Natural Language Engineering **9** (3): 231–267. © 2003 Cambridge University Press
DOI: 10.1017/S1351324903003176 Printed in the United Kingdom

Open-domain textual question answering techniques

SANDA M. HARABAGIU
Department of Computer Science, University of Texas at Dallas, Richardson, TX 75083, USA
(*e-mail*: sanda@cs.utdallas.edu)

STEVEN J. MAIORANO
Department of Computer Science, University of Sheffield, Sheffield S1 4DP, UK

MARIUS A. PAŞCA
Language Computer Corporation, Dallas, TX 75206, USA
(*e-mail*: marius@languagecomputer.com)

(*Received 8 December 2001; revised 11 November 2002*)

Abstract

Textual question answering is a technique of extracting a sentence or text snippet from a document or document collection that responds directly to a query. Open-domain textual question answering presupposes that questions are natural and unrestricted with respect to topic. The question answering (Q/A) techniques, as embodied in today's systems, can be roughly divided into two types: (1) techniques for Information Seeking (IS), which localize the answer in vast document collections; and (2) techniques for Reading Comprehension (RC) that answer a series of questions related to a given document. Although these two types of techniques and systems are different, it is desirable to combine them for enabling more advanced forms of Q/A. This paper discusses an approach that successfully enhanced an existing IS system with RC capabilities. This enhancement is important because advanced Q/A, as exemplified by the ARDA AQUAINT program, is moving towards Q/A systems that incorporate semantic and pragmatic knowledge enabling dialogue-based Q/A. Because today's RC systems involve a short series of questions in context, they represent a rudimentary form of interactive Q/A which constitutes a possible foundation for more advanced forms of dialogue-based Q/A.

1 Introduction

1.1 Background

In today's world, a significant percentage of the information overload is derived from the availability of more and more on-line text documents. Traditionally, locating information of interest in this environment of burgeoning textual data can be achieved through Information Retrieval (IR) technology and systems. Given a query, an IR system returns a list of potentially relevant documents which the user must then scan to search for pertinent information. To take a step closer to

https://doi.org/10.1017/S1351324903003176 Published online by Cambridge University Press

true *information retrieval* rather than *document retrieval*, TREC[1] initiated in 1998 an experimental task, the Question Answering (Q/A) Track whose aim was to foster research in domain-independent textual Q/A. The technology that emerged from this TREC-initiated task approximates an Information Seeking (IS) application, in which a user poses a question in natural language and receives the answer as a text snippet as short as a word or as long as a sentence. The text snippets are derived from a 3 Gigabyte textual collection and contain information that answers the question. The user's question is not restricted to any pre-defined domain which makes the task truly open-domain and inherently complex. More often than not, the document in which the answer resides contains information that would enable a system to handle follow-up questions. This latter processing approximates a Reading Comprehension (RC) task in which a single document can be the focus of a series of questions.

Although the IS Q/A has shown good performance in the TREC evaluations for factual questions and, therefore, proven to be an improvement over traditional information retrieval, it too has shown certain inadequacies and deficiencies. First, current IS Q/A systems handle successfully only trivia-type questions like *"In 1990, what day of the week did Christmas fall on?"*. Secondly, current IS Q/A systems process questions in isolation, disregarding the context of the previous questions. Thirdly, current IS Q/A systems cannot handle systematically ambiguous questions like *"Where is the Taj Mahal?"*, which depending upon the intent of the user, may be answered correctly by both *Agra, India*, when referring to the Indian monument, or by *Atlantic City*, when referring to the casino of the same name. Fourth, since IS Q/A systems handle only trivia-like questions, they do not recognize the open-ended nature of questions like *"What are the causes of violence in the Middle East?"*. The techniques employed in Reading Comprehension (RC) Q/A systems begin to address the first two inadequacies mentioned above. Although this would be a first step in developing a more sophisticated Q/A system, an IS Q/A architecture supplemented by RC Q/A techniques might enable a user to move beyond IS operations and ask dialogue-like follow-up questions. Such an enhancement is important because advanced Q/A, as exemplified by the Advanced Research and Development Activity's (ARDA) AQUAINT program[2], is moving towards Q/A systems that incorporate semantic and pragmatic knowledge enabling dialogue-based Q/A.

1.2 Question answering-based information seeking

While IR systems have made important strides in the past decades, the problem of efficiently locating information in large on-line document collections is far from being solved. The TREC Q/A track defined the open-domain textual Q/A task

[1] The Text REtrieval Conference (TREC) is a series of workshops organized by the National Institute of Standards and Technology (NIST) designed to advance the state-of-the-art in IR.

[2] Information about the Advanced QUestion and Answering for INTelligence (AQUAINT) program can be found at http://www.ic-arda.org/InfoExploit/aquaint/

https://doi.org/10.1017/S1351324903003176 Published online by Cambridge University Press

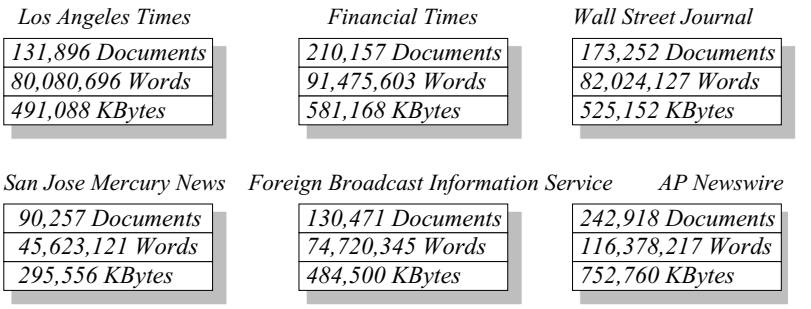


Fig. 1. Text collections used in the TREC Q/A evaluations.

as an important step towards the goal of locating and returning only information sought by a user. More specifically, in the TREC Q/A evaluations the answer output was in the form of a ranked list of *five* [document-id, answer-string] pairs for each question; the answer-strings were either short (50 bytes) or long (250 bytes)[3] and contain the candidate answer to the question. Thus, instead of reading a set of documents in search for the desired information, the user is presented with at most five answers responding to the posed question.

In the TREC Q/A evaluations the answer is known to reside somewhere in the 736,794 documents (3 Gigabytes) from six different news sources (figure 1). Having multiple document sources of data gives the task three added dimensions: (1) *a large document collection* necessitates the processing of thousands of documents in order to answer each question; (2) since the same information is often covered by multiple news sources *answer redundancy* must be handled; and (3) because sometimes a candidate answer contains a *piece* of the complete answer, *supplemental information* must be found and used.

To search for information in voluminous document collections, developers of current IS Q/A systems use IR techniques for processing the documents. This involves both indexing the entire collection and developing a retrieval mechanism. Three forms of indexing are typical in current IS Q/A systems: (a) term or word-based indexing; (b) conceptual indexing; and (c) paragraph indexing.

Term or word indexing methods range from the creation of a simple index structure that associates each word with the documents where it occurs to more complex indexes. The latter are comprised of multi-word term identifiers, document identifiers, syntactic, morphologic and semantic variants of the term as well as the text sequence representing the term. LIMSI's QUALQ system (Ferret, Grau, Hurault-Plantet, Illouz and Jacquemin 2001) is an example of such a complex term indexing scheme. Conceptual indexing is based on a conceptual taxonomy that is built from the document collection and linked to a word-based index. The conceptual

[3] In TREC-8 and TREC-9, both short and long answers were accepted. In TREC-10 only short answers were considered while in TREC-11 the answer must be exact and not embedded in a text snippet. For example, if the question is *"What was the population of Washington D.C. in 2000?"*, the answer should be only *"572,059"*.

https://doi.org/10.1017/S1351324903003176 Published online by Cambridge University Press

TREC8: Questions 1–200

Q3: What does the Peugeot company produce?
Q6: Why did David Koresh ask the FBI for a word processor?
Q8: What is the name of the rare neurological disease with symptoms such as involuntary movements (tics), swearing, and incoherent voclizations (grunts, shouts, etc)?
Q30: What are the Valdez Principles?
Q36: In 1990, what day of the week did Christmas fall on?
Q73: Where is the Taj Mahal?

TREC9: Questions 201–893

Q204: What type of bridge is the Golden Gate Bridge?
Q253: Who is William Wordsworth?
Q285: When was the first railroad from the east coast to the west coast completed?
Q361: How hot does the inside of an active volcano get?
Q412: Name a film in which Jude Law acted.
Q425: How many months does a normal human preganancy last?
Q581: What flower did Vincent Van Gogh paint?

TREC10: Questions 894–1393

Q902: Why does the moon turn orange?
Q905: What person's head is on a dime?
Q916: What river in the US is known as the Big Muddy?
Q949: What does cc in engines mean?
Q1022: What is Wimbledon?
Q1111: What are the spots on dominoes called?
Q1136: What causes gray hair?

Fig. 2. Sample questions used in the first three TREC Q/A evaluations.

taxonomy integrates syntactic, semantic and morphological relationships. Woods, Bookman, Houston, Kuhns, Martin and Green (2000) describe the conceptual indexing mechanisms implemented in the IS Q/A system from SUN. Conceptual indexing, by using conceptual subsumption of question words, enables retrieval of document passages that may contain the answer, as reported in Woods, Green and Martin (2000). Paragraph indexing, which was implemented in the *LASSO* (Moldovan, Harabagiu, Paşca, Mihalcea, Goodrum, Gîrju and Rus 2000) and *FALCON* (Harabagiu, Moldovan, Paşca, Mihalcea, Surdeanu, Bunescu, Gîrju, Rus and Morarescu 2000) IS Q/A systems is motivated by the notion that an answer might contain co-occurrences of question words localized in a single paragraph and, therefore, this scheme associates words with paragraphs in which they co-occur. Document indexes are used in current IS Q/A systems to implement three different forms of retrieval: (1) retrieval that lists and ranks the documents containing the question terms; (2) retrieval of documents followed by ranking of their passages; and (3) retrieval and ranking of document passages. All three types of retrieval mechanisms employ both Boolean and vector retrieval models.

Whenever a question is posed to an IS Q/A system, the question words are processed such that the system generates a query that retrieves document passages for answer extraction. Figure 2 lists some sample questions used in the first three TREC Q/A evaluations. In TREC-8, the 200 questions originated from the FAQFinder (Burke *et al.* 1997) and from the questions generated by TREC participants or the NIST staff. In TREC-9, 504 questions were retrieved from the logs of Microsoft's

https://doi.org/10.1017/S1351324903003176 Published online by Cambridge University Press

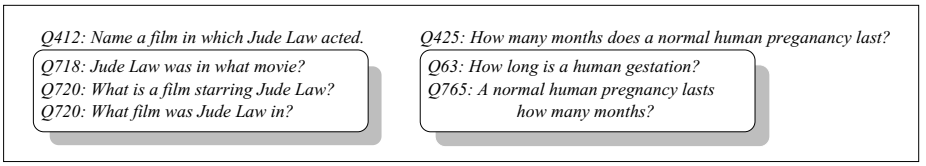


Fig. 3. TREC-9 questions and their reformulations.

Encarta or the Excite search engine. In TREC-10, the 893 questions from TREC-8 and 9 were supplemented by 500 additional questions mined from the logs of MSNSearch logs and AskJeeves. Each question in TREC-8 and 9 had an answer in the document collection, but TREC-10 evaluated questions that had no answers. In the latter case, the correct answer was NIL rather than the [document-id, answer-string] pair.

Recent results from TREC evaluations (Kwok, Grunfeld, Dinstl and Chan 2000; Radev, Prager and Samn 2000; Allen, Connell, Croft, Feng, Fisher and Li 2000) show that Information Retrieval (IR) techniques alone are not sufficient for finding answers to questions similar to those illustrated in figure 2. In fact, more and more systems are adopting architectures in which the question semantics are captured prior to retrieving a text paragraph where the answer may lie (Gaizauskas and Humphreys 2000; Hovy, Gerber, Hermjakob, Lin and Ravichandran 2001; Harabagiu, Paşca and Maiorano 2000). The question semantics are later used to extract the answer from text paragraphs (Abney, Collins and Singhal 2000; Radev *et al.* 2000). When processing a natural language question, two criteria must first be met. First, we need to know what the expected answer type is, i.e. the type of information that is being sought. Secondly, we need to know where to look for the answer, i.e. we need to identify the question keywords used for retrieving the document passage that might contain the answer.

Furthermore, although natural language questions may be phrased in several different ways, IS Q/A systems are expected to return the same answer regardless of the question paraphrase. This capability was evaluated in TREC-9 when the NIST staff reformulated 53 questions derived from the logs of Encarta or Excite into a total of 189 paraphrased questions. Figure 3 lists two TREC-9 questions and their corresponding reformulations.

