Probabilistic Question Answering on the Web

Dragomir Radev
University of Michigan, Ann Arbor, MI 48109. E-mail: radev@umich.edu

Weiguo Fan
Virginia Polytechnic Institute and State University, Blacksburg, VA 24061. E-mail: wfan@vt.edu

Hong Qi, Harris Wu, and Amardeep Grewal
University of Michigan, Ann Arbor, MI 48109. E-mail: {hqi, harriswu, asgrewal}@umich.edu

Web-based search engines such as Google and NorthernLight return documents that are relevant to a user query, not answers to user questions. We have developed an architecture that augments existing search engines so that they support natural language question answering. The process entails five steps: query modulation, document retrieval, passage extraction, phrase extraction, and answer ranking. In this article, we describe some probabilistic approaches to the last three of these stages. We show how our techniques apply to a number of existing search engines, and we also present results contrasting three different methods for question answering. Our algorithm, probabilistic phrase reranking (PPR), uses proximity and question type features and achieves a total reciprocal document rank of .20 on the TREC8 corpus. Our techniques have been implemented as a Web-accessible system, called NSIR.

Introduction to Web-Based Q&A

Given the amount of information that is available on the Web, it is not surprising that it is an ideal source of answers to a large variety of questions. The problem is that existing front ends to the Web such as NorthernLight, Google, and AlltheWeb's FAST search engine are designed to retrieve *documents* that are relevant to a user *query*, posed in an idiosyncratic language and not to a *question* formulated in a *human language* such as English.

We claim that it is much more natural for a user to type a question such as, Who wrote King Lear? or What is the largest city in Northern Afghanistan?, rather than queries such as (wrote OR written OR author) AND (King Lear). We also believe that when a user is looking for an answer, rather than a document, then the search engine should return an actual answer, possibly in the context of the document where it occurs. The process of retrieving answers from questions is known as Natural Language Question Answering (NLQA).

In this article, we will introduce a method for Web-based question answering called Probabilistic Phrase Reranking (PPR). It is fully implemented at the University of Michigan as a Web-accessible system, called NSIR (pronounced answer). The existing NLQA systems that are closest to NSIR are Ask Jeeves,[1] Mulder[2] (Kwok, Etzioni, & Weld, 2001), and Ionaut[3] (Abney, Collins, & Singhal, 2000). Ask Jeeves accepts questions but doesn't actually return answers but rather engages in a menu-driven dialogue with the user. Mulder is no longer available on the Web for a comparison. Ionaut is fast and interactive, but it is based on a local cache of files and doesn't provide access to the full Web.

We should note that all search engines allow a user to enter a natural language question instead of a query. The search engines then remove certain frequent stop words such as is or where and treat the rest of the question as a query. Thus search engines behave as if they can handle the first stage of NLQA (input in the form of a question). However, they still provide documents rather than answers as their output results. For example, when we asked the question What is the largest city in Northern Afghanistan? in the Google[4] search engine, we got the following results back (the full list has been truncated for reasons of space).

Received September 3, 2003; revised March 2, 2004; accepted March 2, 2004

© 2005 Wiley Periodicals, Inc. • Published online 10 February 2005 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/asi.20146

[1] http://www.ask.com
[2] http://mulder.cx
[3] http://www.ionaut.com:8400/
[4] http://www.google.com

Yahoo! Full Coverage - Afghanistan
. . . within three miles of the airport at Mazar-e-Sharif, the **largest city** in **northern Afghanistan**,
held since 1998 by the Taliban. There was no immediate comment . . .
uk.fc.yahoo.com/photos/a/afghanistan.html
- 13k - Cached - Similar pages

washingtonpost.com: World
. . . died in Kano, **northern** Nigeria's **largest city**,
during two days of anti-American riots led by Muslims protesting the US-led bombing of **Afghanistan**,
according to . . .
www.washingtonpost.com/wp-dyn/print/world/-
Similar pages

The result consists of short summaries of all relevant documents plus pointers to the documents themselves. It is clear that a document returned by the search engine as relevant to the input question is likely to contain the answer. If the very first returned document doesn't contain the answer, it is still possible for another top ranked document to contain it. The problem becomes then how to identify the correct answer within the top *n* relevant documents returned by a search engine.

A similar output, produced by AlltheWeb is shown in Figure 1.

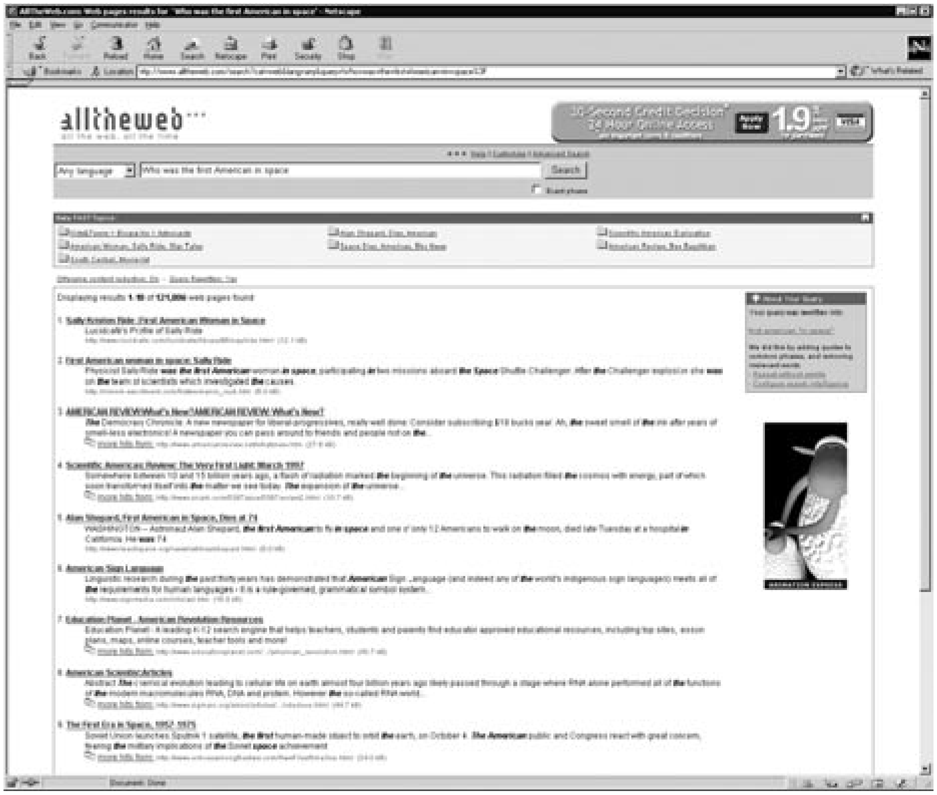
After reviewing relevant previous work, this article will describe our new method for Web-based NLQA. The PPR technique (Probabilistic Phrase Reranking) relies on an existing search engine to return documents that are likely to contain the answer to a user question. PPR goes through several stages until it extracts and ranks the most likely answers to the question. These stages are query modulation, document retrieval, passage (or sentence) retrieval, phrase (answer) extraction, and answer ranking. These are described in more detail in the following sections.

All experiments described in this article were performed by the authors using AlltheWeb, Google, and Northern Light in the period September through November 2001.

Related Work

START (Katz, 1997) is one of the first Web-based question answering (QA) systems. However, it focuses only on questions about geography and the MIT InfoLab. A precompiled knowledge base is used to answer questions. Another earlier system, MURAX (Kupiec, 2001), uses an encyclopedia as a knowledge base to answer trivia questions. Given a question, MURAX uses a shallow parser to extract potential answers from sections of the encyclopedia based on the phrasal relationships between words in the question.

