MURAX: A Robust Linguistic Approach For Question Answering Using An On-Line Encyclopedia

Julian Kupiec

Xerox Palo Alto Research Center
3333 Coyote Hill Road, Palo Alto, CA 94304

Abstract

Robust linguistic methods are applied to the task of answering closed-class questions using a corpus of natural language. The methods are illustrated in a broad domain: answering general-knowledge questions using an on-line encyclopedia.

A closed-class question is a question stated in natural language, which assumes some definite answer typified by a noun phrase rather than a procedural answer. The methods hypothesize noun phrases that are likely to be the answer, and present the user with relevant text in which they are marked, focussing the user's attention appropriately. Furthermore, the sentences of matching text that are shown to the user are selected to confirm phrase relations implied by the question, rather than being selected solely on the basis of word frequency.

The corpus is accessed via an information retrieval (IR) system that supports boolean search with proximity constraints. Queries are automatically constructed from the phrasal content of the question, and passed to the IR system to find relevant text. Then the relevant text is itself analyzed; noun phrase hypotheses are extracted and new queries are independently made to confirm phrase relations for the various hypotheses.

The methods are currently being implemented in a system called MURAX and although this process is not complete, it is sufficiently advanced for an interim evaluation to be presented.

ACM-SIGIR'93-6/93/Pittsburgh, PA, USA

1 Introduction

The paper is organized as follows. First the motivation for the question-answering task is given and a description of the kind of questions that are its concern, and their characteristics. A description of the system components is given in Section 3. These include the encyclopedia and the IR system for accessing it. Shallow linguistic analysis is done using a part-of-speech tagger and finite-state recognizers for matching lexico-syntactic patterns.

Section 4 describes the analysis of a question by considering an example, and the system output is illustrated. Analysis proceeds in two stages. The first, primary query construction, finds articles that are relevant to the question. The second stage (called answer extraction) analyzes these articles to find noun phrases (called answer hypotheses) that are likely to be the answer.

Both stages require searching the encyclopedia. Queries made during the first stage are called primary queries, and only involve phrases from the question. The second stage creates secondary queries which are generated by MURAX to verify specific phrase relations. Secondary queries involve both answer hypotheses and phrases from the question.

Primary query construction is explained in Section 5, followed by a complete description of answer extraction in Section 6. An informal evaluation and discussion are then presented.

2 Task Selection

The task is concerned with answering general-knowledge questions using Grolier's on-line encyclopedia. The task is motivated by several criteria and and goals. Robust analysis is needed because the encyclopedia is composed of a significant quantity of unrestricted text. General-knowledge is a broad domain, which means that it is impractical to manually provide detailed lexical or semantic information for the words of the vocabulary (the

encyclopedia contains over 100,000 word stems). The methods demonstrate that shallow syntactic analysis can be used to practical advantage in broad domains, where the types of relations and objects involved are not known in advance, and may differ for each new question. The analysis must capitalize on the information available in a question, and profit from treating the encyclopedia as a lexical resource.

The methods also demonstrate that natural language analysis can add to the quality of the retrieval process, providing text to the user which confirms phrase relations and not just word matches. The task also serves as a practical focus for the development of linguistic tools for content analysis and reveals what kind of grammar development should be done to improve performance.

The use of closed-class questions means that performance can be evaluated in a straightforward way by using a set of questions and correct answers. Given a correct noun phrase answer, it is generally easy to judge whether a noun phrase hypothesized by the system is correct or not. Thus relevance judgements are simplified, and if one correct hypothesis is considered as good as any other, recall measurements are not required and performance can be considered simply as the percentage of correctly hypothesized answers.

1. What U.S. city is at the junction of the Allegheny and Monongahela rivers?
2. Who wrote "Across the River and into the Trees"?
3. Who married actress Nancy Davis?
4. What 's the capital of the Netherlands?
5. Who was the last of the Apache warrior chiefs?
6. What chief justice headed the commission that declared: "Lee Harvey Oswald ...acted alone."?
7. What famed falls are split in two by Goat Island?
8. What is November's birthstone?
9. Who 's won the most Oscars for costume design?
10. What is the state flower of Alaska?