Not only can information be sought via questions with varying syntax and terms, but appropriate answers too may be phrased differently. Thus *answer redundancy*, in which answers appear in different linguistic guises, is exploited by several current IS Q/A systems (Abney *et al.* 2000; Clarke, Cormack and Lynam 2001; Kwok *et al.* 2000), most notably by Microsoft's AskMSR system (Brill, Lin, Banko, Dumais and Ng 2001). The designers of AskMSR used answer redundancy for two purposes: (a) to facilitate the matching of questions with answers; and (b) to facilitate answer extraction. To match questions to answers, AskMSR uses the Web instead of the TREC collection alone since answer redundancy is far greater on the Web. It is, therefore, more likely to find an answer that matches the original question without first having to transform the question into several of its variants. Furthermore,

https://doi.org/10.1017/S1351324903003176 Published online by Cambridge University Press

QL2: Name 32 countries Pope John Paul II has visited.
QL5: What are 10 U.S. cities that are locations of homes designedby Frank Lloyd Wright?
QL14: Name 5 diet sodas.
QL15: Who are 6 actors who have played Tevye in "Fiddler on the Roof"?
QL18: Name 30 individuals who served as a cabinet officer under Ronald Reagan.

Fig. 4. Example of list questions.

CTX1a: Which museum in Florence was damaged by a major bomb explosion in 1993?
CTX1b: On what day did this happen?
CTX1c: Which galleries were involved?
CTX1d: How many people were killed?
CTX1e: Where were these people located?
CTX1f: How much explosive was used?

Fig. 5. Example of context questions.

although the preponderance of data on the Web facilitates simple matching between question and answer terms, AskMSR answer may still not be correct. In this case, the answer redundancy strategy is used to combine a number of uncertain answers into a single, more viable guess (Brill *et al.* 2001).

Some open-domain questions are harder to answer because they require the IS Q/A systems to assemble or fuse information located in multiple documents. Figure 4 shows several TREC-10 so-called "list questions" whose answers contain several instances of a particular kind of information connoted by the question. In other words, the "list answers" are comprised of several *complementary answers* which are each, in themselves, correct, but only partially so in the overall context of the question. Such list questions are harder to answer because information duplicated in documents must be detected and reported only once. Therefore, although answer redundancy may be beneficial for processing certain questions, this strategy is detrimental to processing list questions.

Advanced IS Q/A systems should not process questions in isolation. Eventual users of IS Q/A systems will likely interact with their systems on a regular basis and would expect to have their questions processed in the context of their previous interactions. In TREC-10 a context Q/A task was designed to represent the dialogue processing that an IS Q/A system would require for supporting an interactive user session. Figure 5 represents a series of questions that were evaluated in the TREC-10 context Q/A task.

As reported in Voorhees (2001), the results of this task were unexpected. The ability to correctly answer questions later in a series was uncorrelated with the ability to answer questions earlier in the series. Apparently, the first question in a series defined a small enough subset of documents such that the results of the overall interaction were dominated by whether the system could answer the particular current question. In contrast, RC Q/A systems use series of questions

https://doi.org/10.1017/S1351324903003176 Published online by Cambridge University Press

posed with reference to a single document, but the ability of answering questions depends more tightly on the previous interactions. Therefore, techniques used in an accurate RC Q/A system could enhance an IS Q/A system and enable it to process successfully questions asked in the context of a dialogue. It was this observation that motivated our interest in RC Q/A systems and the goals of reading comprehension. In contrast to IS Q/A systems, RC questions focus on a single document and a system's ability to answer correlates more closely with the previous Q/A interaction. Therefore, we hypothesized that incorporating this RC functionality into an IS Q/A architecture would advance the state-of-the-art in Q/A overall and track closely to the long term goals of the AQUAINT program.

1.3 Question answering for reading comprehension tests

Traditionally, story comprehension was considered a form of text understanding that provided interesting research problems for narrative inference and for the world knowledge representation imposed by text pragmatics. Initially, researchers of human language believed that successful story-comprehension systems needed to make the same kinds of inferences people make when reading a story, and therefore needed to have access to the same kind of knowledge people use when making inferences. To this end Schank and Abelson (1977) developed a theory of human knowledge structures used as a basis for a number of story-understanding systems (Cullingford 1977; DeJong 1977; Wilensky 1976). One way of evaluating the accuracy of story-understanding systems was to test them through a question-answering process. Two particular story-understanding systems, SAM (Cullingford 1977) and PAM (Wilensky 1976) were used in conjunction with one of the earliest Q/A systems, Wendy Lehnert's QUALM (Lehnert 1978).

QUALM was conceived as an implementation of a general model of question-answering. The primary representation underlying QUALM was Schank's *Conceptual Dependency* (Schank 1972), which is a major departure from the traditional IR model adopted by current IS Q/A systems. QUALM approached its Q/A task in two stages: understanding the question and finding the answer. Each of these stages were further divided in two steps. Question understanding was comprised of (1) *conceptual categorization* responsible for classifying questions and (2) *inferential analysis* for determining what the questioner really intended when a question was not be taken literally[4]. Finding the answer was divided into (3) *content specification* to determine how detailed or elaborate the answer should be, and (4) *searching heuristics* for actually extracting the answer from the memory representation of the story. All four phases of QUALM's processing correspond to areas of research promoted by the advanced Q/A requirements of the AQUAINT program. In QUALM, however, all these phases are dependent upon the Conceptual Dependency schema

[4] Lehnert (1978) shows that when asking *"Do you have a light?"*, the questioner is not asking to see if his interlocutor possesses a light, but merely he asks his interlocutor to offer him a light.

https://doi.org/10.1017/S1351324903003176 Published online by Cambridge University Press

that has several limitations, the major one being the inability to operate on arbitrary documents in various domains.

Recently, story comprehension was revisited from a different perspective, i.e. one that is not interested with specific aspects of knowledge representation or inference techniques, but rather with a simple, bags-of-words approach that would pick a sentence from a story as a response to an ad hoc question. This approach was first reported in the DEEPREAD system (Hirschman, Light, Breck and Burger 1999). DEEPREAD supports *fact retrieval* as opposed to document retrieval by finding the best match between the word set representing a question and the word sets representing the sentences in the document. DEEPREAD measures the match by the size of the intersection of the two word sets. Because match size does not produce a complete ordering on the document sentences, sentences that match on longer words and occur earlier in the document are preferred as answers. Additionally, normalizations and extensions of the word sets are possible by (a) removing the stop words[5], (b) stemming to remove inflectional affixes from the words; and (c) using name recognizers for persons, locations and temporal information and associating them with the question stems. The notion of question stem was introduced by Lehnert (1978), who first observed the association of question categories with words that usually start questions, e.g. *why*, *who* or *how far*, calling these words *question stems*.

DEEPREAD was tested on a corpus of 115 children's stories provided by REMEDIA Publications for reading comprehension. The comprehension of each story is tested by answering five standard questions – starting with *who*, *what*, *when*, *where* and *why*. For RC Q/A the identification of a sentence from the story constitutes an answer. In the REMEDIA corpus, the correct answers are annotated, thus enabling the comparison between the answer sentence returned by an automatic question-answering system and the answer sentence selected by a person at annotation time. The same corpus was later used by several other RC Q/A systems such as QUARC (Riloff and Thelen 2000), AQUAREAS (Ng, Teo and Kwan 2000) and the Brown University statistical language processing class project (Charniak, Altun, de Salvo Braz, Garrett, Kosmala, Moscovich, Pang, Pyo, Sun, Wy, Zang, Zeller and Zorn 2000). We used the same corpus in our effort to incorporate RC Q/A capabilities into an IS Q/A system.

In the REMEDIA corpus, approximately 10% of the questions do not have an answer, and several questions have multiple correct answers. All stories were manually annotated with the correct answers. When we performed our experiments, we considered only questions that have annotated answers. An example story from the REMEDIA corpus and its five associated questions are illustrated in figure 6. Figure 6 shows the annotated answers A_i corresponding to each question Q_i.

The DEEPREAD developers have observed that reference resolution information and named entity-related semantic information played an important role in identifying the answer. Therefore, all stories were also manually annotated with

[5] In IR, nouns, verbs, adjectives and adverbs are known as "content words" whereas all the other words from the vocabulary of the documents are known as "stop words".

https://doi.org/10.1017/S1351324903003176 Published online by Cambridge University Press

School Kids Clean Up Creek!

(EVERETT, WASHINGTON, June, 1988) It has taken five years of hard work. **A3**
But a group of children have cleaned up Pigeon Creek. **A1,A2**
In 1983, the creek was so dirty that the fish had all died. **A3** A group of kids at Jackson School decided to do something about it. It became a project for their whole class.
First they cleaned out the bottom of the creek, called the creek bed. **A4** There were many weeds. They also found lot of trash and litter. Then the students raised fish eggs in a tank of water called an aquarium. When the eggs hatched, they emptied the new fish into the creek. They wanted others to stop making the creek dirty. **A5** So they made signs that said "Don't Dump". They put the signs near the creek. They also made little books, called booklets. The booklets told people how to keep the creek clean. Now, the water is clean and new fish are being born in the creek. Pigeon Creek is a sign of the good things young people can do. **A2**

Q1: *Who cleaned up the creek?* **Q2:** *What was the name of the creek?* **Q3:** *When did the work begin?*
Q4: *Where is the bed of a creek?* **Q5:** *Why did the kids make booklets?*

Fig. 6. A REMEDIA story annotated with answers. The five questions Q1–Q5 are also listed. The A1–A5 annotations in correspond to the questions Q1–Q5.

(a) coreference data indicating the antecedents of pronouns and other anaphors and (b) named entity categories indicating whether a proper name represents a person, an organization, a location or a date. Figure 7 illustrates the mark-ups of two coreference chains: R_1 represents all references to a group of kids from Jackson School and R_2 representing all references to Pigeon Creek. Answering question Q_1 involves the selection of a sentence that mentions the cleaning event, contains a reference to Pigeon Creek and identifies who performs the cleaning. Two candidate sentences satisfy all constraints: the title and the sentence marked A_1. By selecting the most informative sentence, the title is filtered out. Sentence A_1 is more informative because it identifies uniquely the creek as being Pigeon Creek, the only creek mentioned in the story. The words *kids* and *children* are, according to WordNet, synonymous, so there is no difference in their degree of informativeness. Unlike information-seeking Q/A, reading comprehension relies on reference information that spans the entire text. Moreover, since all questions refer to the same text, frequently the answer of a previous question (e.g. A_1) will be used in a follow-up question. Therefore, we see that the subject of question Q_5 – "the kids" – corefers with the "group of children" in A_1, and that "they" in A_5 corefers with both "the kids" and the "group of children".

As noted in Ng *et al.* (2000), text coherence information is needed for answering *why* questions. Discourse cue phrases such as *because* or *so* indicate causation coherence relations between sentences and these relations map to the motivation sought by *why* questions. For example, the sentence *"So they made signs that said*

https://doi.org/10.1017/S1351324903003176 Published online by Cambridge University Press

School Kids Clean Up Creek!
R1 R2
(EVERETT, WASHINGTON, June, 1988) It has taken five years of hard work.

But a group of children have cleaned up Pigeon Creek.
R1 R2
In 1983, the creek was so dirty that the fish had all died. A group of kids at Jackson
R2 R1
School decided to do something about it. It became a project for their whole class.
R1
First they cleaned out the bottom of the creek, called the creek bed. There were many
R1 R2 R2
weeds. They also found lot of trash and litter. Then the students raised fish eggs in a
R1 R1
tank of water called an aquarium. When the eggs hatched, they emptied the new fish
R1
into the creek. They wanted others to stop making the creek dirty. So they made signs
R2 R1 R2 R1
that said "Don't Dump". They put the signs near the creek. They also made little books,
R1 R2 R1
called booklets. The booklets told people how to keep the creek clean. Now, the water
R2
is clean and new fish are being born in the creek. Pigeon Creek is a sign of the good
R2 R2
things young people can do.

Fig. 7. A REMEDIA story annotated with reference links.

"Don't Dump" elaborates upon the intent of the kids expressed in A_5. The sentence *"They also made little books, called booklets"* has two functions: (1) it is the only sentence in the text that refers to both booklets and the kids and (2) it indicates an additional effect of the event mentioned in sentence A_5. Therefore, the processing of Q_5 is based on reference data and the combined coherence effect of the cue phrases *so* and *also*.