A large number of QA systems have emerged recently. Primarily, they follow two directions: one direction is to use the TREC Q&A (Voorhees & Tice, 2000) data as the test corpus and develop their own search engines and answer extraction techniques on top of the corpus; the other direction is to use the World Wide Web as the potential answer source and use generic search engines, such as Google, to retrieve


AlltheWeb.com: Web pages results for 'Who was the first American in space' - Netscape

Any language | Who was the first American in space | Search

Displaying results 1-10 of 121,006 web pages found

1 Sally Kristen Ride: First American Woman in Space
Lucidcafé's Profile of Sally Ride

2 First American woman in space: Sally Ride
Physicist Sally Ride was the first American woman in space, participating in two missions aboard the Space Shuttle Challenger. After the Challenger explosion she was on the team of scientists which investigated the causes.

3 AMERICAN REVIEW:What's New?AMERICAN REVIEW: What's New?
The Democracy Chronicle. A new newspaper for liberal-progressives, really well done. Consider subscribing $10 bucks year. Ah, the sweet smell of the ink after years of smell-less electronics! A newspaper you can pass around to friends and people not on the...
more hits from

4 Scientific American Review: The Very First Light: March 1997
Somewhere between 10 and 15 billion years ago, a flash of radiation marked the beginning of the universe. This radiation filled the cosmos with energy, part of which soon transformed itself into the matter we see today. The expansion of the universe...
more hits from

5 Alan Shepard, First American in Space, Dies at 74
WASHINGTON -- Astronaut Alan Shepard, the first American to fly in space and one of only 12 Americans to walk on the moon, died late Tuesday at a hospital in California. He was 74.

6 American Sign Language
Linguistic research during the past thirty years has demonstrated that American Sign Language (and indeed any of the world's indigenous sign languages) meets all of the requirements for human languages - it is a rule-governed, grammatical symbol system...

7 Education Planet - American Revolution Resources
Education Planet - A leading K-12 search engine that helps teachers, students and parents find educator approved educational resources, including top sites, lesson plans, maps, online courses, teacher tools and more!
more hits from

8 American Scientific Articles
Abstract The chemical evolution leading to cellular life on earth almost four billion years ago likely passed through a stage where RNA alone performed all of the functions of the modern macromolecules RNA, DNA and protein. However the so-called RNA world...
more hits from

9 The First Era in Space, 1957-1975
Soviet Union launches Sputnik 1 satellite, the first human-made object to orbit the earth, on October 4. The American public and Congress react with great concern, fearing the military implications of the Soviet space achievement.
more hits from

Document: Done

FIG. 1. Sample output from AlltheWeb.

information related to the question and do further post-processing to extract the answers for the questions. We will review some recent work in this section.

Related Work From TREC

The TREC question answering evaluation (Voorhees & Tice, 2000) is the motivating force behind a recent surge in question answering research. Systems participating in TREC have to identify short passages from a 2-GB text corpus that contain answers to a list of factual questions. The IBM team (Prager, Radev, Brown, & Coden, 1999; Radev, Libner, & Fan, 2000) introduced the technique of predictive annotation, a methodology for indexing texts for fact-seeking question answering. The idea of this approach is that texts in documents are annotated with labels anticipating their being targets of certain kinds of questions. Given a question, their system retrieves a set of passages that may contain the answers. The potential answers are then extracted from these passages and ranked using a linear ranking function with various heuristics, which is learned by logistic regression.

Hovy, Gerber, Hermjakob, Junk, and Lin (2000) developed a Q&A system called Webclopedia based on information retrieval (IR) and natural language processing (NLP) techniques. A given question is first parsed to create a query to retrieve the top ranked documents. These top ranked documents are then split into segments and further ranked. Potential answers are then extracted and sorted according to a ranking function involving the match with the question type and patterns. These question patterns are manually constructed. The classification of each question and potential answer to these patterns is done using rules learned from a machine-learning based grammar parser.

Abney et al. (2000) described a method based on named entity identification techniques. For each question, a set of relevant passages that mostly contain the answers is first identified. A candidate set of entities are extracted from these retrieved passages. Both the question and these extracted entities are classified into a predefined set of categories. Only those entities that match the category required by the question are retained and ranked again using the frequency and other position related information.

Clarke, Cormack, Kisman, and Lynam (2000) also applied the passage retrieval techniques for initial preprocessing. The passage ranking algorithm utilizes semantic match information between the query type and term, the IDF-like (inverse document frequency, a measure of the spread of a word or phrase among documents in a corpus) term, weighting information of each term, and also the coverage of these query related terms in the passage itself. A statistical context free grammar parser based on WordNet is used to determine the question category. Those top ranked passages are then scanned for patterns matching the answer category, and the potential answers are extracted and ranked using various quality heuristics.

A different technique, named boosting, which integrates different forms of syntactic, semantic, and pragmatic knowledge, is presented in Harabagiu et al. (2000). For example, question reformulation is used to construct a query that contains more semantic information based on WordNet, and named entity recognition techniques are employed to ensure high quality passage retrieval. Potential answers are extracted from the semantically rich passages that match the question type. These candidate answers are further justified by using abductive reasoning, and only those that pass the test are retrieved. The system, named Falcon, scored very high in recent TREC Q&A evaluation contests (Voorhees & Tice, 2000; Voorhees, 2001).

TextRoller, the best Q&A system in recent TREC10 contest (Soubbotin, 2002), used a relatively unique approach called pattern-based approach for answer identification. In this approach, NLP techniques are no longer used. Instead, various patterns for different types of questions are defined and used in pattern matching from passages and answer selection and ranking of those potential answers.

Related Work on the Web

The TREC conference offers an exciting environment for competitive research on question answering. However, the questions that can be answered from the fixed text corpus as in TREC are limited. Various efforts are now under way that try to port existing Q&A techniques to a much larger context—the World Wide Web. Our earlier study (Radev et al., 2001a) already shows that the current WWW search engines, especially those with a very large index like Google, offer a very promising source for question answering.

Agichtein, Lawrence, and Gravano (2001) presented a technique on how to learn search engine specific query transformations for question answering. A similar transformation technique also appeared in Glover et al. (2001). The idea is that the current query interfaces of most generic search engines, such as Google, do not provide enough capability for direct question answering in a natural language mode. By transforming the initial natural questions into a certain format that includes more domain specific information, one can dramatically improve the chances of finding good answers at the top of the search hit lists. A set of transformation rules are learned from a training corpus and applied to the questions at search time, and the experiments have shown promising results. Their work, however, is focused on improving the chances of getting high quality documents from a search engine. It does not provide any mechanism to identify the true answers from the search results. An even more general framework on query transformation is discussed in Radev et al. (2001b). A total of 15 query modulation operators are identified and trained using the EM algorithm over a wide range of training questions. The best operators tailored to different questions are then identified and applied for later online question answering using the Web search engines.

A relatively complete, general-purpose, Web-based Q&A system is discussed recently in Kwok et al. (2001). Techniques similar to those used in the TREC conferences are applied to the web as a whole. For example, the user question is processed by a parser to learn its syntactic structure. In order to help extract the answers and make the classification task easier, the questions are classified into three categories: nominal, numerical and temporal. Similar to the techniques used in Agichtein et al. (2001) and Radev et al. (2000), various query modulation techniques, such as query expansion, noun phrase formation, and transformation are applied to the initial questions to get high quality results for later answer extraction. Their answer extraction module utilizes both IDF information and word distance to extract answers. These extracted answers are then assigned scores based on their contexts in documents and then further clustered into groups. The member with the highest score in each group is selected as a representative for that group and is presented to the user as a potential answer.

