
when	92	6	15	1	1	75	15	1	1	75	0	0	0	6	0	0	0	6
colour	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
unknown	156	53	1	2	1	152	1	2	1	152	4	1	1	47	4	1	1	47
Totals	**425**	**75**	**45**	**4**	**5**	**371**	**42**	**3**	**7**	**373**	**5**	**1**	**1**	**68**	**5**	**1**	**1**	**68**

Table 2: Results by Query Type. The columns C and NC show the numbers of queries of a particular type which were classified correctly and not correctly. Those classified correctly are then broken down into Right, ineXact, Unsupported and Wrong for each of the two runs Run 1 and Run 2. Finally, those classified incorrectly are broken down in the same way.

4. Question Types and Corresponding Techniques

Given that time was short we were forced to build a basic system based on simple premises. We went about this by trying to identify the question types which occurred frequently in the 2001 question set and which could be answered using the above method. By the time the system had to be frozen for evaluation, there were twenty different question types as summarised in Table 1. Many more types could of course be added We outline here the techniques used for each one.

state_bird: Each state in the United States has its own state bird – a fascinating fact in itself made even more intriguing by the choice of the same bird by different states in several cases. A list of all such birds was readily drawn up. Any instances of these in documents could therefore be identified. This technique was hindered by two facts. Firstly, texts do not necessarily refer to a state bird by its official name – they might say Carolina Wren or even wren instead of Great Carolina Wren. Secondly, discussions about the bird for a particular state tend to occur close to mentions of other states and other state birds.

state_flower: The same strategy as for state birds was used, with the same limitations.

what_city: The TIPSTER Gazetteer Version 4.0 from the Consortium for Lexical Research (TIPSTER, 1992) was used to create a recogniser for place names. This gazetteer contains 246,907 entries of various kinds, mostly place names. Unfortunately our recogniser was too inefficient to use, so the list had to be reduced temporarily to a set of 199 US cities together with 167 capital cities for other counties. No heuristics were used for city recognition (e.g. X City is probably a city) due to lack of time. Some experiments with 'where' questions during system development suggested that the highest_scoring

strategy was not very satisfactory for places because when one city is mentioned others can be also (e.g. in texts about air travel). This was the reason for developing the most_frequent strategy.

what_state: A list of all US states was readily obtained. Each name has three official forms (e.g. 'Massachusetts', 'Mass.' and 'MA'). Unofficial abbreviations are rare so the standard names are probably sufficient (compare this to state flowers and birds above). State names can appear in isolation or in combination with other units to form a location specifier (e.g. 'Boston, MA').

what_county: The TIPSTER Gazetteer contains a complete list of US counties but this proved too long to handle so a simple heuristic was used: Any name of the form X County was deemed to be a US county. This proved quite effective.

what_country: A list of 162 non-US countries was used together with a list of nine names for the US itself.

what_continent: A list of seven continents was used.

where: Templates were devised which specify the form of a place using various combinations of country, state, county and city. The longest possible specifier (i.e. the most complete one) was used whenever a candidate place was found in a text.

how_many3: A recogniser was developed for numbers by inspecting a large number of examples. Forms recognised include '1', '1.1', '1.1 million', 'sixty four', 'sixty four million'. Since tokenisation of the text only joined contiguous sequences of alphabetic characters and left all others separate, parsing numbers was fairly straightforward. Each was converted into a canonical representation for comparison purposes. Initially this was an integer in the case of whole numbers but interestingly some numbers found in the texts are so large that this caused integer overflow. A string representation was thus used instead. For how_many3 questions, the units are picked up from the query and must match those used in the document. For example in 'How many chromosomes' the units are 'chromosomes' and these must follow the number found in the text.

how_much: In these queries the units are not specified in the query and must be deduced from the text. The word following a number is accepted as the units if it is '$', '%' or another word longer than two characters which is not a number.

distance: A distance is a number followed by some distance units. Twelve plural and twelve singular distance units were collected by inspection of TREC texts.

speed: A speed is a number followed by some speed units. 21 plural and eleven singular speed units were collected from the texts.

temp: A temperature is a number followed by appropriate units with various premodifiers (e.g. minus, '-') and postmodifiers (e.g. 'above Absolute Zero'). Fifteen basic units, two premodifiers and eight postmodifiers were collected from the texts. Interestingly, 'degrees' with no explicit units implies Fahrenheit in a US text.

population: A population is considered to be any number of value one million or more which occurs in a text portion matching keywords from the query. The unit is assumed to be 'People'.