2 System architectures for question answering

2.1 Answer engines

The typical architecture of an answer engine used for IS Q/A comprises three modules as shown in figure 8:

1. The *Question Processing* module captures the semantics of the natural language question which enables the recognition of the *expected answer type*. For example, given question Q_3 illustrated in figure 3, the expected answer type is identified as a PRODUCT or ARTIFACT such as *cars*, *bikes* or *pens*. In addition, the question keywords that are used to retrieve text passages where the answer may lie are identified during question processing.
2. The *Document Processing* module uses an index of the document collection to retrieve text passages that (a) contain all the question keywords and (b) have at least one concept of the same semantic class as the expected answer type if the latter is known. For question Q_3, passages mentioning *Peugeot* and any PRODUCTS are considered relevant.
3. The *Answer Processing* module compares the semantics of the answer against the semantics of the question before extracting the answer.

https://doi.org/10.1017/S1351324903003176 Published online by Cambridge University Press

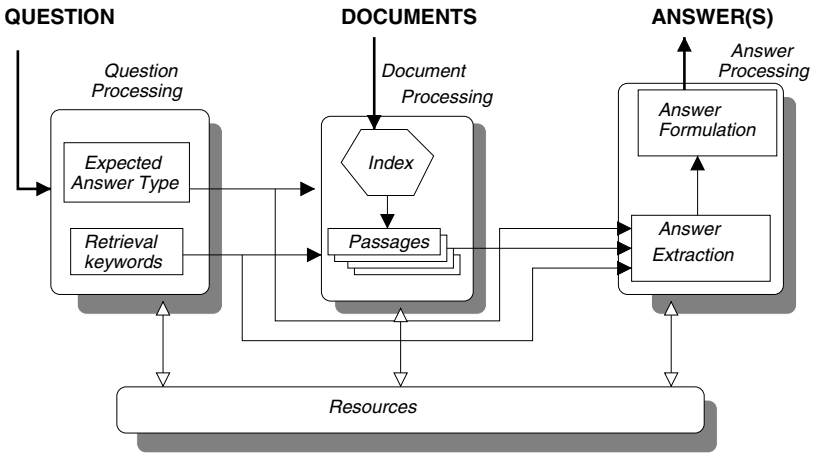


Fig. 8. The architecture of an answer engine.

Several notable variations from the standard IS Q/A architecture were implemented in (1) the IR-based Q/A system reported in Kwok *et al.* (2000); (2) the statistical Q/A system presented in Ittycheriah, Franz, Zhu, Ratnaparkhi and Mammone (2001); (3) the Webclopedia Q/A system based on using Q/A typologies to pinpoint answers, as described in Hovy *et al.* (2001); and (4) the NE-focused Q/A system detailed in Srihari and Li (2000). Yet another new IS Q/A architecture was implemented in the *FALCON* Q/A system, which used several feedback loops that enhanced its chance of finding the correct answer (Pasca and Harabagiu 2001). Since *FALCON* was one of the top-performing systems in the TREC 9 and 10 evaluations, we chose to use it as a basis for enhancing an IS Q/A system with RC Q/A capabilities. The architecture of *FALCON* is illustrated in figure 9.

In *FALCON*, the question is parsed to extract (1) the expected answer type and (2) an ordered list of keywords used to retrieve relevant text passages, or *text paragraphs*, where the keywords and concepts of the expected answer type are found. The parse tree of the question establishes the dependencies between the question words. Since the parse tree is produced by a probabilistic parser similar to the one reported in Collins (1996), each binary syntactic dependency is assigned a probabilistic weight, thus inducing an order on the list of keywords used for retrieval. This ordered list is used by *FALCON*'s paragraph retrieval module. The module is an extension of the *SMART* IR engine (Salton 1969), modified in two major ways: (1) Boolean operators were added, e.g. AND, OR; (2) document paragraphs rather than full documents were retrieved. The well-known disadvantage of Boolean retrieval – its imprecision – was handled by dropping some of the keywords when the search space became too restrictive and too few paragraphs were returned; or, by adding keywords when the search space was too broad and too many meaningful paragraphs were found. This process of adding or dropping keywords until either an acceptable number of paragraphs are retrieved or the entire list of keywords has been processed

https://doi.org/10.1017/S1351324903003176 Published online by Cambridge University Press

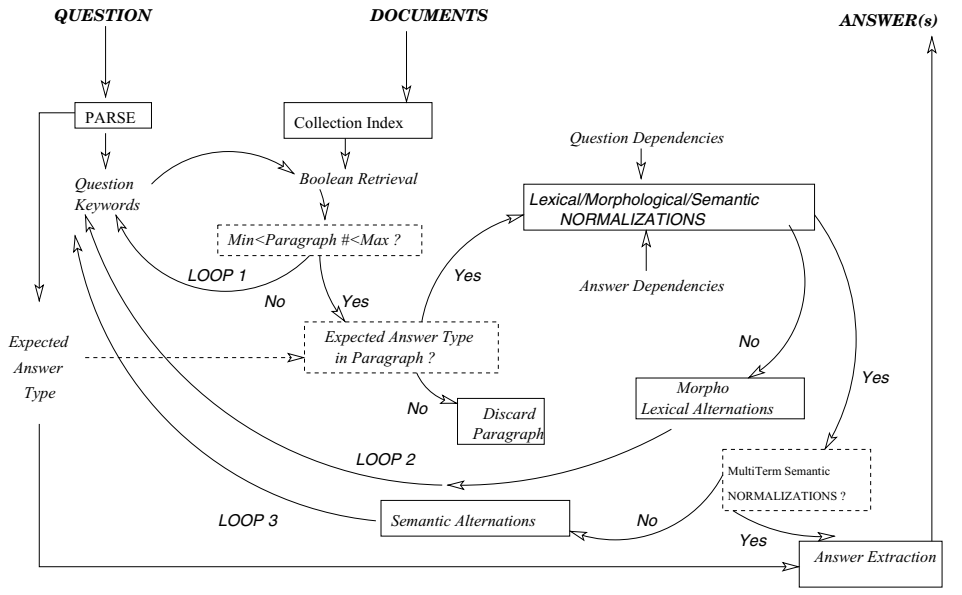
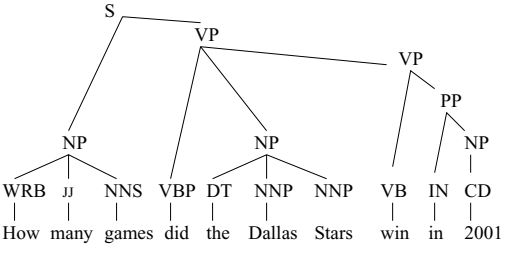


Fig. 9. Lexico-semantic feedbacks for Q/A.

constitutes the first feedback loop in the *FALCON* retrieval mechanism (figure 9). The minimum and maximum number of paragraphs depends on the size of the document collection. For the TREC collection, we determined empirically that this number should not exceed 500.

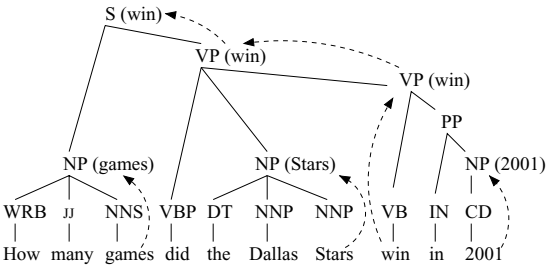
Paragraphs that do not contain the expected answer type are discarded. The remaining paragraphs are parsed and lexico-semantic unifications are tried. By implementing our own version of the publicly available Collins parser (Collins 1996), we also learned a *dependency* model that enables the mapping of parse trees into sets of binary relations between the head word of each constituent and its sibling words. For example, the parse tree of the question *"How many games did the Dallas Stars win in 2001?"* is:



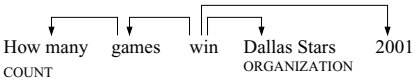
For each possible constituent in a parse tree, rules first described in Magerman (1995) and Jelinek, Lafferty, Magerman, Mercer, Ratnaparkhi and Roukos (1994) identify the head child and propagate it to its parent. For the parse tree illustrated

https://doi.org/10.1017/S1351324903003176 Published online by Cambridge University Press

below the propagation is:



When the propagation is completed head-modifier relations are extracted generating the following dependency structure, called the *question semantic form* in Harabagiu *et al.* (2000). Some of the lexical expressions from these dependency structures are also assign semantic labels by the Named Entity recognizer (e.g. COUNT or ORGANIZATION):



A similar dependency structure is generated from each paragraph containing the keywords *win*, *Dallas Stars*, *games* and *2001*. The semantic/dependency structures of questions and answers are unified by matching both the structure and the lexical labels. For example, the paragraph containing the snippet *"Dallas Stars lost ten games in 2001 and won twelve"* would produce a successful unification. However, seldom is the case when the correct answer can be unified with the question without allowing any form of paraphrase of the information. For this reason, we allow lexical, morphological and semantic variations of the keywords to be able to detect the answer in snippets like *"the 12 victories of the Dallas Stars in 2001 were surprising"* or *"Dallas Stars defended other teams 12 times in 2001"*. For this reason, whenever the unification between questions and paragraph snippets is unsuccessful the search for new relevant paragraphs begins anew by replacing question keywords with some of their morphological and lexical alternations. This is the second feedback loop. A third feedback loop takes place when semantic alternations are allowed in the normalization of the question keywords. Depending on the forms of linguistic knowledge employed, the alternations used in feedback lops two and three can be classified as:

- *Morphological alternations.* Depending on the specificity of the question keyword that determines the expected answer type, we access all the morphological derivations available in WordNet. For example, in the case of question *Q209: Who invented the paper clip,?* we allow all the morphological alternations of the verb *invented*. For this question, the verb was mapped to its nominalized form, *inventor*, which is in the sub-hierarchies of the answer

https://doi.org/10.1017/S1351324903003176 Published online by Cambridge University Press

type PERSON. Therefore, we input to the retrieval engine the query:

QUERY(Q209):[*paper AND clip AND (invented OR invent OR inventor OR inventing)*]

- *Lexical alternations.* Since WordNet encodes at most seven types of semantic relations per concept, synonyms together with a wealth of other semantic information can be easily mined. Such alternations improve the recall of the answer paragraphs. For example, in the case of question *Q221: Who killed Martin Luther King,?* by using the synonym for *killer* – the noun *assassin* – the system retrieved paragraphs with the correct answer. For the question *Q206: How far is the moon,?* since the adverb *far* is encoded in WordNet as being an attribute of *distance*, adding the noun *distance* to the retrieval keywords produces the correct answer.
- *Semantic alternations.* Mining semantic knowledge from WordNet that is not always localized in the conceptual synset provides semantic alternations. An example in question *Q258: Where do lobsters like to live,?* since in WordNet the genus of the definition of the verb *prefer* is *liking better*, the query becomes:

QUERY(Q58):[*(lobster OR lobsters) AND (like OR prefer)*]

In this way the likelihood of retrieving the correct answer is greatly enhanced.

2.2 Reading comprehension systems

Q/A-based reading comprehension systems locate answers in a single document as a result of implementing the following steps:

1. *Question processing* proceeds according to the extraction of information content from each question. Possible representations of the information expressed in each of the five questions associated with every story range from bag-of-words approaches (e.g. Hirschman *et al.* 1999)) to a full parse of the question (e.g. Riloff and Thelen 2000) combined with recognition of named entitysemantic information (figure 10).
2. *Story processing* consists of (a) considering all coreference links annotated in the text, and (b) recognizing the semantic categories of named entities from the text.
3. *Answer searching* focuses on the identification of information encoded in the wording of the question and matching this against information from the document. The search is implemented as a pattern matching (bag-of-words) augmented with additional linguistic processing (e.g. stemming, name alias identification, semantic class recognition).

In the AQUAREAS RC Q/A system (Ng *et al.* 2000), the search is based on a machine learning approach that has two steps. First, a set of features is devised to capture information that helps to distinguish answer sentences from non-answer

https://doi.org/10.1017/S1351324903003176 Published online by Cambridge University Press

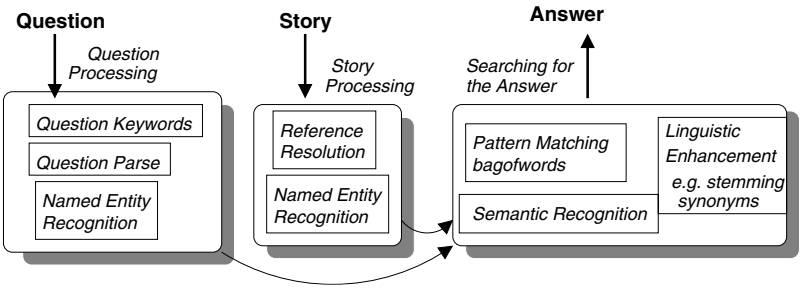


Fig. 10. Architecture of a reading comprehension system.