Our system is different from Kwok et al. (2001) in that we do not use a deep natural language parser. As the authors admit on the Mulder Web page, such approaches are very slow and are not usable on the Web until faster parsers and/or hardware become available. In fact, the Mulder system runs on 100 workstations in parallel (O. Etzioni & D. S. Weld, personal communication, February 27, 2002) to make it scalable. Instead, we replace these time-consuming parts with approaches based on rule-based classifiers and probabilistic phrase reranking and still achieve reasonable performance on a single Sun workstation. As will be shown later, these new additions offer a much more scalable approach for the Web context than previous work.

A Probabilistic Approach to Phrase Reranking

The General Architecture of Our System

Our general approach includes the following stages as shown in Figure 2:

- Query modulation—the process of converting a question to an appropriate query. This is an optional stage. We have already described an approach for query modulation (Radev et al., 2001b). In this article, we will present an approach that doesn't need query modulation to answer natural language questions.
- Question type recognition—in this stage questions are subdivided depending on the type of answer that expected: for example, person, place, organization, distance, and so on.
- Document retrieval—the process of returning documents that are likely to contain, among other things, the answer(s) to the input question.
- Passage (or sentence) retrieval—the process of identifying in the retrieved documents the textual units that contain the answers.
- Answer extraction—that is where the relevant sentences or passages are split up into constituent phrases, each of which is a potential answer candidate.
- Phrase (answer) ranking—these phrases extracted in the previous stage are ranked with the goal of getting the right answer near the top of the phrase-level hit list.

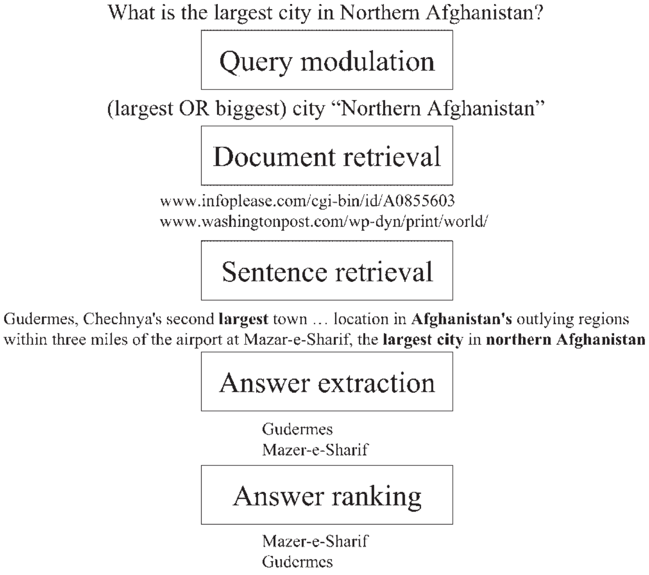

What is the largest city in Northern Afghanistan?
Query modulation
(largest OR biggest) city "Northern Afghanistan"
Document retrieval
www.infoplease.com/cgi-bin/id/A0855603
www.washingtonpost.com/wp-dyn/print/world/
Sentence retrieval
Gudermes, Chechnya's second **largest** town … location in **Afghanistan's** outlying regions
within three miles of the airport at Mazar-e-Sharif, the **largest city** in **northern Afghanistan**
Answer extraction
Gudermes
Mazer-e-Sharif
Answer ranking
Mazer-e-Sharif
Gudermes

FIG. 2. The architecture of the NSIR system.

Evaluation Metric

As an evaluation metric, we use total reciprocal document rank (TRDR). The value of TRDR is the sum of the reciprocal values of the rank of all correct documents among the top *n* (arbitrary) extracted by the system:

$$\mathrm{TRDR} = \sum_{i}^{N} \frac{1}{rank_i} \qquad (1)$$

where *N* is the number of returned documents, passages, or phrases. The main reason of using the proposed TRDR as the main performance measure is that there may be multiple correct answers per question or the same correct answer may appear in multiple documents of different ranks. For example, if the system has retrieved 10 documents, of which three, the second, eighth, and tenth, contain the correct answer, TRDR for that given paraphrase is $\frac{1}{2} + \frac{1}{8} + \frac{1}{10} = .725$. TRDR can be defined similarly for the sentence level or the phrase level. In these cases, it is based on the rank of the correct sentences (respectively, phrases). The TRDR metric is similar to the maximal marginal relevance (MRR) metric used in the TREC evaluation. The difference between the two is that in TREC, only the top ranked answer counts whereas, in our case, we want to be able to distinguish between two runs, one of which only gets a correct answer in second place and another, which gets correct answers not only in second, but also in eighth and tenth places. Using TRDR rather than the metric employed in TREC, we are able to make finer distinctions in performance. Another departure from the TREC scheme is our inability to perform a manual evaluation due to the large number of judgments that are needed to evaluate the different methods described in this article. We need to rely instead on an automatic evaluation scheme in which a document (resp., sentence or phrase) is checked by a program for the presence (even unjustified) of the expected answer. This sort of evaluation is similar to the lenient evaluation described in Voorhees and Tice (2000). In the next subsection, we will show that it is a reasonable compromise to use an automatic evaluation mechanism to measure performance in a large system. We should note here that for all experiments described in this article, we have used the TREC8, TREC9, and TREC10 corpora from National Institute for Standards and Technology (NIST) (Voorhees & Tice, 2000). They contain a total of more than 1,000 question/answer pairs. Unless otherwise indicated, we have tuned our system based on the 200 questions from TREC8 since this is period when most of our training experiments were conducted. The Mulder system uses a different metric for evaluation, namely *user effort*, which measures the total number of words shown on a page before the first correct answer is identified. We chose to use a metric closer in spirit to the established metrics in TREC.

Manual Versus Automatic Evaluation Consistency

There are some cases where the document does match the answer patterns but does not actually support the answer. For example, 1987 is one of the answers for the question: *When was London's Docklands Light Railway constructed;* automatic evaluation cannot tell if a document containing 1987 is really about when the railway was constructed or about other events that happened in 1987, a distinction which a human can judge correctly. The question that arises is to what extent the automatic evaluation is consistent with human judges. We experimented to examine the consistency between the results of automatic and manual evaluation.

We ran the first 100 TREC8 questions on the AlltheWeb search engine and evaluated the first 5 hits automatically and manually. Document-level TRDR performance scores are computed for each question and for both methods. We used the formula to get the Pearson correlation between the two data sets,

$$r = \frac{n\sum xy - \sum x \sum y}{\sqrt{(n\sum x^2 - (\sum x)^2)(n\sum y^2 - (\sum y)^2)}} \tag{2}$$

where x and y are the 100 reciprocal performance scores of manual evaluation and automatic evaluation, respectively. The Pearson correlation score derived from this formula is .538, which shows reasonable correlation between the manual and automatic performance scores and, as a result, justifies the use of automatic evaluation when manual evaluation is too expensive (e.g., on tens of thousands of question-document pairs). Note that this automatic method for evaluation contrasts with the small-scale manual evaluation described in Agichtein et al. (2001).

Question Type Identification

To identify the semantic type of the question is an important step before extracting the actual answer. For example, a question like *Who was the tallest U.S. president?* expects a person as the answer. Currently, we have 17 question types, listed in Table 1. Two methods have been implemented to categorize questions: decision rule induction using Ripper (Cohen, 1996) and a heuristic rule-based algorithm. We used 1200 questions from TREC8, TREC9 and TREC10 in our experiment. The automatically identified question types (qtype) are then compared against manually annotated question types.

TABLE 1. List of question types (qtypes).