Figure 1: Example Questions

2.1 Question Characteristics

A closed-class question is a direct question whose answer is assumed to lie in a set of objects and is expressible as a noun phrase. Such questions are exemplified in Figure 1. These questions appear in the general-knowledge "Trivial Pursuit"[1] game and typify the form of question that is the concern of the task. They have the virtue of being created independently of the retrieval task (i.e. are unbiassed) and have a consistent and simple stylized form; yet they are flexible in their expressive power.

Who/Whose:	*Person*		
What/Which:	*Thing,*	*Person,*	*Location*
Where:	*Location*		
When:	*Time*		
How Many:	*Number*		

Table 1: Question Words and Expectations

The interrogative words that introduce a question are an important source of information. They indicate particular expectations about the answer and some of these are illustrated in Table 1. Notable omissions are the words *why* and *how*, expecting a procedural answer rather than a noun phrase [2] (e.g. "How do you make a loaf of bread?").

These expectations can be used to filter various answer hypotheses. The answers to questions beginning with the word "who" are likely to be people's names. This fact can be used to advantage because various heuristics can be applied to verify whether a noun phrase is a person's name.

A question introduced by "what" may or may not refer to a person; however, other characteristics can be exploited. Consider the following sentence fragments, where *NP* symbolizes a noun phrase: "What is the *NP*..." and "What *NP*...". The noun phrase at the start of such questions is called the question's *type phrase* and it indicates what type of thing the answer is. The encyclopedia can be searched to try to find evidence that an answer hypothesis is an instance of the type phrase (details are in Section 6.1.1). The verbs in a question are also a useful source of information as they express a relation that exists between the answer and other phrases in the question.

The answer hypotheses for "Where ..." questions are likely to be locations, which often appear with locative prepositions or as arguments to verbs of motion. Questions of the form "When ..." often expect answer hypotheses that are dates or times and the expectation of questions beginning "How many ..." are numeric expressions.

Closed-class questions are also addressed by a system [Wendlandt and Driscoll, 1991] for accessing public in-

[1] Copyright Horn Abbot Ltd., Trivial Pursuit is a Registered Trademark of Horn Abbot Ltd.

[2] Questions requiring procedural answers are not considered unimportant, but of more concern after initial goals have been attained.

formation documents at NASA Kennedy Space Center (e.g. "What are the dimensions of the cargo area in the shuttle?"). In the system, conventional word-based similarity measures are augmented with terms for thematic roles, obtained from a manually constructed lexicon.

3 Components

An on-line version of Grolier's Academic American Encyclopedia [Grolier, 1990] was chosen as the corpus for the task. It contains approximately 27,000 articles, which are accessed via the Text Database (TDB) [Cutting *et al.*, 1991], which is a flexible platform for the development of retrieval system prototypes and is structured so that additional functional components (e.g. search strategies and text taggers [Cutting *et al.*, 1992]) can be easily integrated.

The components responsible for linguistic analysis are a part-of-speech tagger and a lexico-syntactic pattern matcher. The tagger is based on a hidden Markov model (HMM). HMM's are probabilistic and their parameters can be estimated by training on a sample of ordinary untagged text. Once trained, the Viterbi algorithm is used for tagging. To assess performance, an HMM tagger [Kupiec, 1992b] was trained on the untagged words of half of the Brown corpus [Francis and Kučera, 1982] and then tested against the manually assigned tags of the other half. This gave an overall error rate of 4% (corresponding to an error rate of 11.2% on words that can assume more than one part-of-speech category). The percentage of tagger errors that affect correct recognition of noun phrases is much lower than 4%. The tagger uses both suffix information and local context to predict the categories of words for which it has no lexicon entries.