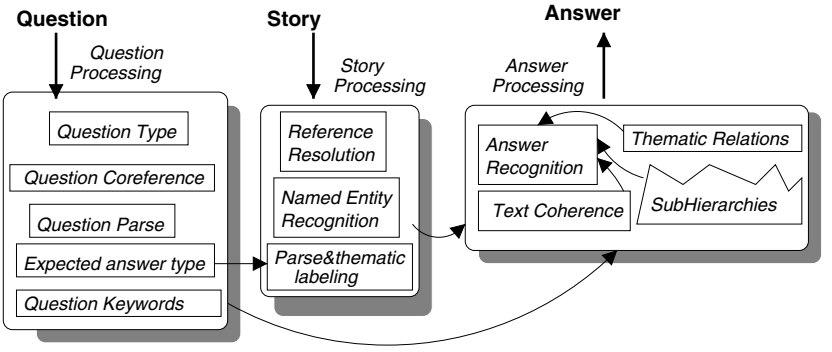


Fig. 11. Reading comprehension based on expected answer type.

sentences. Then a classifier is learned for identifying the answer of each question type from training examples. A different enhancement of the typical RC Q/A architecture was implemented in the QUARC system (Riloff and Thelen 2000). For each type of question a set of rules is implemented. Each rule awards a specific number of points to each sentence as a measure of confidence that it has found the answer. The sentence with the highest number of points is selected as the answer.

None of the current RC Q/A architectures use the notion of expected answer type, first implemented in the *LASSO* IS Q/A system (Moldovan *et al.* 1999) and then further extended in *FALCON* (Harabagiu *et al.* 2000b). When the expected answer type is recognized, the *FALCON* IS Q/A system identifies correct answers with more than 60% precision (Pasca and Harabagiu 2001). We argue that the same observation holds true for RC Q/A systems too, thus we have implemented an RC Q/A architecture that performs answer recognition based on the expected answer type during question processing. The architecture of our RC Q/A system is illustrated in figure 11.

In the system illustrated in figure 11, each question is parsed and the expected answer type is identified in the same way as in *FALCON*. Question keywords are also extracted in the same way. Since some questions contain pronouns, reference resolution must be carried out during the question processing phase. We resolve all question anaphors with the algorithm reported in Harabagiu, Moldovan, Paşca, Mihalcea, Surdeanu, Gîrju, Rus, Lăcătuşu, Morarescu and Bunescu (2001). We also

https://doi.org/10.1017/S1351324903003176 Published online by Cambridge University Press

recognize the question type by the question stem and use the question type for better recognizing the expected answer type. For example, *who* questions are asking about people, animals or organizations. The answers are either (a) subjects of the events mentioned in the question; (b) names of people having the properties expressed in the question; or (c) antecedents of pronouns collocated in the same sentence with some of the question keywords or with concepts semantically related to them.

Story processing is enhanced with parsing and thematic labeling. Answer recognition is fundamentally different in our system, called *FALCON-RC* from other RC Q/A systems. Answers are identified as a result of combining (1) semantic information derived from taxonomies we have developed for the RC Q/A task with (2) thematic relations recognized from the parse of the story and (3) cues of text coherence. Text coherence is especially important for processing *why* questions. *Why* questions are a special case in that they necessitate the recognition of causes, motivations and reasons of actions and events. The coherence structure of a text highlights such causality relations. Often, discourse cue phrases, such as *because* or *since*, are strong indicators of causality relations.

The main architectural difference between *FALCON* and *FALCON-RC* is derived from the need for identifying text paragraphs from large collections of documents in the case of the information seeking task and the need for selecting a sentence from a short story text in the case of reading comprehension. Furthermore, while document processing concentrates on paragraph retrieval in the former instance, reference resolution throughout the entire text is paramount in the latter application. One may assume that these processing differences dictate the ways in which questions are handled and answers are extracted. However, from our experiments, we conclude that there are many similarities between the question processing and answer extraction techniques implemented in *FALCON* and *FALCON-RC* and, aside from the previously mentioned main architectural difference, other differences are a matter of degree in terms of the amount of processing resources each application should use.

3 Question processing

The role of question processing is to (1) determine the type of the question; (2) identify the expected answer type; (3) transform the question into one or several keyword-based queries that identify the answer paragraph or sentence; and (4) identify the lexico-semantic and discourse relations between the expected answer type and the question keywords used to locate the answer. A strong indicator of the question type is provided by the question stem, e.g. *who*, *when*, etc. Most of the question stems are ambiguous; therefore, additional mechanisms of identifying the expected answer type are necessary.

The 893 test questions evaluated in TREC-8 and 9 share the property that there is at least one answer in the text collection. For the most part, this is true for the reading comprehension test as well. The questions for the two tasks, however, have different properties. The TREC questions employ eight different question stems that are unevenly distributed (see figure 12). The 565 questions provided for the Remedia stories are equally distributed among the five traditional question

https://doi.org/10.1017/S1351324903003176 Published online by Cambridge University Press

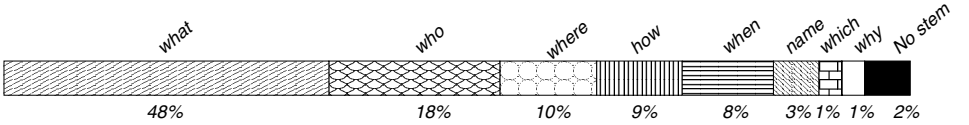


Fig. 12. Distribution of question stems in the TREC evaluation questions.

stems: who_1, $what_2$, $when_3$, $where_4$ and why_5. Therefore, except for 2% of the TREC questions that have no question stem, for virtually all the TREC and REMEDIA questions processing is based on the information cued by their stem. To classify questions according to their stems, most TREC Q/A systems developed rules for associating question stems with categories; named entity recognizers were often used to support this. In contrast, our reading comprehension system associates the question stem with a thematic feature that needs to be recognized in the text. Figure 13(a) represents the associations used for the TREC Q/A systems, whereas figure 13(b) presents associations employed for reading comprehension. From figure 13 it is clear that for TREC questions, the stems induce a classification based on conceptual classes such as PERSON and ORGANIZATION. In the case of REMEDIA questions, the stems are mapped into thematic roles, such as SUBJECT, OBJECT, REASON, like those defined in FRAMENET (Baker, Fillmore and Lowe 1998) or some categories like PARENT or REWARD. Some of the answer categories are identical with those employed by the IS Q/A systems, but most of them are new, less abstract categories.

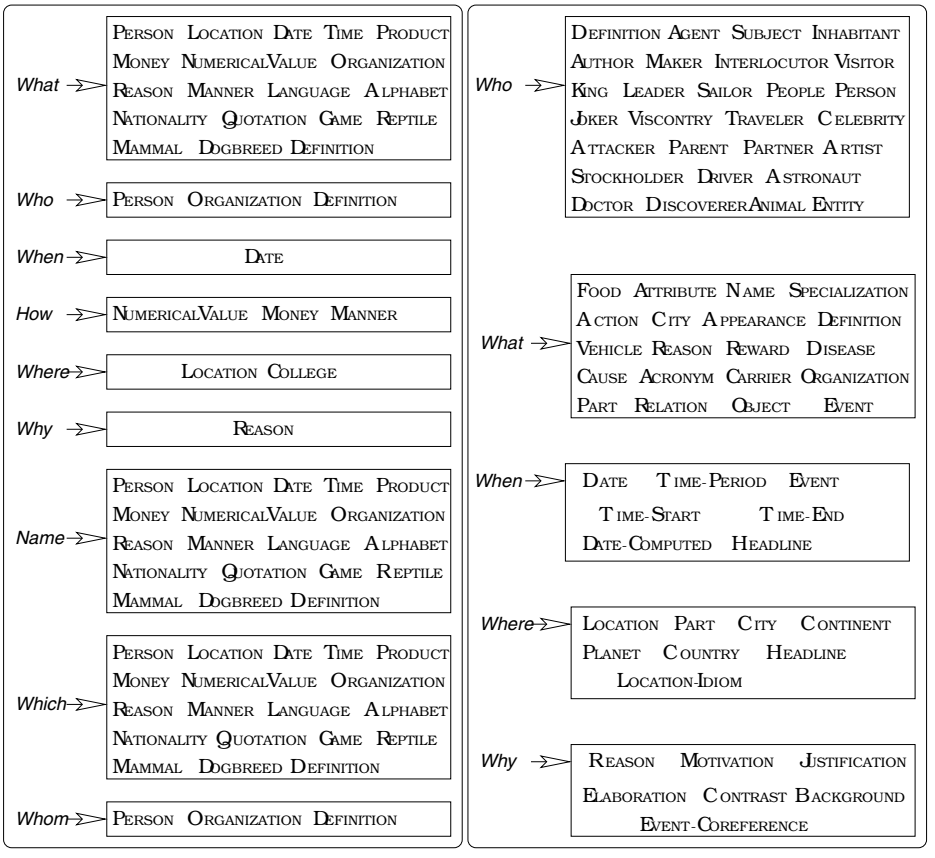
3.1 Question stems

As can be seen from figure 13, most of the question stems are highly ambiguous; that is, the stems map to many different conceptual classes or thematic roles. For example, questions starting with the stem *what* can ask about almost anything. To recognize the expected answer type, one of the question concepts must be selected and taken into account. For example, in the case of TREC question *Q3: What does the Peugeot company produce?*, the verb *produce* elicits a response for some PRODUCT. To recognize the appropriate question concept that maps to the correct *expected answer type*, we have devised a methodology that is based on three forms of knowledge: (1) the *dependency structure* of the question; (2) mappings from question stems to possible answer types that are built off-line; and (3) answer taxonomies that link the answer types to WordNet subhierarchies that cover a large percentage of the English noun and verb lexemes. Figure 13 illustrates mappings between question stems and possible answer types used for both IS Q/A and RC Q/A.

Given the three forms of knowledge acquired from the off-line analysis of large question logs available at EXCITE, we have devised the following procedure to select the expected answer type:

1. Determine $\{A\}$, all the categories corresponding to the question stem (see figure 13).

https://doi.org/10.1017/S1351324903003176 Published online by Cambridge University Press



(a) TREC questions (b) REMEDIA questions

Fig. 13. Mappings from question stems to answer types.

2. Select N the node from the dependency structure of the question that:
   (a) is connected to the question stem;
   (b) has the highest number of dependencies among all lexemes connected to the question stem.
3. Search for the node N along all answer hierarchies subsumed by $\{A\}$.
4. Return the answer type as the top of the hierarchy where N was located.

For example, figure 14 illustrates the entire dependency structure of a TREC question which indicates that *manufacture* has the largest number of connected concepts. Therefore, the node *manufacture* is searched in the WordNet subhierarchy linked to any of the concepts from $\{A\}$, mapped from the question stem. The node *manufacture* is found in the subhierarchy of PRODUCT, which becomes the expected answer type of the question. As reported in Harabagiu *et al.* (2001), the set of possible answer classes for the IS *FALCON* Q/A system as well as their connection to WordNet subhierarchies was performed manually by creating an off-line answer taxonomy. Moreover, whenever the expected answer type is discovered, it substitutes the question stem in the dependency structure, enabling unifications with candidate

https://doi.org/10.1017/S1351324903003176 Published online by Cambridge University Press

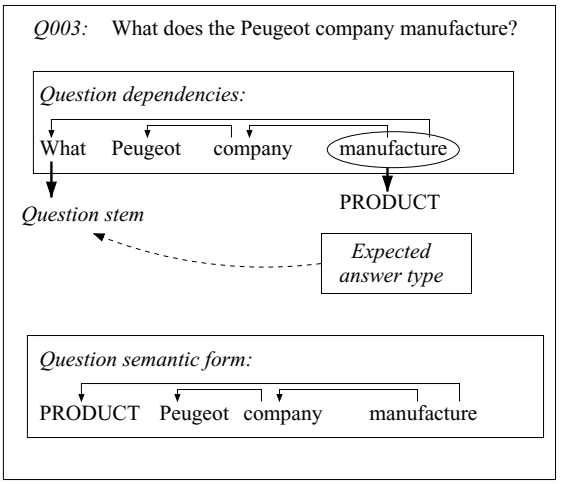


Fig. 14. Disambiguation of question stems.

answers. Figure 14 illustrates the result of the substitution of the question stem with the expected answer type.

For REMEDIA questions, this form of question stem disambiguation did not perform as well as for TREC questions. Alternatively, in the *FALCON-RC* Q/A architecture we associated question stems with *thematic roles* similar to those encoded in the FrameNet project (Baker *et al.* 1998)[6]. Gildea and Jurafsky (2000) proposed a thematic labeling scheme that is based on the same parser (Collins 1996) that we employed both for our IS Q/A architecture and for our RC Q/A architecture. We trained a specialized theme-labeler for the REMEDIA questions which employed several additional features such as (1) the question stem, (2) the phrase type, (3) the head word, (4) the verb voice and WordNet class, (5) the dependency structure of the question/answer, and (6) the frame elements of the verb. We employ the same training and testing techniques as those reported in Gildea and Jurafsky (2000). We trained the theme labeler on 30% of the REMEDIA stories and questions and tested on the rest of the stories. The results of the theme recognition were comparable with those reported by Gildea and Jurafsky (2000) because we employed the same parser, namely the Collins parser (Collins 1996).