PERSON	PLACE	DATE
NUMBER	DEFINITION	ORGANIZATION
DESCRIPTION	ABBREVIATION	KNOWNFOR
RATE	LENGTH	MONEY
REASON	DURATION	PURPOSE
NOMINAL	OTHER	

Description is used for questions seeking a description of some person, such as *Who was Whitcomb Judson?*; while questions like *Who invented the paper clip?* should be labeled as a Person. Nominal describes questions which have nominal phrases as answers but cannot be assigned to other specific categories such as person or organization. Questions not belonging to any of the above types fall in Other.

Machine Learning Approach

Ripper is the machine learning tool used for question categorization. In our experiment, each question is represented by 13 features, 9 of which are semantic features based on WordNet. For instance, one of the semantic features is ifNounIsMoney, which checks if the hypernyms of the first noun in the sentence contains money related words such as monetary, economic, liability, and so on. The questions have also been processed by Language Technologies Chunks (LTCHUNK) (Mikheev, 2000), which yields the NumberOfNounPhrases features. All the features are listed in Table 2.

Several experiments have been done using Ripper for question type identification. Questions from TREC9, TREC8, and TREC10 were incrementally added to the training data set. In addition, we manually added 37 data points to the training data set, which helps produce more robust rules. For example, , how, many, n, , y, , , , , , , , LENGTH:4. means that questions which have wh-word how many followed by a Length noun should be categorized as a Length type. These manual data points are very helpful when Ripper uses a training data set to produce the hypotheses. For the example data point, Ripper generates a hypothesis which can be represented as LENGTH 5 0 IF (ifNounIsLength LIKE y) AND (wordbeside LIKE many). So when Ripper predicts question types on the test data set,

TABLE 2. Features for question representation.

QuestionWords	Whword
WordBesideWhwords	ifWordBesideIsWealthy
ifNounIsMoney	ifNounIsLength
ifNounIsDuration	ifNounIsPerson
ifNounIsLocation	ifNounIsGroup
ifNounIsRate	ifNounIsNumber
NumberOfNounPhrases	

TABLE 3. Results (in error rate) of using Ripper to identify question types.

Train	Test	Train error	Test error
TREC9	TREC8	22.4%	24%
TREC8,9	TREC10	17.03%	30%
TREC8,9,10	—	20.69%	—

it will know that the questions such as *How many miles is it from London, England to Plymouth, England* are expecting a length as the answer. The Results of using Ripper to identify question types are listed in Table 3.

Heuristic Algorithm

The second method for type identification that we use to categorize questions is heuristic in nature. Question categorization seems trivial since the question word is often a good indication of the semantic type of the answer. However, this is not true even for the who questions. We examined 484 TREC9 questions containing wh-words with respect to the mapping between wh-words and question types.

As it can be seen from Table 4, only a small number of these wh-words can determine the question types, such as when and why. What questions can cover almost all the different types. For the exceptions of who and where questions, we use the simple rules like those listed in Table 5.

But for what/which questions, syntactic and semantic analysis is needed. In our system, Brill's transformation-based POS tagger is used to tag questions (Brill, 1995). What/which can either be tagged as a WDT (determiner) as in *What state in the United States covers the largest area?*, or as a WP (wh-phrase) as in *What is the population of Japan?* The base noun phrase right after a WDT what can often be used to determine the question type. We select the last noun of this noun phrase as the informative noun which will be further used to determine the question type with semantic analysis. For example, in the question What/WDT card/ NN company/NN sells/VBZ Christmas/NNP ornaments/ NNS?/., the first base noun phrase is card company, so company becomes the informative noun for this question; we then categorize this question as ORGANIZATION. Compared to WDT what, questions with a WP what are more complicated for this task. First, the verbs in questions are used to categorize questions like *What caused the Lynmouth floods?* in which the verbs indicate the types; then questions are typed as DEFINITION if the number of question words is one or two excluding wh-words or any determiners; for the remaining questions, the system needs to extract the informative noun. Our general heuristic for finding the informative noun is to locate the last noun of the first base noun phrase. For example, in the question *What's the average salary of a professional baseball player?*, we get two base noun phrases, which are the average salary and a professional baseball player, and we then use salary as the informative noun for this question. Different heuristics are used for questions like *What was the name of the first Russian astronaut to do a spacewalk?* or *What person's head is on a dime?* These heuristics have also been applied to the questions containing no wh-word, like *Name a film in which Jude Law acted.*

The decision of some questions' types are left to the informative nouns. This is implemented through a lexicon built manually that maps nouns to their corresponding categories, such as length | circumference | diameter | diam | radius→ LENGTH. So if the informative noun is diameter, then the question will be categorized as a LENGTH type. WordNet has been used to build the mapping lexicon. Table 6 shows the result of using our heuristics to determine the question types. The first line is the results of the heuristics generated by only looking at TREC9 questions; the second includes both TREC8 and TREC9 questions; the last line shows the results of the heuristics after being modified by all three TREC question sets. It is somewhat disappointing to see that the addition of TREC8 questions in training does not help the testing performance on TREC10 questions.

The accuracy has been greatly improved by using heuristics. When using Ripper, the training error rate is around 20%, and the test error rate is even higher, going to 30%

TABLE 4. Analysis of Wh-words and their corresponding types.

Wh-word	Types
who(102)	PERSON(77) DESCRIPTION(19) ORG(6)
where(60)	PLACE(54) NOMINAL(4) ORG(2)
when(40)	DATE(40)
why(1)	REASON(1)
what/which(233)	NOMINAL(78) PLACE(27) DEFINITION(26) PERSON(18) ORG(16) NUMBER(14) ABBREVIATION(13) DATE(11) RATE(4) KNOWNFOR(8) MONEY(3) PURPOSE(2) REASON(1) TRANSL(1) LENGTH(1) DESCOTHER(10)
how(48)	NUMBER(33) LENGTH(6) RATE(2) MONEY(2) DURATION(3) REASON(1) DESCOTHER(1)

TABLE 5. Sample rules.

Template	Types
who is <Person Name>	Description
who (manufacture | produce | grow | provide) . . .	Organization

TABLE 6. Results (in error rate) of using heuristics to identify question types.

	Test		
Train	TREC9	TREC8	TREC10
TREC9	7.8%	15%	18%
TREC8,9	7.4%	6%	18.2%
TREC8,9,10	4.6%	5.5%	7.6%

when trained on TREC8,9 and tested on TREC10. As it can be seen from Table 6, training error rate never goes above 8%, and the testing error is around 18%. It should be noted that the training error rate exists because some questions are really hard to categorize without any additional information. For example, questions like Who won . . . could expect a person as its answer such as *Who won the Nobel Peace Prize in 1991*, but it also could expect an organization such as *Who won the Superbowl in 1982*.

Document Retrieval

We use the offline interfaces to three of the major search engines, AlltheWeb, Northern Light, and Google. The input to these interfaces is a query (or a question in our case). The output is a list of the URLs of the top matching documents. We use the Perl LWP::Download module to retrieve the actual documents before we split them up into sentences and/or phrases. We retrieve the top 40 documents from the search engine. In the evaluation section, we will show our method's performance on the document level as well as the passage and phrase levels. Note: all retrievals used in this article were performed in late October and early November of 2001.

Sentence Ranking

The purpose of sentence ranking is to reduce the computational complexity of the later phrase ranking stage. Producing phrases directly from the downloaded documents requires substantial computation.

To perform sentence ranking, two models were used in our system. One is based on an N-gram model. The other is based on the traditional Vector Space model.