Query Type	Run 1 Harsh %	Run 1 Kind %	Run 2 Harsh %	Run 2 Kind %
state_bird	0	0	0	0
state_flower	0	0	0	0
what_city	15	15	15	15
what_state	0	0	0	0
what_county	0	0	0	0
what_country	22	25	22	25
what_continent	50	50	50	50
where	26	28	16	18
how_many3	40	40	40	40
how_much	0	0	8	8
distance	19	19	19	19
speed	0	0	0	0
temp	50	50	50	50
population	0	0	0	0
who	8	9	8	9
when_interval	0	0	0	0
length_of_time	14	14	14	14
when	15	16	15	16
colour	0	0	0	0
unknown	0	1	0	1
Totals	9	11	8	10

Table 3: Overall Performance. The harsh figure in each case indicates the percentage of queries which were classified as being of that particular type (either rightly or wrongly) and which were answered correctly. The kind figure indicates the percentage of queries which were rightly classified as being of that type and which were answered correctly.

who: To answer questions about names, a simple name recogniser was constructed. This allows pre-name titles (e.g. 'Sen.') a basic name sequence (e.g. Dwight G. Morgan) and a post-name title (e.g. 'III'). Given names are constrained to come from the MOBY given name word list (Ward, 1996). Originally the surname was drawn from the MOBY surnames but this proved too restrictive. Any capitalised word is thus accepted as a surname.

when_interval: These questions ask about a range of dates e.g. 'When is the hurricane season'. A recogniser was thus built which can handle 'from DATE1 through to DATE2' and so on.

length_of_time: A length of time is simply a number followed by one of nineteen different time units. Composed lengths (e.g. 4 min 33 sec) are not handled at present.

when: These imply the occurrence of a date in the text. Accordingly a recogniser for most of the US date forms was developed (e.g. 'Saturday, October 17, 1987' etc.). The value of a year specifier was constrained to be less than 10,000 if accompanied by 'AD' etc. (either before or after the date) thus allowing cases like '5,000 B. C.' On the other hand years with no 'AD' etc. were constrained to lie within the range 1700 and 2010. Dates in 'slash' or 'dot' formats (e.g. '02.06.02' or '02/06/02' to mean June 2, 2002 or perhaps February 6, 2002) are not handled.

Run	Qns	Time	Docs	Words	Characters	Min/Q	Words/Min	Words/Sec
Run 1	500	11h 0m	25,000	15,000,000	93,000,000	1.3	23,000	383
Run 2	500	7h 5m	25,000	15,000,000	93,000,000	0.85	35,294	588

Table 4: Timings and Counts for DLT. The columns indicate the identity of the run, the number of questions, the time to process all questions, the number of documents, words and characters processed, the time to process one query in minutes, and the numbers of words processed per minute and per second. See text for the system specification. The difference in timings for the two systems is caused merely by the fact that results of the SGML parsing were saved the first time around and therefore did not have to be done again.

colour: A list of nineteen colours was used. No doubt there are more but a longer list could not be found. In any case, no colours came up this year, only lists of colours.

unknown: This is the default query category. Approximately 156 queries fall into it. By definition these can not be answered by the system. As a crude experiment, all unknown queries were treated as if they were 'where', 'when' or 'who' in that order, accepting the first answer found.

5. Runs and Results

5.1 Two Experiments

We conducted two experiments each of which resulted in one run. All that varied in the runs was the method used for answer selection relative to each of the 20 query types. In the first run, what_continent, where, how_much and length_of_time were answered using the most_frequent strategy while the remainder were answered using highest_scoring. In the second run, all were answered using highest_scoring.

5.2 Results

Results are summarised in Tables 2 and 3. In Table 2, the two columns entitled 'Classif.' show the number of queries which were classified correctly (C) and incorrectly (NC) broken down by query type. Overall therefore, 425 of the 500 queries (85%) were categorised correctly. This seems quite a good result considering that the method used was based solely on identification of keywords and did not involve any structural analysis of the queries. By far the largest category of course is 'unknown' which accounts for approximately 156 queries (31%). We can conclude from this that 31% of the TREC 2002 questions fall completely out of the scope of this system. The remaining columns give counts of answers falling into the standard Right, ineXact, Unsupported and Wrong categories according to NIST assessors. The figures under 'Correct Classification' refer to queries which were correctly classified. Those under 'Incorrect Classification' refer to queries which should not be in the category. Very occasionally these were also answered correctly by chance.

Table 3 shows the overall performance of the system as a simple percentage of answers which were correct, broken down by question type. The Harsh figures take into account incorrectly classified queries as well as correctly classified ones. Clearly incorrectly classified queries will be answered incorrectly in the vast majority of cases, as shown by the last eight columns of Table 2. The Kind figures leave out incorrectly classified queries. We ignore the confidence rating of the system since we built in no mechanism for this. Dealing with Harsh figures, therefore, the overall performance of the system is 9% in Run 1 and 8% in Run 2. The best level of performance is for types 'what_continent' (50%), 'temp' (50%) and 'how_many3' (40%) but these are not representative because the number of each is small.