Figure 15 illustrates a REMEDIA question and the corresponding labeling of its parse tree with thematic roles encoded in FrameNet. In the example shown in figure 15, the verb *place* has two *goal-location* relations: (a) to the question stem *where* and (b) to the prepositional attachment *"around the country."* Since it is unusual to have a repetition of the same thematic label, it is assumed that one of the thematic selections includes the other one. In the case of the example illustrated

[6] A semantic frame is a script-like structure linking by linguistic convention the meanings of a lemma to several other linguistic items. Each frame identifies a set of *frame element* having the function of thematic roles for the frame lemma, which is typically a verb or a nominalization.

https://doi.org/10.1017/S1351324903003176 Published online by Cambridge University Press

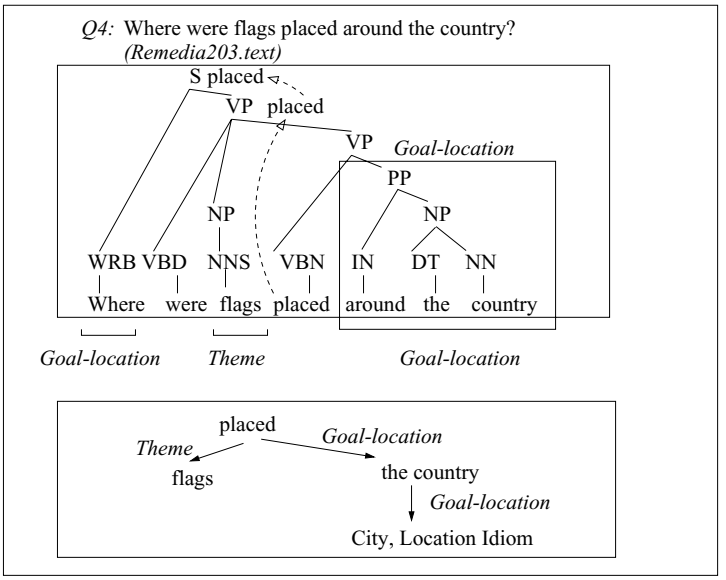


Fig. 15. Disambiguation of question stems.

in the Figure 15, the location of the flags is included somewhere in the country, thus the expected answer types allowed are only CITY or some idiomatic expressions that pertain to locations (see figure 13). The mapping into thematic roles is a form of semantic analysis less shallow than the one provided by the dependency structures in the TREC example since it assigns labels to the binary relations and moreover these labels have a semantic functionality. As defined in Gildea and Jurafsky (2000), thematic roles represent participants in an action or relationship formalized in a semantic frame organized around a lemma lexicalizing the action/state. For the REMEDIA question illustrated in figure 15, the stem *where* indicates a special form of location – one manifested by the semantic role *Goal* in the frame of the verb *place*. This thematic role was defined in FRAMENET (Baker *et al.* 1998). Since the expected answer type is disambiguated to either a CITY or a LOCATION-IDIOM, it is recognized in the prepositional attachment *"in every school"* from the sentence *"He asked that a flag be placed in every school"* and is therefore the correct answer. For REMEDIA questions, some of the tops of the semantic hierarchies representing the answer type are theme labels. This is different from the TREC case, where we saw no need for such answer types.

3.2 Expected answer types

For TREC questions prior recognition of the expected answer type is dependent upon having an ANSWER TAXONOMY, which we generated off-line in the following way:

Step 1 We devise a set of top categories modeled after the semantic domains encoded in the WordNet database containing 25 noun categories and 15

https://doi.org/10.1017/S1351324903003176 Published online by Cambridge University Press

Table 1. *Named entity categories*

date	*time*	*organization*	*city*	*product*	*price*
country	*money*	*human*	*disease*	*phone number*	*continent*
percent	*province*	*other location*	*plant*	*mammal*	*alphabet*
airport code	*game*	*university*	*dog breed*	*number*	*quantity*

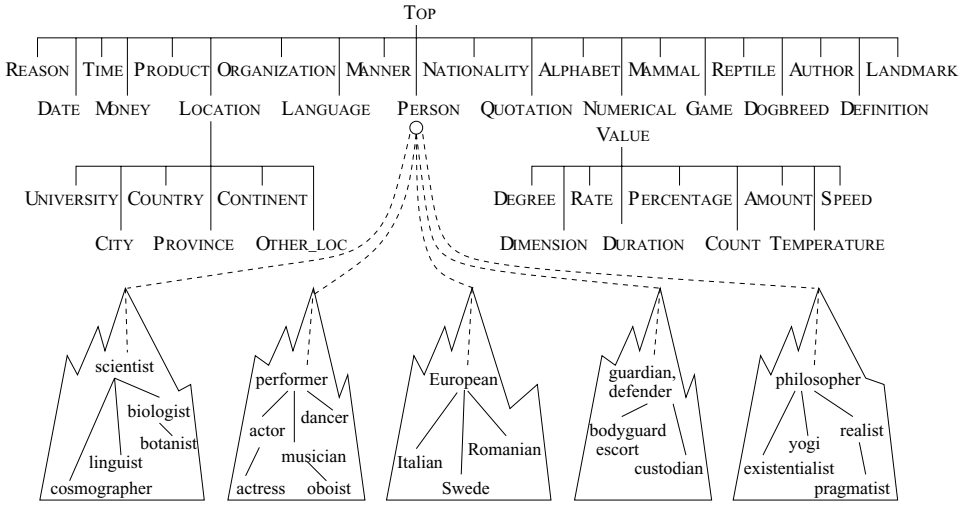


Fig. 16. Examples of tops from the ANSWER TAXONOMY.

verb categories. The top of each WordNet hierarchy was manually inspected to select the most representative nodes and add them to the tops of the ANSWER TAXONOMY. Furthermore, we added open domain named entity-type semantic categories. For example, Table 1 lists the named entity categories we have considered in our experiments. Many of the tops of the ANSWER TAXONOMY are further categorized as illustrated in figure 16. We included 33 concepts in all as tops for the taxonomy.

Step 2 The more refined categorization of the top ANSWER TAXONOMY generates a many-to-many mapping of the Named Entity categories in the tops of the ANSWER TAXONOMY. For example, see figure 17.

Step 3 Each leaf from the top of the ANSWER TAXONOMY is connected to one or several WordNet sub-hierarchies. Figure 16 illustrates a fragment of the ANSWER TAXONOMY comprising several WordNet sub-hierarchies.

The *Answer Type* categories listed as tops of the ANSWER TAXONOMY are too general for the RC Q/A system tested on the REMEDIA stories. For example, the expected answer type for *who* questions is PERSON, but too often several people are mentioned in the same story and, therefore, accuracy suffers. For this reason, we instead expect that the subject of the event mentioned in a *who* question will be the appropriate response and, similarly, the object of the verb in a *what* question will be the right answer.

https://doi.org/10.1017/S1351324903003176 Published online by Cambridge University Press

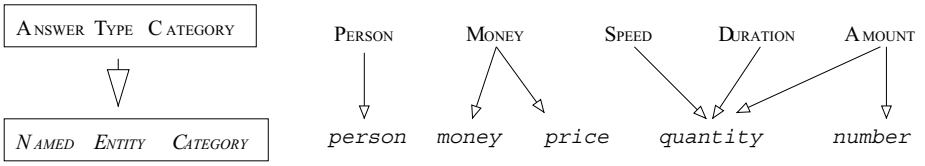


Fig. 17. Answer types named entity mappings.

1. *Who left Joplin on October 26, 1861?*	*The last Pony Express rider leaves town today.*
2. *Who wrote the "Pledge of Allegiance"?*	*The pledge was written by Frances Bellamy.*
3. *Who kills Florida panthers on purpose?*	*But hunters still find them and hunt them.*
4. *What did Alex eat on the island?*	*He ate plums, cray fish, peppers and turnips.*
5. *What will the spacecraft shoot at the moons?*	*It will point laser beams at the surface and shoot.*

Fig. 18. Examples of answers that are subjects or objects of the question verb.

Figure 18 illustrates five questions and their corresponding answers, the exact answer being underlined. The subject precedes the verb in the answer to the first question since the answer is in active voice. By contrast, however, the answer of second question is in the passive voice so the object indicated by the preposition *by* is the correct answer. Recognizing verbal phrase coordination in the answer of the third question as well as resolving the pronoun *them* to *panthers*, enables identification of *hunters* as the subject by inference. The answer to the fourth question is obtained by recognizing noun phrase coordination. Deducing the answer to question five is done by verbal phrase coordination and object-ellipsis inference. In addition to resolving ellipsis through recognizing verbal phrase coordination, pronominal coreference also needs to be resolved.

When mapping REMEDIA questions to the answer types listed in figure 17 and to syntactic roles, e.g. *subject*, *object*, we need to incorporate in the *FALCON-RC* Q/A system more language processing resources than in the IS Q/A system. These resources include reference and ellipsis resolution methods as well as phrase coordination recognizers derived from the parse of sentences. Moreover, for both Q/A systems several additional cases arise. For these cases we make the following observations:

1. The question seeks a named entity or the name of a concept when WordNet synonyms of the verb *name* or its paraphrases are identified in the question. We have added the top NAME to our ANSWER HIERARCHY, as shown in figure 19(a). Note that the hierarchy contains not only nouns, but also verb sub-hierarchies. Figure 19(b) illustrates some questions that require this new top and its corresponding sub-hierarchy.
2. When the question seeks the definition of a concept, the question is matched to some predefined patterns, such as those listed in figure 20(a). A definition question requires special treatment even for TREC questions. We match

https://doi.org/10.1017/S1351324903003176 Published online by Cambridge University Press

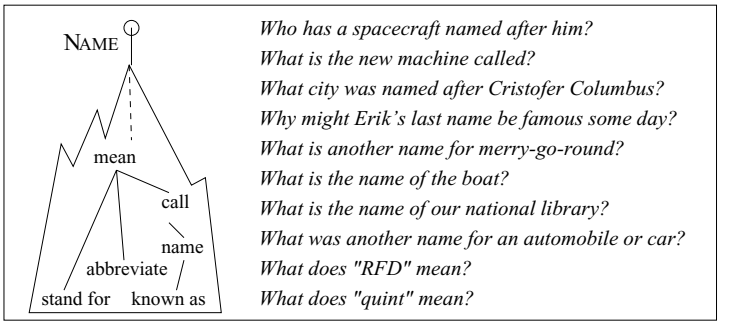


Fig. 19. A novel answer type.

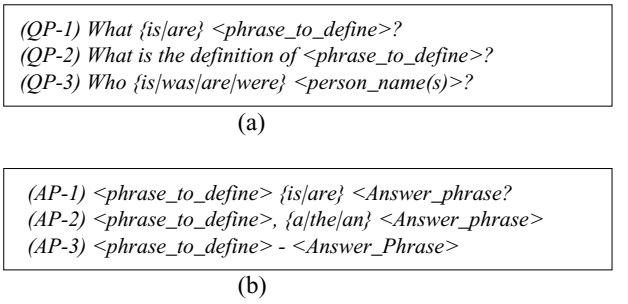
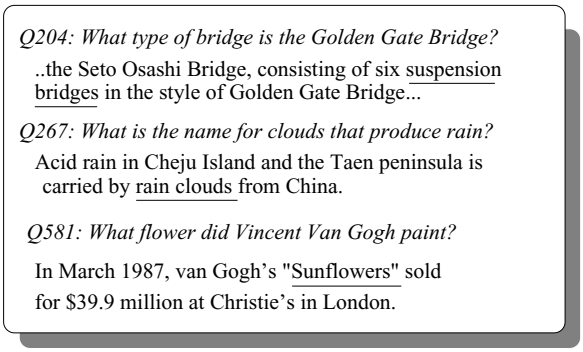


Fig. 20. Patterns of questions and answers that express definitions.

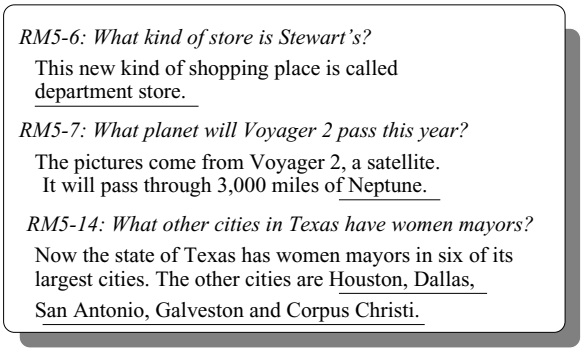
candidate answers against a similar set of patterns where the exact answer is given by the head of <*Answer_phrase*> shown in figure 20(b).