For the N-gram model, the question submitted by a user is parsed to generate unigrams, bigrams, and trigrams. Various lexical statistics about these N-grams are used for sentence ranking. The formula for scoring sentences (passages) by proximity to the words of the query using the N-gram model is as follows:

$$Score = \frac{w_1 \sum_{i=1}^{N_1} tf_i * idf_i + w_2 \sum_{j=1}^{N_2} tf_j + w_3 \sum_{k=1}^{N_3} tf_k}{Normalized\_Factor} \quad (3)$$

where $N_i (i = 1, 2, 3)$ is the total number of occurrences of unigram, bigram, and trigram in a sentence. $w_i (i = 1, 2, 3)$ is the linear combination weight. We set the weights as 1, 1.5, and 4, respectively in our experiments. tf_i is the term frequency of the *i*-gram. *idf* is the inverse document frequency, which measures the rarity of a unigram. *Normalized_Factor* is defined as follows:

$$\begin{cases} 1 & \text{if } Sentence\_Length < 40 \\ Sentence\_Length/40 & \text{if } Sentence\_Length > 40 \end{cases}$$

Another sentence ranking function is designed based on modification of the Okapi ranking function used for document ranking (Robertson, Walker, Jones, Hancock-Beaulieu, & Gatford, 1996). It is defined as follows:

$$Score(S) = \sum_{T \in Q} \frac{3 \times tf \times idf}{0.5 + 1.5 \times \frac{Sentence\_Length}{Sentence\_Length_{avg}} + tf} \quad (4)$$

where *Sentence_Length* is the length of a sentence in words and $Sentence\_Length_{avg}$ is the average sentence length in the top 20 documents returned from a search. *tf* and *idf* have a similar meaning as in the linear combination formula, except that they are calculated based on single terms (unigrams) only. In the evaluation section, we show the performance of NSIR on the sentence level.

Phrase Extraction and Ranking

Potential Answer Identification

In our study, we convert all retrieved documents from the Web into chunks using an off-the-shelf chunker (Mikheev, 2000). For a typical question, after downloading the top 40 hits from a given search engine, the chunker produces several tens of thousands of phrasal chunks. Here is an example. Given the question (from TREC8) *Who is the author of the book, "The Iron Lady: A Biography of Margaret Thatcher"?*, 10,360 phrases are returned. Of these 10,360 phrases, 10 contain the correct answer Hugo Young. However, these 10 phrases are scattered among the 10,360 phrases and need to be identified automatically. To address this issue, our current algorithm utilizes the following feature: proximity between the text and the question—that is, a phrase that contains most query terms gets a high score, however a phrase that is *near* a phrase that contains most query terms will get a slightly lower score. Figure 3 shows the effect of proximity on the score of a phrase. Phrases that are next to a large number of query words get higher scores than phrases that are further away.

Phrase Ranking

Answer Extraction and Ranking is the module that looks at additional sources of information to improve the performance of the previous module. We have been experimenting

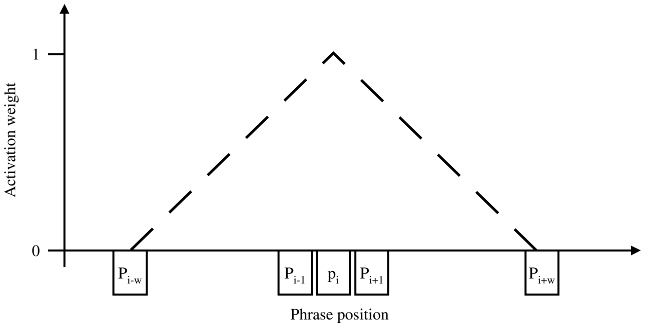


FIG. 3. Proximity feature.

with the so-called part-of-speech phrase signatures which identify with a particular question or answer type. For example, the expected answer type for the question about Margaret Thatcher's biography is a person. We can compute the probabilities $P(\text{PHRASETYPE}|\text{SIG})$ for all possible signatures and phrase types. For example, the phrase signature for the phrase Hugo Young is determined by the chunker as NNP NNP. The probability $P(\text{PERSON}|\text{NNP NNP})$ is .458. We will call this probability the *signature score*. For a given phrase, we can then multiply two numbers, the proximity score from the previous subsection and the signature score from the current subsection. Figure 4 shows some of the most common part of speech (POS) signatures and the phrase types that correspond to each of them. The part of speech symbols are generated by the text chunker (Mikheev, 2000). For instance, VBD indicates the past participle form of a verb, JJ is an adjective, DT is a determiner (e.g., the), and NNP is a proper noun. Other Q&A systems use commercial grade named entity taggers. Our approach could also benefit from such taggers although phrase signatures seem to work quite well.

Signature	Phrase Types
VBD	NO (100%)
DT NN	NO (86.7%) PERSON (3.8%) NUMBER (3.8%) ORG (2.5%)
NNP	PERSON (37.4%) PLACE (29.6%) DATE (21.7%) NO (7.6%)
DT JJ NN	NO (75.6%) NUMBER (11.1%) PLACE (4.4%) ORG (4.4%)
NNP NNP	PLACE (37.3%) PERSON (35.6%) NO (16.9%) ORG (10.2%)
DT NNP	ORG (55.6%) NO (33.3%) PLACE (5.6%) DATE (5.6%)

FIG. 4. Common part-of-speech signatures and their corresponding phrase types. A NO means that this particular signature cannot be the answer to any type of question. The frequencies are derived from a manually annotated corpus of text unrelated to the TREC corpus.

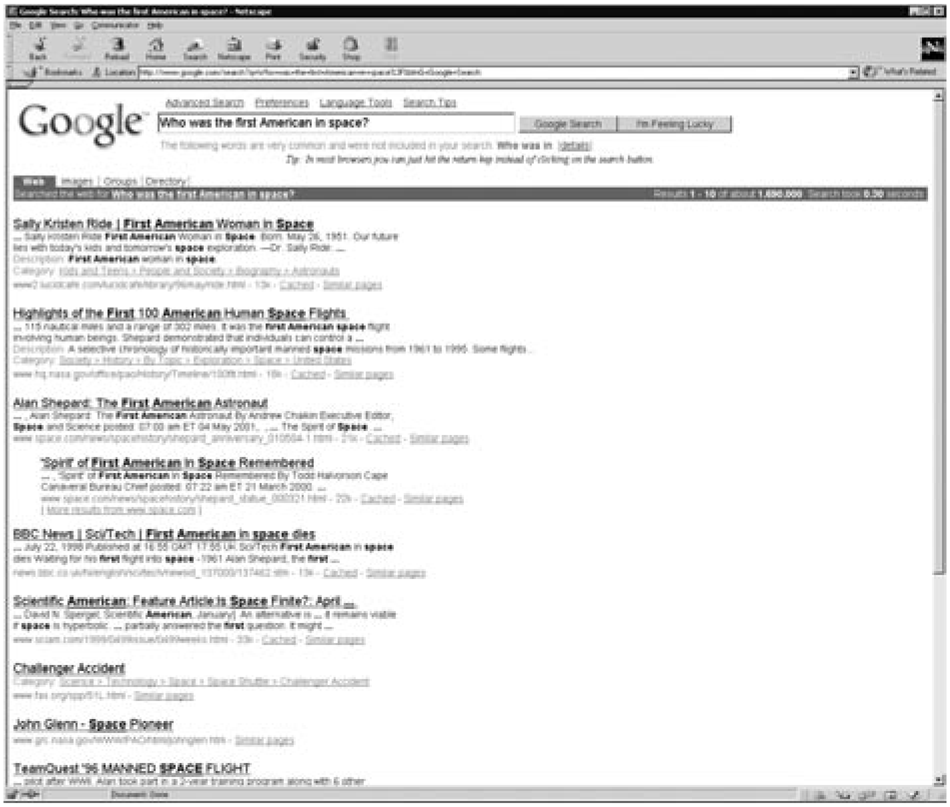
Note that the step of answer identification and ranking can be done based on sentences instead of the original downloaded documents. We will compare the performances of these approaches in the experiments section.