Perhaps the optimum view of the system is suggested by the figure of 26% Harsh (28% Kind) for 'where' queries in Run 1.

5.3 Timings

To give an indication of performance in terms of times, we used a Dell OptiPlex GX200 running NT4.0 at a clock speed of 733 MHz and having 256 Mb RAM. The whole system was written Quintus Prolog Release 3.4. Run times and related figures are provided in Table 4. Interestingly, the SGML parsing component is the slowest in the system and it is this which accounts for the difference in run times shown in the table. Speeding up this component therefore would be the most effective way of increasing performance.

6. Conclusions

Our objective was to get started in open domain question answering and to make as much progress in three weeks as possible. By the end of the time our system was up and running, giving a performance of 9% overall in Run 1. It has proved a useful experience and has helped us to focus on possible techniques for answering closed domain questions such as those analysed in Sutcliffe and Kurohashi (2000). There are of course many next steps to be taken which we summarise below:

- Analysing the query in more detail by parsing and hence identifying query type more accurately together with other information (e.g. the units in less simple cases such as 'How many pieces of clothing');
- Tidying up existing named entity recognisers and checking their performance;
- Adopting answer patterns rather than just using keyword co-occurrence, e.g. in the manner of Hovy et al. (2002);
- Determining other simple query types which account for the majority of the queries currently classified as unknown;
- Designing a strategy for identifying queries which have no answers;
- Developing the ability to recognise ad hoc named entities. For example in 'What type of car is used by Linda Ronstadt' we need to know all the car types despite having no pre-planned recogniser for them;
- Assigning a confidence rating to responses.

To finish on a more general note, two issues came to mind during this research. Firstly, a general theme within question answering seems to be the development of increasingly sophisticated pattern matching devices based on more and more detailed classifications of questions. To what extent however, will this lead to general findings about question answering or natural language processing rather than more and more brittle sets of rules?

Secondly, in TREC we can not always predict what types of question might be asked in the next competition. There were a few surprises this year, two examples being 1641 'Where did 'N Sync get their name?' and 1710 'What are the colors of the Italian flag?'. The first question is open ended in scope while the second involves lists of objects which we had assumed would be restricted to the list sub-track. Perhaps it would be fair for the general characteristics of new query types to be indicated so that some kind of strategy could be worked out for them.

8. References

Gaizauskas, R., Wakao, T., Humphreys, K., Cunningham, H., & Wilks, Y. (1995). University of Sheffield: Description of the LaSIE System as used for MUC-6. In *Proceedings of the Sixth Message Understanding Conference (MUC-6)* (pp. 207-220). Morgan Kaufmann.

Hovy, E. et al. (2002). *A Typology of over 140 Question-Answer Types.* Accessed 2002. http://www.isi.edu/natural-language/projects/webclopedia/Taxonomy/taxonomy_toplevel.html

Rennert, P. (2002). Word proximity QA system. In E. M. Voorhees and D. K. Harman (Eds.) *Proceedings of the Tenth Text REtrieval Conference (TREC 2001)* (pp. 179-191). Gaithersburg, Maryland: Department of Commerce, National Institute of Standards and Technology, NIST Special Publication 500-250.

Ruiz-Cascales, R. (2002). *A Specification and Validating Parser for Simplified Technical Spanish.* M.Sc. Thesis, University of Limerick, Ireland.

Sutcliffe, R. F. E. (2000). Using A Robust Layered Parser to Analyse Technical Manual Text. *Cuadernos de Filología Inglesa*, **9**(1), 167-189. Número Monográfico: *Corpus-based Research in English Language and Linguistics*, University of Murcia, Spain.

Sutcliffe, R. F. E., & Kurohashi, S. (2000). A Parallel English-Japanese Query Collection for the Evaluation of On-Line Help Systems. *Proceedings of the Second International Conference on Language Resources and Evaluation, Athens, Greece, 31 May - 2 June, 2000*, 1665-1670.

Sutcliffe, R. F. E., & Nashimoto, N. (2000). Robust Parsing of Japanese Technical Text: An Initial Study. *Proceedings of the Eleventh Irish Conference on Artificial Intelligence and Cognitive Science, National University of Ireland Galway, 23-25 August, 2000.*

TIPSTER (1992). *Gazetteer 4.0.* ftp://crl.nmsu.edu/CLR/lexica/gazetteer/

Ward, G. (1996). *MOBY Project Wordlists.* ftp://ftp.dcs.shef.ac.uk/share/ilash/Moby