3. The question asks about an event when it specifies the subject, object or other thematic relations of the event. Such questions are recognized when the main verb *do* is used in the question, e.g. "What did the French people *do* in 1789?" or "What was she the first woman to *do*?".
4. Questions asking about distinct attributes of an entity are indicated by the presence of words *kind*, *type*, or *name*. Figure 21 lists a set of such TREC and REMEDIA questions and their corresponding answers. In this case, the answer type is determined from the WordNet sub-hierarchy headed by the concept attached to *kind*, *type* or *name* in the question. In this situation the *expected answer type* does not belong to the top of the ANSWER TAXONOMY, but it is rather dynamically created by the interpretation of the dependency graph. For example, the dynamically created *bridge*, generated for *Q204* in figure 21, contains 14 member instances, including *viaduct*, *rope bridge* and *suspension bridge*. Similarly, question *Q581* generates a dynamic expected answer type *flower*, with 470 member instances, comprising *orchid*, *petunia* and *sunflower*. For dynamic categories all member instances are searched in the text passages.

https://doi.org/10.1017/S1351324903003176 Published online by Cambridge University Press



(a) TREC questions



(b) REMEDIA questions

Fig. 21. Questions seeking instances of special concepts.

4 Extracting answers from texts

4.1 Background

Several different methods of extracting answers have been implemented in the current IS Q/A systems. These methods share some similarities with the RC Q/A search techniques, but also have differences. The first method, implemented in TexTract (Srihari and Li 2000) performs a text matching of the question template with the processed documents. Two different rankings are used to evaluate the matching. First, a count of the unique question keywords in each document sentence is made. Then, a secondary ranking takes place to account for variants or alternations of the question keywords. This method is similar to the DeepRead pattern matching of words and the result of word stemming.

The second method is answer selection with maximum entropy implemented in the system described in (Ittycheriah *et al.* 2001). This method is based on five different distances: (a) matching words, which is a TF-IDF sum of the words from the question and document sentence that matched in the morphological space; (b) thesaurus match, measuring the TF-IDF of all words that are synonymous matches in WordNet; (c) mismatch words, measuring the TF-IDF of all words that did not match; (d) dispersion, counting the number of question words that

https://doi.org/10.1017/S1351324903003176 Published online by Cambridge University Press

are matched in a; and (e) cluster words, counting the number of words from the candidate sentence that occur adjacently both in the sentence and in the question. The RC Q/A system described in Charniak *et al.* (2000) has implemented only the TF-IDF distance of matching words, thus it could be extended to implement the other measures as well.

The third method is the predictive annotation technique implemented by IBM's IS Q/A system (Radev *et al.* 2000). Answer extraction is performed by the *Ansel* and *Werlect* algorithms, based on logistic regression for ranking potential answer using a training set with seven features. Next, we present the method used by our systems, *FALCON* and *FALCON-RC*, respectively.

4.2 Extracting answers from the TREC document collection

This section introduces an alternative answer extraction method based on simple machine learning technique, also exhibited in *FALCON* (Harabagiu *et al.* 2000b). The method learns a *comparison* function between potential candidate answers based on seven features and the expected answer type. We found that the same method when trained with a different set of features, performed very well in our implementation of an RC Q/A system, *FALCON-RC*.

The fact that questions are not restricted to a number of predefined domains is a challenging feature of modern textual Q/A systems. Developing a machine learning approach for open-domain Q/A has several advantages: (1) it eliminates the need for a knowledge engineer to craft the rules that extract answers; and (2) it can scale up to a large number of new, unseen questions and adapt new knowledge for their resolution. We incorporated in our system a machine learning approach for answering TREC questions. We used the 200 TREC-8 questions for training and tested the resulting procedure on the 693 TREC-9 questions. Our learning technique is based on the observation that the results of multiple feedback retrieval is always a set of paragraphs in which at least one paragraph contains the correct answer. Typically, the cardinality of the set of paragraphs is between 500 and 3000 elements. Any sorting algorithm, e.g. *quicksort* can order this set of paragraphs if a *comparison function* is provided. The goal of the TREC Q/A evaluations is to return five ordered text snippets that represent the most likely answers to any given question. Therefore, we need to sort all the paragraphs and return the text snippets extracted from the first five paragraphs.

Answer extraction is performed in three steps: (1) we learn off-line a comparison function for answer ranking; (2) we apply the comparison function at test time, while sorting a set of unseen paragraphs; (3) we select only the first five paragraphs from which we extract 50 bytes (for short answers) or 250 bytes (for long answers) centered around a concept of the same semantic category as the *expected answer type*.

To learn the comparison function we have experimented with numerous possible features and obtained the best results with the seven features listed in figure 22(a). The learning technique we employ is the *perceptron* that associates the input vector of the features $\vec{f}$ with an output function o by first adding the weighted features and then applying a signature function. Since the comparison function needs only to

https://doi.org/10.1017/S1351324903003176 Published online by Cambridge University Press

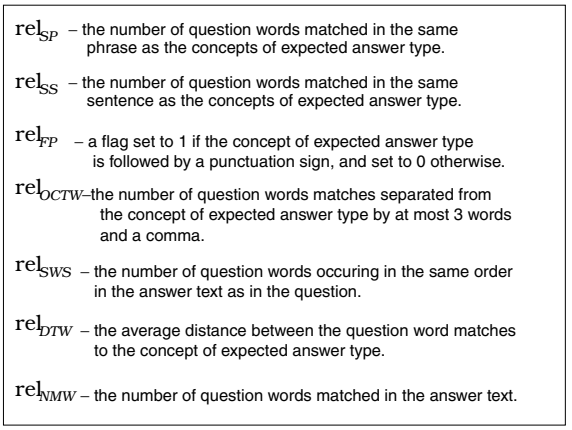


Fig. 22. Learning features for Answer Extraction.

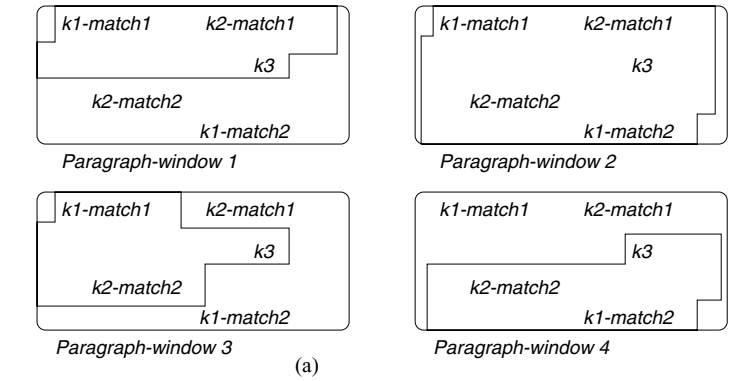
indicate which element precedes the other in the comparison, therefore, producing only two possible values, we find the perceptron an ideal vehicle for learning comparison functions between paragraphs. Indeed, our experiments with decision trees did not perform as well.

In the training phase, we annotate the paragraphs containing the correct answer for each TREC-8 question. The paragraphs used in the training phase are produced automatically by the IR component of the Q/A system. We train the perceptron on an equal number of positive and negative examples. For training purposes, we compare pairs of paragraphs in which at least one of the paragraphs contains the exact answer. Whenever the first paragraph is the one containing the correct answer, we have a positive example; similarly a negative example is generated by considering the paragraph containing the answer to be the second paragraph of the pair. For each pair of paragraphs (P_1, P_2), we compute $\Delta rel_{SP} = rel_{SP}^{P_1} - rel_{SP}^{P_2}$; $\Delta rel_{SS} = rel_{SS}^{P_1} - rel_{SS}^{P_2}$; $\Delta rel_{FP} = rel_{FP}^{P_1} - rel_{FP}^{P_2}$; $\Delta rel_{OCTW} = rel_{OCTW}^{P_1} - rel_{OCTW}^{P_2}$; $\Delta rel_{SWS} = rel_{SWS}^{P_1} - rel_{SWS}^{P_2}$; $\Delta rel_{DTW} = rel_{DTW}^{P_1} - rel_{DTW}^{P_2}$; and finally $\Delta rel_{NMW} = rel_{NMW}^{P_1} - rel_{NMW}^{P_2}$. The goal of training the perceptron is to obtain the value of the weights w_i and of the *threshold* from the formula: $rel_{pair} = w_{SP} \times \Delta rel_{SP} + w_{SS} \times \Delta rel_{SS} + w_{FP} \times \Delta rel_{FP} + w_{OCTW} \times \Delta rel_{OCTW} + w_{SWS} \times \Delta rel_{SWS} + w_{DTW} \times \Delta rel_{DTW} + threshold$.

The perceptron learns the seven weights as well as the value of the threshold used for future tests on the remaining 793 TREC-9 questions. We obtained the following values for the seven weights: $w_{SWS} = 12.45$; $w_{FP} = -4.41$; $w_{OCTW} = 3.16$; $w_{SP} = 4.46$; $w_{SS} = 22.14$; $w_{NMW} = 42.28$; $w_{DTW} = -49.97$. The learned value of the *threshold* is -15.05.

At the test phase, given any pair of paragraphs, when the value of the resulting rel_{pair} is positive, we select the first paragraph, otherwise we chose the second one. In addition, we found that prior to answer extraction, the ordering of the paragraphs has a significant effect on the overall performance of the Q/A system. Moreover, when the expected answer type cannot be identified, many of the features used to learn the comparison function cannot be used since they depend on the expected

https://doi.org/10.1017/S1351324903003176 Published online by Cambridge University Press



(a)

rel_{SWS} - computes the number of words from the question that are recognized in the same sequence in the current paragraph window.

rel_{DAW} - represents the number of words that separate the most distant pair of keywords in the window.

rel_{UNMW} -computes the number of unmatched keywords from the question.

(b)

$$ord_{pair} = q_{SWS} \times \Delta rel_{SWS} + q_{DAW} \times \Delta rel_{DAW} + q_{UNMW} \times rel_{UNMW} + threshold$$

(c)

Fig. 23. (a) Four different windows defined on the same paragraph; (b) features for learning comparison functions of windows; (c) comparison function formula for paragraph-windows.

answer type. To order the paragraphs we used again a perceptron, but this time we employed only three features. The definition of these three features depends on the notion of *paragraph-window*, first defined in Moldovan *et al.* (1999).

Paragraph-windows are determined by the need to consider separately each match of the same keyword in the same paragraph. For example, if we have a set of keywords $\{k1, k2, k3, k4\}$ and in a paragraph $k1$ and $k2$ are matched each twice, whereas $k3$ is matched only once, and $k4$ is not matched, we are going to have four different windows, defined by the keywords: [*k1-match1, k2-match1, k3*], [*k1-match2, k2-match1, k3*], [*k1-match1, k2-match2, k3*], and [*k1-match2, k2-match2, k3*]. A window comprises all the text between the lowest positioned keyword in the window and the highest position keyword in the window. Figure 23(a) illustrates the four windows for our example.

For each paragraph window we compute the scores listed in figure 23(b). Note that rel_{UNMW} is identical for all windows from the same paragraph, but varies for windows from different paragraphs. The formula employed by the perceptron that learns how to order paragraphs by their paragraph-window scores is listed in figure 23(c). We obtained the following values for the three weights: q_{SWS}=13.47; q_{DAW}= –163.20; whereas q_{NMW}= –11.48 and the threshold has the value 72.88. At testing time, when the relative order measure ord_{pair} is positive, the first paragraph precedes the second one, otherwise their order is reversed.

https://doi.org/10.1017/S1351324903003176 Published online by Cambridge University Press

4.3 Extracting answers from the REMEDIA stories

The procedure involved is generally the same but it relies on quite different features from answering TREC questions for the following reasons:

1. The set of question-type answer-type mappings has little relevant overlap in the two Q/A tasks (figure 13).
2. Extracting the answer from the TREC document collection involves ranking a large number of candidate paragraphs that come from different documents whereas, in the case of REMEDIA stories, the answer is selected from at most three sentences contained in a short text as we discovered in our experiments.
3. Q/A-based text comprehension requires a finer degree of answer recognition than the Q/A-based information seeking characterized by TREC. Therefore thematic cues (e.g. prepositional attachments, discourse cues) or discourse coherence relations (especially for answering *why* questions) must be taken into account.
4. Answering REMEDIA questions is highly dependent on coreference data. In our experiments we used the coreference annotations included in the REMEDIA story collection (see figure 7).

After analyzing these differences, we devised an answer extraction approach for text comprehension that postulates the use of four features, each of which are accompanied by an additional number of possible values. Figure 24 lists the four features we have considered and their corresponding values.

The first feature is based on the semantic categories associated with question stems. For example, in the case of *who* questions, if the answer is recognized as a subject, the value of the first feature is larger than the value given when the answer is a specialization of one of the question concepts. For each type of questions associated with a story, we defined an order between the values of the FUNCTION feature. The values listed in figure 24 are illustrated as an ordered set, thus if a set has four values (e.g. for when questions), the values are 4 (for DATE, 3 (for HEADLINE), 2 (for TIMEPERIOD) and 1 (for DISCOURSECUE). The second feature uses the coreference annotations and distinguishes over four cases: (1) when the candidate answer is identified without using coreference data; (2) when the candidate answer refers to question concepts; (3) when the candidate answer is an antecedent of an anaphor; (4) when the candidate answer is an anaphor. The value of this feature are: 0 for the first case, 1 for the second, 2 for the third and 3 for the fourth case. For the example illustrated in figure 6(a), sentence A_1 is extracted as a correct answer to Q_1 because the second feature of A_1 has a higher score than the second feature of the title.