Example

Let's consider question 21 from TREC8: *Who was the first American in space?*[5] The expected correct answer is Alan Shepard.

We submit this question to the Google search engine and get the following results back (see Figure 5). Note that a number of person names appear in the hit list. These include several occurrences of the correct answer as well as many other names. The top 40 documents returned by Google for this question contain a total of 14,717 phrases, of which approximately 2% are names of people for a total of 300 names. Among these names, our system needs to pick out the instances of Alan Shepard.

[5] This question was used as a running example in Kwok et al. (2001) as well.



Who was the first American in space?

The following words are very common and were not included in your search: Who was in. [details]

Tip: In most browsers you can just hit the return key instead of clicking on the search button.

Searched the web for Who was the first American in space? Results 1 - 10 of about 1,690,000. Search took 0.30 seconds.

Sally Kristen Ride | First American Woman in Space
... Sally Kristen Ride First American Woman in Space. Born: May 26, 1951. Our future lies with today's kids and tomorrow's space exploration. —Dr. Sally Ride: ...
Description: First American woman in space.
Category: Kids and Teens > People and Society > Biography > Astronauts
www2.lucidcafe.com/lucidcafe/library/96may/ride.html - 13k - Cached - Similar pages

Highlights of the First 100 American Human Space Flights
... 115 nautical miles and a range of 302 miles. It was the first American space flight involving human beings. Shepard demonstrated that individuals can control a ...
Description: A selective chronology of historically important manned space missions from 1961 to 1995. Some flights...
Category: Society > History > By Topic > Exploration > Space > United States
www.hq.nasa.gov/office/pao/History/Timeline/100fl.html - 18k - Cached - Similar pages

Alan Shepard: The First American Astronaut
... , Alan Shepard: The First American Astronaut By Andrew Chaikin Executive Editor, Space and Science posted: 07:00 am ET 04 May 2001, ... The Spirit of Space ...
www.space.com/news/spacehistory/shepard_anniversary_010504-1.html - 21k - Cached - Similar pages

'Spirit' of First American in Space Remembered
... 'Spirit' of First American in Space Remembered By Todd Halvorson Cape Canaveral Bureau Chief posted: 07:22 am ET 21 March 2000 ...
www.space.com/news/spacehistory/shepard_statue_000321.html - 22k - Cached - Similar pages
[More results from www.space.com]

BBC News | Sci/Tech | First American in space dies
... July 22, 1998 Published at 16:55 GMT 17:55 UK Sci/Tech First American in space dies Waiting for his first flight into space - 1961 Alan Shepard, the first ...
news.bbc.co.uk/hi/english/sci/tech/newsid_137000/137462.stm - 13k - Cached - Similar pages

Scientific American: Feature Article:Is Space Finite?: April ...
... David N. Spergel, Scientific American, January). An alternative is ... it remains viable if space is hyperbolic. ... partially answered the first question. It might ...
www.sciam.com/1999/0499issue/0499weeks.html - 33k - Cached - Similar pages

Challenger Accident
Category: Science > Technology > Space > Space Shuttle > Challenger Accident
www.fas.org/spp/51L.html - Similar pages

John Glenn - Space Pioneer
www.grc.nasa.gov/WWW/PAO/html/johnglen.htm - Similar pages

TeamQuest '96 MANNED SPACE FLIGHT
... pilot after WWII. Alan took part in a 2-year training program along with 6 other

FIG. 5. Google result given TREC8 Question 21.

American in space **Alan Shepard** Becomes the First American in Space 1961 On May 5 1961

Alan Shepard Jr. was launched into space.

Keep in mind that **Alan Shepard** was the first American in space not the first man.

First American woman in space: **Sally Ride** First American woman in space: **Sally Ride** **Sally Ride** was the first American woman to travel into space.

Submitted by Anonymous Contributor Title: First American woman in space: **Sally Ride** Description: Physicist **Sally Ride** was the first American woman in space participating in two missions aboard the Space Shuttle Challenger.

It was the first American space flight involving human beings.

Privacy Copyright Disclaimer First American in Space May 5 In 1961 I lived in a small town in Florida called Neptune Beach.

As a Mercury astronaut he was chosen to be the first American in space.

He was the first American to dare the unknown and enter space.

FIG. 6. The top ranked sentences from our system (names of PERSONs are shown in bold face, and the correct answer is boxed).

It is not diffcult to see from Figure 5 that articles in the top two of the hit list are about the first American woman, not the first American in general. The correct answer appears in the third hit.

The output of our system at the sentence level is shown in Figure 6.

Figure 7 shows the top phrases retrieved by our system directly from the documents and using the proximity feature only. Figure 8 reflects the use of both features (proximity and qtype) but without taking into account the top ranked sentences. Figure 9 shows how these phrases are re-ranked after taking into account the top sentences from the returned documents and also the qtype feature. As we will show in the following sections, such combination of features improves overall system performance by a factor of at least 2.

Let’s consider an example. The top ranked phrase in Figure 7 is *the Space Flight Operations contractor.* Its signature is DT NNP NNP NNP NN. That signature is not associated with the expected question types PERSON or PLACE. In other words,

Rank	Probability and phrase
1	0.600 the_DT Space_NNP Flight_NNP Operations_NNP contractor_NN._.
2	0.599 International_NNP Space_NNP Station_NNP Alpha_NNP
3	0.598 International_NNP Space_NNP Station_NNP
4	0.598 to_TO become_VB
5	0.595 a_DT joint_JJ venture_NN United_NNP Space_NNP Alliance_NNP
6	0.594 NASA_NNP Johnson_NNP Space_NNP Center_NNP
7	0.587 will_MD form_VB
8	0.585 The_DT purpose_NN
9	0.577 prime_JJ contracts_NNS
10	0.568 First_NNP American_NNP
11	0.567 this_DT bulletin_NN board_NN
12	0.566 Space_NNP:_:
13	0.563 ’Spirit_NN ’_” of_IN
14	0.562 space_NN
15	0.561 February_NPN
...	
41	**0.516 Alan_NNP Shepard_NNP**

FIG. 7. Document + Phrase, proximity only, Question 21 from TREC8. The highest-ranking correct answer is in position 41 out of 14,717 phrases when phrases are ranked solely based on proximity.

Rank	Probability and phrase
1	0.465 Space_NNP Administration_NNP ._.
2	0.446 SPACE_NNP CALENDAR_NNP _.
3	0.414 First_NNP American_NNP
4	0.399 International_NNP Space_NNP Station_NNP Alpha_NNP
5	0.396 her_PRP$ third_JJ space_NN mission_NN
6	0.396 NASA_NNP Johnson_NNP Space_NNP Center_NNP
7	0.394 the_DT American_NNP Commercial_NNP Launch_NNP Industry_NNP
8	0.390 the_DT Red_NNP Planet_NNP ._.
9	0.380 First_NNP American_NNP
10	**0.376336 Alan_NNP Shepard_NNP**
11	0.376 February_NNP
12	0.375 Space_NNP
13	0.374 International_NNP Space_NNP Station_NNP
14	0.372 Als_NNPS
15	0.371 The_NNP Spirit_NNP

FIG. 8. Document + Phrase, proximity + qtype, Question 21. The highest-ranking correct answer moves up to 10th place based on proximity and question type.

$$\begin{aligned} &P(\text{PERSON}|\text{DT NNP NNP NNP NN}) \\ &\quad = P(\text{PLACE}|\text{DT NNP NNP NNP NN}) = 0 \end{aligned}$$

As a result, the combined probability is 0. On the other hand, the highest-ranking Alan Shepard is in forty-first place out of 14,717 phrases with a proximity score of .516.

When taking into consideration the qtype feature in addition to proximity, that phrase moves up to tenth place with a score of .376. That score is the product of the proximity score .516 and the qtype score .729. How was the qtype score computed? It is equal to

$$P(\text{PLACE}|\text{NNP NNP}) + P(\text{PERSON}|\text{NNP NNP}) \quad (5)$$

or in terms of actual numbers, .373 + .356. We should note here that we have modified the output of Ripper to specify the top two candidate categories for each question—that is why we are getting both PERSON and PLACE as candidate question types for this question.

Finally, the phrase Alan Shepard moves even higher in the list (to sixth place) when the list of phrases is limited to the 1,935 phrases in the highest ranking 50 sentences returned by our sentence ranking component. Overall, our TRDR for this question is .18 (.14 + .03 + .01).

Experimental Comparison

We performed several experiments at different levels of granularity to study how we can effectively improve our chances of finding good answers from the search results. As we mentioned earlier, we have several different ways of pruning the search space to get the answers at different levels: document, sentence, and phrase.

Rank	Probability and phrase
1	0.479 Neptune_NNP Beach_NNP ._.
2	0.449 February_NNP
3	0.447 Go_NNP
4	0.438 Space_NNP
5	0.432 Go_NNP
6	**0.425 Alan_NNP Shepard_NNP**
7	0.424 First_NNP American_NNP
8	0.421 Space_NNP May_NNP
9	0.411 First_NNP American_NNP woman_NN
10	0.402 Life_NNP Sciences_NNP
11	0.386 Space_NNP Shuttle_NNP Discovery_NNP STS-60_NN
12	0.382 the_DT Moon_NNP International_NNP Space_NNP Station_NNP
13	0.370 Space_NNP Research_NNP A_NNP Session_NNP
14	0.367 First_NNP American_NNP
15	0.359 Sally_NNP Ride_NNP Sally_NNP Ride_NNP

FIG. 9. Document + Sentence + Phrase, proximity + qtype, Question 21. The correct answer moves to 6th place when a sentence filter is used before extracting answer phrases.

We will next compare each of these ranking schemes from searches of all three search engines.

Document Level Performance

The performance at the document level from all three engines is summarized in Table 7, where Average is the average of all total reciprocal scores. $\# > 0$ is the number of times that correct answers are identified in the top 40 documents. The numbers in the parenthesis are the total number of questions for evaluation in a track. Using the results from AlltheWeb as the baseline, we can see from the TREC8 results that both NorthernLight and Google did better overall than AlltheWeb in getting good answers, with Google completely dominant in this category. Using Google improves the chance of finding correct answers by almost 60%. Moreover, Google returns documents containing correct answers for 164 questions out of 200, a 10% improvement over AlltheWeb. Google is the best engine as the source for potential answers in Web-based question answering. We list in Table 7 the document level performance results of Google for TREC9 and TREC10 questions.

TABLE 7. The performance comparison among search engines at the document level. A score of 1 or above means that either the top ranked document is the single correct answer or that the correct hits have a sum of reciprocal ranks equal to 1 or above (e.g., 2, 4, 5, 20).

Question source	Search engine	Average	Compared to AlltheWeb	# > 0	Compared to AlltheWeb
TREC8	AlltheWeb	0.836	—	149(200)	—
	NorthernLight	1.050	25.61%	163(200)	9.40%
	Google	1.336	59.92%	164(200)	10.07%
TREC9	Google	1.553	—	603(693)	—
TREC10	Google	1.339	—	410(500)	—

Sentence Level Performance

Table 8 summarizes the performance results for sentence ranking using results from various search engines. The sentence ranking results using the linear combination formula is listed in the row under Linear, and similar results using the modified Okapi formula are listed under Okapi.

In each cell of the row for Average, the three values reported are: total reciprocal value for the top 50 sentences of each question, improvement over the upper-bound for a particular engine, and the improvement using one engine over the baseline engine—AlltheWeb. The upper-bound performance is obtained by calculating the total reciprocal score for all those sentences containing the answer with the assumption that these sentences are ranked at the top of the list. For example, if there are a total of five sentences containing the answer, then the upper-bound for the sentence ranking is $(1 + 1/2 + 1/3 + 1/4 + 1/5)$. An ideal phrase ranking algorithm would achieve the upper bound. Giving upper bounds at each stage (document, passage, phrase) allows for the performance at each stage to be measured separately. A similar upper-bound definition is used for phrase ranking evaluation discussed in the next subsection.

As can be seen from the TREC8 results (shown in Table 8) that Linear combination sentence ranking, using sentences extracted from the top 20 search results from Google, gives the best results. Similarly, the column for $\# > 0$ reports the number of questions where the correct answers have been found in the sentences produced by the sentence ranking formula, along with the comparison with the upper-bound and that of the baseline engine of AlltheWeb. Again using the linear combination formula with the Google engine gives the best result: 137 out of 200. Note that we lose some of the correct answers for 22 (159 – 137) questions. This is due to the fact that some of the sentences that contain the answers are not ranked at the top according

TABLE 8. Sentence ranking comparison for TREC questions.

Question source	Search engine	Measure	Average	Percentage of the upper bound	Compared to AlltheWeb	# > 0	Percentage of the upper bound	Compared to AlltheWeb
TREC8	AlltheWeb	Upper	2.131	—	—	148/200	—	—
		Linear	0.313	14.68%	—	99/200	66.89%	—
		Okapi	0.262	13.60%	—	99/200	66.89%	—
	Northern-Light	Upper	2.526	—	—	159/200	—	—
		Linear	0.477	18.87%	52.33%	121/200	76.10%	22.22%
		Okapi	0.440	17.41%	51.73%	119/200	74.84%	20.20%
	Google	Upper	2.553	—	—	159/200	—	—
		Linear	0.540	21.15%	72.63%	137/200	86.16%	38.38%
		Okapi	0.493	19.32%	70.19%	135/200	84.91%	36.36%
TREC9	Google	Upper	2.769	—	—	566/693	—	—
		Linear	0.685	24.74%	—	495/693	87.46%	—
		Okapi	0.611	22.07%	—	496/693	87.63%	—
TREC10	Google	Upper	2.356	—	—	406/500	—	—
		Linear	0.642	27.25%	—	320/500	78.82%	—
		Okapi	0.610	25.89%	—	317/500	78.08%	—

to the sentence ranking formula. Our later manual evaluation of these results shows that some of these problems are caused by our simple sentence segmenter while others simply reflect the existence of spurious answers. To further test the sentence level performance on more questions, we apply the sentences ranking formulas to TREC9 and TREC10 questions as well using Google as the backend. The results, summarized in Table 8, show that linear sentence ranking is still better than the Okapi sentence ranking. Overall, the results for TREC9 and TREC10 are relatively better than those for TREC8.

Phrase-Level Performance

The phrase level performance is shown in Table 9. D + P means phrases are generated and ranked from the original downloaded documents. D + S + P means phrases are created and ranked from the top 50 sentences after sentence ranking. Four different types of performance are reported. The upper-bound definition is similar to the one used in sentence ranking. Appearance Order means that the phrases are ordered based on the appearance position of these phrases in the order of their original ranked documents. Proximity utilizes only the proximity information, i.e., the overlap between a phrase and a user query, to rank phrases. Proximity and qtype uses both the proximity and the signature information to do the phrase ranking, which is essentially our proposed PPR (probabilistic phrase ranking) approach.