The third and fourth features model the lexico-semantic support for textual coherence and are measured by adding the *strengths* associated with each WordNet relation that occurs within a path. Is-A relations are prioritized over ENTAIL relations or MERONYM relations. Given a lexico-semantic path, the weights of all relations are added to give a value to these two features. We have considered the weight of Is-A=10; the weight of HAS-PART = the weight of HAS-MEMBER = the weight

https://doi.org/10.1017/S1351324903003176 Published online by Cambridge University Press

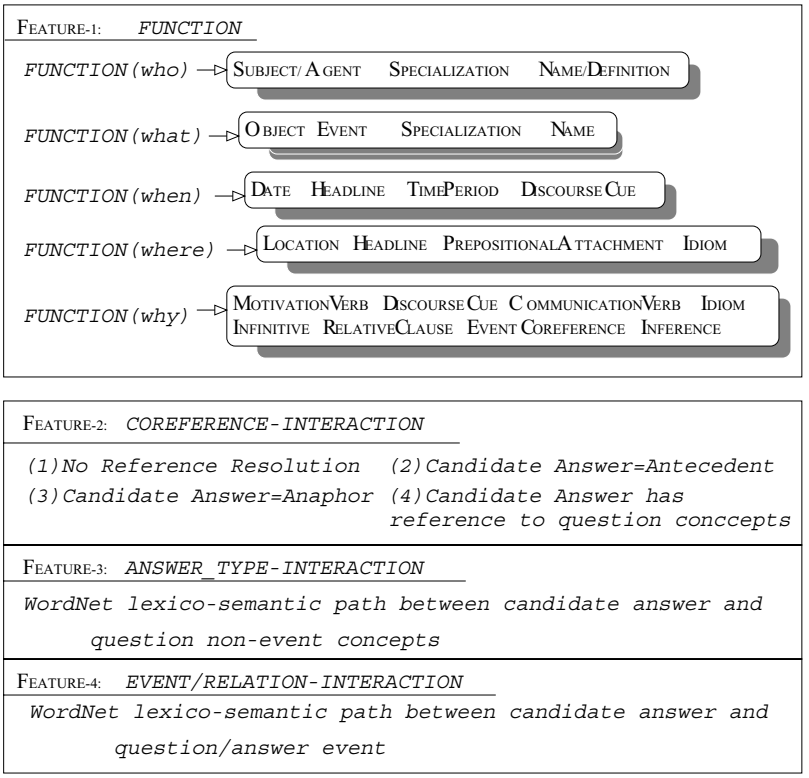


Fig. 24. Features used for extracting answers from REMEDIA stories.

of HAS-STUFF = 5; the weight of ENTAIL = 3; the weight of each GLOSS relations between a concept and any word from its defining gloss as 1.

The candidate answers are determined using the techniques described in section 3.2. Overall, we have trained a perceptron that learns an extraction function which selects the answer when multiple candidates are available. This function uses the above-mentioned four features. Although we used only four features, each of them are applicable to five REMEDIA questions types; also, for feature number one, multiple values are associated with each question type. By incorporating more lexico-semantic knowledge, we have obtained a performance that almost doubled the accuracy of the learning technique presented in Ng *et al.* (2000).

5 Evaluation

5.1 Scoring metrics

IS and RC Q/A systems cannot be evaluated with the same set of metrics. To measure the performance of an IS Q/A system, TREC evaluations compute the reciprocal value of the rank (RAR) of the highest-ranked correct answer given by the system. There are two types of answers, namely *short* (50-bytes) and *long* (250-bytes). Since only a system's first five answers are scored, if the RAR is defined as $RAR = \frac{1}{rank_i}$ its value is 1 if the first answer is correct; 0.5 if the second answer and

https://doi.org/10.1017/S1351324903003176 Published online by Cambridge University Press

not the first one is correct; 0.33 when the correct answer is the third returned answer; 0.25 if the fourth answer is correct; 0.2 when the fifth answer is correct; and 0 if none of the system's five answers are correct.

The Mean Reciprocal Answer Rank (MRAR) is used to compute the overall performance of the systems participating in the TREC evaluation using the formula $MRAR = \frac{1}{n}(\sum_i^n \frac{1}{rank_i})$. In addition, TREC-9 required that an answer be judged correct only when the document's context is relevant to the question. When the human assessors were convinced this condition was satisfied, they considered the RAR to be *strict*, otherwise, the RAR was considered *lenient*.

To measure the performance of the RC Q/A system we used two of the evaluation methods reported in Hirschman *et al.* (1999): (1) HUMSENT compares the system's response to the answer key provided by the REMEDIA publisher; and (2) AUTSENT, an automated routine which compares the system response to sentences selected based on the highest number of matching content words compared against the question. For both evaluations the comparison scores one point for an acceptable response and zero point otherwise. The score of the set of questions is the average of the scores for each question.

We have also evaluated the accuracy of *FALCON-RC* when coreference information or named entity tags are included. Experiments using human-generated coreference tags are labeled as *HumRef*; experiments employing coreference data generated by our automatic coreference resolver COCKTAIL are labeled as *AutRef*; and experiments that did not use any coreference information are labeled as *NoRef*. Similarly, experiments in which human-generated named entity categories were employed are tagged as *HumNE*; those in which the Named Entity Tagger from *FALCON* was employed are labeled as *AutNE*; and experiments that did not rely on any named entity information are labeled as *NoNE*. Figure 26 shows the results of the experiments.

5.2 Impact of feedback in the basic IS Q/A system

Figure 25(a) summarizes the MRARs provided by NIST for the *FALCON* IS Q/A system. These results were superior due to the feedback loops (see section 2.1). As reported in Pasca and Harabagiu (2001), this portion of *FALCON*'s processing had the greatest positive impact upon the system's overall performance. We wanted to discover, therefore, how *FALCON* would perform if several iterations of the feedback loops were conducted. In other words, would a ceiling be established where a tradeoff between accuracy and processing efficiency (throughput) would be encountered if the loops were repeated several times.

Figure 25(b) shows a quantitative analysis of the feedback loops which speaks to the efficiency issue. Among the three feedback loops, Loop 1 has the largest maximum number of iterations (7), as well as the largest average number of iterations (1.384). Overall, however, the average number of iterations turned out to be unexpectedly small thereby adding little processing overhead to the system.

More revealing is the qualitative analysis of the impact of the feedback loops on the TREC Q/A evaluation because this speaks to the precision issue. As figure 27

https://doi.org/10.1017/S1351324903003176 Published online by Cambridge University Press

	MRAR (lenient)	MRAR (strict)
Short answers	0.599	0.580
Long Answers	0.778	0.760

(a)

	Average Number of Iterations	Maximum Number of Iterations
LOOP 1	1.384	7
LOOP 2	1.15	3
LOOP 3	1.07	5

(b)

Fig. 25. Evaluation results for the *FALCON* IS Q/A system.

HumRef	✓		✓				✓		
AutRef		✓		✓				✓	
NoRef					✓	✓			✓
HumNE	✓	✓		✓	✓				
AutNE			✓			✓			
NoNE							✓	✓	✓
HumSent	76.4%	69.2%	72.4%	48.8%	44.4%	65.3%	59.2%	50.2%	37.4%
AutSent	60.5%	48.2%	58.8%	41.4%	38.8%	52.5%	49.6%	44.6%	31.5%

Fig. 26. Evaluation results for the *FALCON* RC Q/A system.

LOOP 1		✓		✓		✓		✓
LOOP 2			✓	✓			✓	✓
LOOP 3					✓	✓	✓	✓
MRAR (short)	0.321	0.451	0.490	0.554	0.347	0.488	0.510	0.568
MRAR (long)	0.385	0.553	0.592	0.676	0.419	0.589	0.629	0.737

Fig. 27. Effect of retrieval feedback on IS Q/A accuracy.

shows, precision increased substantially when all the loops were enabled. Individually, the effect of Loop 1 was an accuracy increase of over 40% to an MRAR of 0.451; Loop 2 produced a greater than 52% enhancement while Loop 3 produced only an 8% improvement. Figure 27 shows that when all feedback loops are enabled we obtained an MRAR of 0.568 (an increase of 76%) for short answers and 0.737 (an increase of 91%) for long answers.

5.3 Recognition of the expected answer type in the IS Q/A system

Since we hypothesized that the recognition of the expected answer type may impact the accuracy of both IS and RC Q/A systems, one aspect of our evaluation looked at the coverage of our technique for finding the expected answer type. The *coverage* measures the fraction of questions of a given category whose answer type is recognized as pertaining to that category.

Figure 28 lists the coverage and precision of the recognition for the IS Q/A system. Currently our Answer Type Taxonomy encodes 8707 concepts from 129 WordNet hierarchies covering 81% of the expected answer types. This shows that we have to continue encoding more top concepts in the taxonomy and link them to more WordNet concepts because, as figure 28 shows, our answer type coverage and Q/A precision percentages improve dramatically as more tops are added to our Answer Taxonomy.

https://doi.org/10.1017/S1351324903003176 Published online by Cambridge University Press

# Tops Answer Taxonomy	Answer Type Coverage	Q/A Precision
8	44%	42%
22	56%	55%
33	81%	78%

Fig. 28. Evaluation of the answer type recognition in the *FALCON* IS Q/A system.

# Frames	# Thematic Roles (Frame Elements)	Theme Labeler Accuracy	Answer Type Coverage	Q/A Precision
24	93	78%	44%	40%
52	188	76%	56%	51%
112	352	74%	79%	76%

Fig. 29. Evaluation of the answer type recognition in the *FALCON* RC Q/A system.

5.4 Recognition of the expected answer type in the RC Q/A system

Since the disambiguation of the question stems for RC Q/A systems depends on the recognition of thematic roles, we have also looked at the accuracy of our theme labeler and its impact on the recognition of expected answer types. We also considered the coverage of frames and the number of thematic roles/frame elements as well as the number of links between frame elements and WordNet subhierarchies. Figure 29 lists the coverage of frames and thematic roles, the accuracy of the thematic labeler and the RC Q/A precision.

For *FALCON-RC* we encoded 79 additional frames that were not currently available from FrameNet, obtaining 212 new thematic roles. All 352 thematic roles were manually linked to 216 WordNet subhierarchies, covering 78% of the expected answer types of the Remedia tested questions.

5.5 Comparison between IS and RC Q/A

To investigate further the overall generic efficacy of the IS Q/A answer extraction techniques, we applied *FALCON* directly to the Remedia stories. In the place of the large number of paragraphs retrieved from thousands of documents a la TREC, we considered all Remedia stories as the set of retrieved paragraphs. The scoring metric that we used for the evaluation was HumSent which is the percentage of test questions for which the correct answer sentence was found. This metric was originally proposed in Hirschman *et al.* (1999). The results indicate that for 300 test questions we obtained an MRAR of 37.4%. However, when we (a) changed the question and answer typing and (b) trained the perceptron for the features designed for the Remedia questions, the performance greatly improved. Since we had to adapt to learning the answer extraction in a manner similar to the one employed in the IS Q/A systems, we used a different division of texts for training and testing than the other RC Q/A systems. Similar to Hirschman *et al.* (1999), we used 60 texts for training and tested the extraction procedure on the remaining 60 texts from the Remedia collection. The overall precision was 76.4%, which is the highest score of

https://doi.org/10.1017/S1351324903003176 Published online by Cambridge University Press

	who	what	when	where	why	Overall
IS Q/A Method (FALCON)	57%	31%	43%	36%	18%	37%
RC Q/A Method (FALCON-RC)	78%	70%	77%	69%	42%	76%

Fig. 30. Breakdown of Q/A accuracy per question type for REMEDIA data.

any system working on REMEDIA questions. Figure 30 details the breakdown per question type for both experiments using the REMEDIA texts.

As we anticipated, figure 30 shows that the results of applying *FALCON* to reading comprehension are less than impressive, especially when compared to the results of *FALCON-RC* which was trained on REMEDIA texts and questions. Note that for both Q/A methods, precision was highest for *who* questions and lowest for *why* questions. In separate experiments, when applying *FALCON-RC* to the TREC-10 context questions, we obtained (1) an MRAR of 0.82 and (2) when a question in the context of a dialogue is not correctly answered, only 20% of the follow-up questions in the same dialogue are answered correctly.

Finally, we evaluated the ability of *FALCON* to answer questions in the context of a dialogue by using the 42 series of questions used in TREC-10 for the context task. We have also evaluated *FALCON-RC* on the same questions but instead of using the whole TREC collection of documents, we considered a single document. In our first experiment, this document was the one containing the paragraph from which the first answer of the first question on the series was extracted. In a second experiment we selected the document containing the paragraph that was used most frequently by *FALCON* to extract an answer for any of 42 evaluated questions.

To answer context questions, *FALCON* was enhanced by modeling the notion of dialogue context through: (1) anaphoric reference between a question and one of the questions or answers preceding it in the interaction between the user and the Q/A system; and (2) lists of keywords that are used to retrieve paragraphs containing the answer to a question that alludes to a prior question or answer. To this end we modified the question processing of *FALCON* by incorporating a reference resolution module that was described in Harabagiu and Maiorano (2002). Whenever a question Q1 contains a pronoun, the question refers to a prior question Q0 that either (a) has the expected answer type of the same semantic category as the pronouns (e.g. PERSON for he or LOCATION for there) or (b) contains a word that has a WordNet lexico-semantic relation to the anaphoric word from the question. Additionally, the pronoun *it* from idioms such as *it happened* or *it occurred* is considered to refer to the last referred event from the closest prior question.

The effect of knowing the referred question Q0 is that when searching for answers to Q1 the keywords extracted from Q0 are inserted before the keywords extracted from Q1 and used for retrieval. The keyword extraction and insertion is illustrated by two of the TREC-10 context questions, namely *"In what country was the first telescope to use adaptive optics used?"* (Q0) and *"Where in the country is it located?"* (Q1). When Q1 is processed, due to the occurrence of the pronoun *it*, the keywords

https://doi.org/10.1017/S1351324903003176 Published online by Cambridge University Press

	MRAR	Number of followup questions answered correctly
FALCON (Text snippet)	0.79	24
FALCON (Sentence)	0.79	24
FALCON-RC (Experiment 1)	0.83	29
FALCON-RC (Experiment 2)	0.85	31

Fig. 31. Results on TREC context questions.

first, *adaptive* and *telescope* from Q0 are inserted before the keyword *located* extracted from Q1. After two iterations in the first feedback loop, the keywords used for passage retrieval are *first* and *adaptive*. The first answer returned for Q1 is *"The first telescope to use 'adaptive optics' was inaugurated at La Silla in 1989"*. Without importing the keywords from the previous question Q0, it would be virtually impossible to find the answer for Q1.

Figure 31 shows the results of FALCON and FALCON-RC on the context questions evaluated in TREC-10. The evaluation of FALCON was performed by NIST human assessors, considering 50-byte long text snippets as acceptable answers. The evaluation of FALCON-RC was performed by a computational linguist student that allowed the answer to be a full sentence from the document. The same student also evaluated the answers of FALCON as full sentences containing the text snippet originally returned as the answer and did not notice any difference in accuracy. Figure 31 also shows the number of correct answers to any of the follow-up questions from the series of 42 TREC context questions. The increased number of follow-up questions answered correctly by FALCON-RC shows that, when introducing reading comprehension capabilities in an IS Q/A system, promising Q/A dialogue results can be obtained. The results also indicate that the document selected for reading comprehension has a significant effect on the overall results. Redundancy of paragraphs was a better indicator for document selection than the intuitively preferred document containing the answer to the first question.

6 Conclusion

This paper has presented two different kinds of Q/A systems: information seeking Q/A systems used for retrieving answers to natural language questions from large text collections and Q/A systems employed for reading comprehension. For IS Q/A systems combining a mechanism of identifying the expected answer type of open-domain natural language questions with a novel, multi-feedback retrieval scheme is highly successful on large collections of texts, but shows less impressive quality for reading comprehension tests applied on single texts. A different set of possible answer types as well as an answer extraction mechanism that uses a distinct set of features such as discourse cues, coreference relations, and thematic relations, is shown to be necessary for answering REMEDIA questions. The enhanced set of answer types and the ability of processing RC questions shows great promise for answering correctly follow-up questions in the context of Q/A dialogues.

https://doi.org/10.1017/S1351324903003176 Published online by Cambridge University Press

This work has addressed the systematic differences and commonalities between Q/A-based information seeking and Q/A-based reading comprehension. The paper also presents successful design principles for the two Q/A-based applications, as well as a new method for enhancing an IS Q/A system with RC capabilities by customizing (a) the identification of the expected answer type and (b) the answer extraction.

References

Allen, J., Connell, M., Croft, W. B., Feng, F., Fisher, D. and Li, X. (2000) INQUERY at TREC-9. *Proceedings 9th TExt Retrieval Conference (TREC-9)*, Gaithersburg, MD.

Abney, S., Collins, M. and Singhal, A. (2000) Answer extraction. *Proceedings 6th Applied Natural Language Processing Conference (ANLP-2000)*, pp. 296–301. Seattle, WA.

Baker, C., Fillmore, C. and Lowe, J. (1998) The Berkeley FrameNet project. *Proceedings COLING-ACL'98*.

Brill, E., Lin, J., Banko, M., Dumais, S. and Ng, A. (2001) Data-intensive question-answering. *Proceedings 10th TExt Retrieval Conference (TREC-10)*, Gaithersburg, MD.

Burke, R., Hammond, K., Kulyukin, V., Lytinen, S., Tomuro, N. and Schoenberg, S. (1997) Question answering from frequently asked question files: experiences with the Faqfinder system. *AI Magazine*, **18**(2).

Charniak, E., Altun, Y., de Salvo Braz, R., Garrett, B., Kosmala, M., Moscovich, T., Pang, L., Pyo, C., Sun, Y., Wy, W., Yang, Z., Zeller, S. and Zorn, L. (2000) Reading comprehension programs in a statistical-language-processing class. *Proceedings ANLP/NAACL 2000 Workshop on Reading Comprehension Test as Evaluation for Computer-Based Language*, Seattle, WA.

Clarke, C. L. A., Cormack, G. V. and Lynam, T. R. (2001) Exploiting Redundancy in Question Answering. *Proceedings 24th Annual International ACM SIGIR Conference on Research and Developement in Information Retrieval (SIGIR-2001)*, pp. 358–365. New Orleans, LA.

Collins, M. (1996) A new statistical parser based on bigram lexical dependencies. *Proceedings 34th Annual Meeting of the Association of Computational Linguistics (ACL-96)*, pp. 184–191. Santa Cruz, CA.

Cullingford, R. E. (1977) Organizing world knowledge for story understanding by computer. Thesis, Department of Engineering and Applied Science, Yale University. New Haven, CT.

DeJong, G. F. (1977) Skimming newspaper stories by computer. Research Report No. 104, Department of Computer Science, Yale University. New Haven, CT.

Ferret, O., Grau, B., Hurault-Plantet, M., Illouz, G. and Jacquemin, C. (2001) Terminological variants for document selection and question/answering matching. *Proceedings Workshop on Open-Domain Question Answering (ACL-2001)*, pp. 46–53. Toulouse, France.

Gaizauskas, R. and Humphreys, K. (2000) A combined IR/NLP approach to question answer ing against large text collections. *Proceedings 6th Content-Based Multimedia Information Access Conference (RIAO-2000)*, pp. 1288–1304. Paris, France.

Gildea, D. and Jurafsky, D. (2000) Automatic labeling of semantic roles. *Proceedings 38th Annual Meeting of the Association of Computational Linguistics (ACL-2000)*, pp. 512–520. Hong Kong.

Harabagiu, S. and Maiorano, S. (2002) Three ways to customize reference resolution. *2002 International Symposium on Reference Resolution*, Alicante, Spain.

Harabagiu, S., Paşca, M. and Maiorano, S. (2000) Experiments with Open-Domain Textual Question Answering. *Proceedings 18th International Conference on Computational Linguistics (COLING-2000)*, Saarbrucken, Germany.

Harabagiu, S., Moldovan, D., Paşca, M., Mihalcea, R., Surdeanu, M., Bunescu, R., Gîrju, R., Rus, V. and Morărescu, P. (2000) FALCON: boosting knowledge for answer engines. *Proceedings 9th TExt Retrieval Conference (TREC-9)*, Gaithersburg, MD.

https://doi.org/10.1017/S1351324903003176 Published online by Cambridge University Press

Harabagiu, S., Moldovan, D., Paşca, M., Mihalcea, R., Surdeanu, M., Bunescu, R., Gîrju, R., Rus, V. and Morărescu, P. (2001) The role of lexico-semantic feedback in open-domain textual question-answering. *Proceedings 39th Annual Meeting of the Association for Computational Linguistics (ACL-2001)*, Toulouse, France.

Harabagiu, S., Moldovan, D., Paşca, M., Mihalcea, R., Surdeanu, M., Gîrju, R., Rus, V., Lăcătuşu, F., Morărescu, P. and Bunescu, R. (2001) Answering complex, list and context questions with LCC's Question-Answering Server. *Proceedings 10th TExt Retrieval Conference (TREC-10)*, Gaithersburg, MD.

Hirschman, L., Light, M., Breck, E. and Burger, J. D. (1999) Deep Read: a reading comprehension system. *Proceedings 37th Annual Meeting of the Association for Computational Linguistics (ACL-99)*, pp. 325–332. College Park, MD.

Hovy, E., Gerber, L., Hermjakob, U., Lin, C.-Y. and Ravichandran, D. (2001) Toward semantics-based answer pinpointing. *Proceedings DARPA Human Language Technology Conference (HLT-2001)*, San Diego, CA.

Ittycheriah, A., Franz, M., Zhu, W. J., Ratnaparkhi, A. and Mammone, R. J. (2001) Question answering using maximum entropy components. *Proceedings 2nd Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL-2001)*, pp. 33–39. Pittsburgh, PA.

Jelinek, F., Lafferty, J., Magerman, D., Mercer, R., Ratnaparkhi, A. and Roukos, S. (1994) Decision tree parsing using a hidden derivational model. *Proceedings Human Language Technology Workshop*, pp. 272–277.

Kwok, K. L., Grunfeld, L., Dinstl, N. and Chan, M. (2000) TREC-9 Cross Language, Web and Question-Answering Track Experiments using PIRCS. *Proceedings 9th TExt Retrieval Conference (TREC-9)*, Gaithersburg, MD.

Magerman, D. (1995) Statistical decision-tree models of parsing. *Proceedings 33rd Annual Meeting of the Association for Computational Linguistics (ACL-95)*, pp. 276–283. New Orleans LA.

Lehnert, W. G. (1978) *The Proces of Question Answering*, Lawrence Erlbaum.

Moldovan, D., Harabagiu, S., Paşca, M., Mihalcea, R., Goodrum, R., Gîrju, R. and Rus, V. (1999) LASSO: a tool for surfing the answer net. *Proceedings 8th TExt Retrieval Conference (TREC-8)*, Gaithersburg, MD.

Ng, H. T., Teo, L. H. and Kwan, J. (2000) A machine learning approach to answering questions for reading comprehension tests. *Proceedings Joint Conference on Empirical Methods in NLP and Very Large Corpora (EMNLP/VLC-2000)*, Hong Kong.

Paşca, M. and Harabagiu, S. (2001) High performance question/answering. *Proceedings 24th Annual International ACL SIGIR Conference on Research and Development in Information Retrieval (SIGIR-2001)*, pp. 366–374. New Orleans LA.

Radev, D., Prager, J. and Samn, V. (2000) Ranking suspected answers to natural language questions using predictive annotation. *Proceedings 6th Applied Natural Language Processing Conference (ANLP-2000)*, pp. 150–157. Seattle, WA.

Riloff, E. and Thelen, M. (2000) A rule-based question answering system for reading comprehension tests. *Proceedings ANLP/NAACL 2000 Workshop on Reading Comprehension Test as Evaluation for Computer-Based Language*, pp. 13–19. Seattle, WA.

Salton, G. (1969) *The SMART Retrieval System.* Prentice-Hall.

Schank, R. C. (1972) Conceptual dependency: a theory of natural language understanding, *Cognitive Psychology* **3**(4), 552–631.

Schank, R. C. and Abelson, R. P. (1977) *Scripts, Plans Goals and Understanding*. Lawrence Erlbaum.

Srihari, R. and Li, W. (2000) A question answering system supported by information extraction. *Proceedings 6th Applied Natural Language Processing Conference (ANLP-2000)*, pp. 166–172. Seattle, WA.

Voorhees, E. (2001) Overview of the TREC 2001 Question Answering Track. *Proceedings 10th TExt Retrieval Conference (TREC-10)*, Gaithersburg, MD.

https://doi.org/10.1017/S1351324903003176 Published online by Cambridge University Press

Wilensky, R. (1976) Using plans to understand natural language. *Proceedings Annual Conference of the ACM*, Gaithersburg, MD.

Woods, W., Bookman, L. A., Houston, A., Kuhns, R. J., Martin, P. and Green, S. (2000) Linguistic knowledge can improve information retrieval. *Proceedings 6th Applied Natural Language Processing Conference (ANLP-2000)*, pp. 262–267. Seattle, WA.

Woods, W., Green, S. and Martin, P. (2000) Halfway to question answering. *Proceedings 9th Text Retrieval Conference (TREC-9)*, pp. 400–410. Gaithersburg, MD.

https://doi.org/10.1017/S1351324903003176 Published online by Cambridge University Press