As can be seen in Table 9, using our PPR approach (the last column) produced much better results than using "Appearance Order" and "Proximity" alone. This indicates the advantage of our PPR approach over simple heuristics. For TREC 8 questions, Google again is the best in the D + P (document + phrase) performance category among the three search engines examined. A further test using D + S + P (document + sentence + phrase) on Google shows that the performance is dramatically improved over the D + P approach using our PPR approach (the last column), which demonstrates that sentence extraction and ranking from original retrieved documents can indeed improve the Q&A performance than using original documents directly. The final TRDR performance of NSIR is 0.199. As a comparison, the value of MRDR (mean reciprocal document rank) would be 0.151 which is significantly lower than those in TREC (the top performance at TREC8 was around .40). There are multiple reasons for this discrepancy. First, TREC8 questions were pulled directly from the 2-GB TREC corpus by rephrasing existing sentences. Second, all questions were guaranteed to have answers in the corpus. Third, the TREC corpus consists only of clean news articles while the Web contains significantly more heterogeneous texts.

We also test the Google (D + S + P) on TREC9 and TREC10 questions and find the results are comparable to what we get for TREC8 questions. These results are summarized in Table 9 as well.

Conclusions and Future Work

We presented a probabilistic method for Web-based Natural Language Question Answering. It has been implemented in a robust system and has been tested on a realistic corpus of questions.

One thing we didn't address in this article is the scalability issue. Even though the current system performs relatively faster than other Web-based question answering systems, the current system's performance for real-time question answering remains to be improved. One thing that deserves further attention is that, after extensive testing, we found that many preprocessing steps such as page downloading, sentence segmentation, part of speech tagging, and so on, take most of the response time. Even though parallel processing can be used to speed up the downloading phase, the dependence of existing Web search engines as answer sources is really the bottleneck of our system. We expect to improve performance significantly by using a prebuilt snapshot of a search engine's content.

NSIR currently takes between 5 and 30 seconds per question depending on the (user-specified) number of documents to be downloaded from the Web and on the (again user-specified) number of phrases to extract. The current version of NSIR doesn't include query modulation (Radev et al., 2001b; the process of converting a question to the best query for a given search engine).

In our future work, we plan to add query modulation to the system, fine tune various phrases of the system, and experiment with additional heuristics in answer selection. We also plan to combine the PPR approach with other efficient heuristics to further improve the final performance of our system.

TABLE 9. Phrase ranking performance.

Question source	Search engine	Upper bound	Appearance order	Proximity	Proximity and qtype
TREC8	AlltheWeb D+P	2.176	0.026	0.038	0.105
	NorthernLight D+P	2.652	0.048	0.054	0.117
	Google D+P	2.698	0.068	0.058	0.157
	Google D+S+P	1.941	0.065	0.065	0.199
TREC9	Google D+S+P	2.017	0.068	0.071	0.162
TREC10	Google D+S+P	1.873	0.051	0.059	0.158

Acknowledgments

We would like to thank Bojan Peovski and Michael Gimbel for help with data annotation, and Timothy Allison and Airong Luo for proofreading.

References

Abney, S., Collins, M., & Singhal, A. (2000). Answer extraction. In The Proceedings of ANLP 2000 (pp. 296–301). Morgan Kaufman.

Agichtein, E., Lawrence, S., & Gravano, L. (2001). Learning search engine specific query transformations for question answering. In The Proceedings of the 10th World Wide Web Conference (WWW 2001). New York: ACM.

Brill, E. (1995). Transformation-based error-driven learning and natural language processing: A case study in part of speech tagging. Computational Linguistics, 21(4), 543–566.

Clarke, C.L.A., Cormack, G.V., Kisman, D.I.E., & Lynam, T.R. (2000). Question answering by passage selection (multitext experiments for TREC-9). In NIST Special Publication 500-249: The Ninth Text REtrieval Conference (TREC 9) (pp. 673–683). Maryland: NIST.

Cohen, W.W. (1996). Learning trees and rules with set-valued features. In Proceedings of the Thirteenth National Conference on Artificial Intelligence and the Eighth Innovative Applications of Artificial Intelligence Conference (pp. 709–716). Menlo Park: AAAI Press/MIT Press.

Glover, E.J., Flake, G.W., Lawrence, S., Birmingham, W.P., Kruger, A., Giles, C.L., & Pennock, D.M. (2001). Improving category specific web search by learning query modifications. In The Proceedings of the Symposium on Applications and the Internet, SAINT 2001 (pp. 23–31). IEEE Computer Society.

Harabagiu, S., Moldovan, D., Pasca, M., Mihalcea, R., Surdeanu, M., Bunescu, R., Gîrju, R., Rus, V., & Morarescu, P. (2000). Falcon: Boosting knowledge for answer engines. In NIST Special Publication 500-249: The Ninth Text REtrieval Conference (TREC9) (pp. 479–488). Maryland: NIST.

Hovy, E., Gerber, L., Hermjakob, U., Junk, M., & Lin, C.-Y. (2000). Question answering in webclopedia. In NIST Special Publication 500-249: The Ninth Text REtrieval Conference (TREC9) (pp. 655–664). Maryland: NIST

Katz, B. (1997). From sentence processing to information access on the World Wide Web. In Natural Language Processing for the World Wide Web: Papers from the 1997 AAAI Spring Symposium (pp. 77–94). California: AAAI.

Kupiec, J. (2001). Murax: A robust linguistic approach for question answering using an on-line encyclopedia. In Proceedings of the 16th SIGIR Conference (pp. 150–161). New York: ACM.

Kwok, C., Etzioni, O., & Weld, D.S. (2001). Scaling question answering to the web. In Proceedings of the 10th World Wide Web Conference (WWW 2001).

Mikheev, A. (2000). Document centered approach to text normalization. In Proceedings of SIGIR 2000 (pp. 136–143). New York: ACM.

Prager, J., Radev, D., Brown, E., & Coden, A. (1999). The use of predictive annotation for question answering in TREC8. In NIST Special Publication 500-246: The Eighth Text REtrieval Conference (TREC8) (pp. 399–411). Maryland: NIST.

Radev, D.R., Libner, K., & Fan, W. (2001a). Getting answers to natural language queries on the web. Journal of the American Society for Information Science and Technology, 53(5), 359–364.

Radev, D.R., Prager, J., & Samn, V. (2000). Ranking suspected answers to natural language questions using predictive annotation. In Proceedings of the Sixth Conference on Applied Natural Language Processing (ANLP) (pp. 150–157), Morgan Kaufman.

Radev, D.R., Qi, H., Zheng, Z., Blair-Goldensohn, S., Zhang, Z., Fan, W., & Prager, J. (2001b). Mining the web for answers to natural language questions. In Proceedings of ACM CIKM 2001: Tenth International Conference on Information and Knowledge Management (pp. 143–150). California: ACM.

Robertson, S.E., Walker, S., Jones, S., Hancock-Beaulieu, M.M., and Gatford, M. (1996). Okapi at TREC-4. In D.K. Harman (Ed.), Proceedings of the Fourth Text REtrieval Conference (pp. 73–97). NIST Special Publication 500-236.

Soubbotin, M.M. (2002). Patterns of potential answer expressions as clues to the right answers. In NIST Special Publication 500-250: The Tenth Text REtrieval Conference TREC-10 (pp. 293–302). Gaithersburg, MD: NIST

Voorhees, E. (2001). The TREC-9 question answering track evaluation. In NIST Special Publication 500-249: The Ninth Text REtrieval Conference TREC-9. Gaithersburg, MD: NIST.

Voorhees, E., & Tice, D. (2000). The TREC-8 question answering track evaluation. In Text REtrieval Conference TREC-8. Gaithersburg, MD: NIST.