The HMM used for tagging the encyclopedia text was also trained using the encyclopedia. A benefit of such training is that the tagger can adapt to certain characteristics of the domain. An observation in this regard was made with the word "I". The text of the encyclopedia is written in an impersonal style and the word is most often used in phrases like "King George I" and "World War I". The tagger trained on encyclopedia text assigned "I" appropriately (as a proper noun) whereas the tagger trained on the Brown corpus (a mixture of different kinds of text) assigned such instances as a pronoun.

Given a sentence of text, the tagger produces a sequence of pairs of words with associated part-of-speech categories. These enable phrase recognition to be done. Phrases are specified by regular expressions in the finite-state calculus [Hopcroft and Ullman, 1979]. Noun phrases are identified solely by part-of-speech categories, but more generally categories and words are used to define lexico-syntactic patterns against which text is matched. This kind of pattern matching has also been exploited by others (e.g. [Jacobs *et al.*, 1991, Hearst, 1992]).

Initially, only simple noun phrases are identified because they are recognized with the greatest reliability. Analysis involving prepositional phrases or other coordination is applied subsequently as part of more detailed matching procedures. Word-initial capitalization was found to be useful for splitting a noun phrase appropriately, thus "New York City borough" is split into "New York City" and "borough". Such splitting improves the efficiency of boolean query construction (enabling direct phrase matches, rather than requiring several words to be successively dropped from the phrase).

3.1 Title Phrases

A multi-word phrase that is the title of a film, book, play, etc., is usefully treated as a single unit. Furthermore, it may not be a simple noun phrase (e.g. Play Misty for Me). Such phrases are readily identified when marked typographically by enclosing quotes or italics. However, title phrases may be marked only by word-initial capitalized letters; furthermore, some words (such as short function words) may not be capitalized. Thus, the correct extent of the phrase may be ambiguous and alternative possibilities must be accommodated. The most likely alternative is chosen after phrase matching has been done and the alternatives compared, based on the matches and frequency of the alternative interpretations.

4 Operational Overview

This section presents an informal description of the operation of the system, by tracing the analysis steps for an example question, shown in Figure 2.

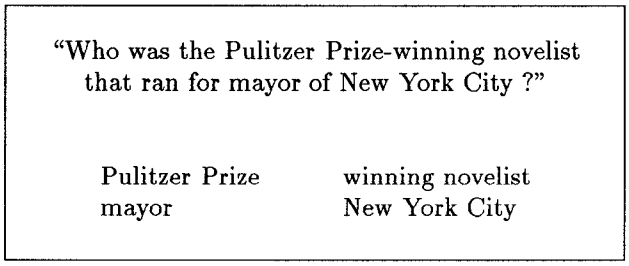

Figure 2: Example Question and Component NP's

4.1 Primary Document Matches

Simple noun phrases and main verbs are first extracted from the question, as illustrated in the figure. These question phrases are used in a query construction/refinement procedure that forms boolean queries

with associated proximity contraints (Section 5). The queries are used to search the encyclopedia to find a list of relevant articles from which primary document matches are made. These are sentences containing one or more of the question phrases.

Primary document matches are heuristically scored according to the degree and number of matches with the question phrases. Matching head words in a noun phrase receive double the score of other matching words in a phrase. Words with matching stems but incompatible part-of-speech categories are given minimal scores. Primary document matches are then ranked according to their scores.

4.2 Extracting Answers

It is assumed that primary document matches contain answer hypotheses, so answer extraction begins by finding all simple noun phrases contained in them. Each noun phrase is an answer hypothesis distinguished by its components words, and the article and sentence in which it occurs. Answer hypotheses are themselves scored on a per-article basis according to the sum of the scores of primary document matches in which they occur. The purpose of this is to minimize the probability of overlooking the correct answer hypothesis if a subsequent non-exhaustive search is performed using the hypotheses.

For each answer hypothesis the system tries to verify phrase relations implied by the question. For the question in Figure 2, we note that the answer is likely to be a person (indicated by "who"). The type phrase indicates the answer is preferably a "Pulitzer Prize winning novelist", or at least a "novelist" as indicated by the head noun of the type phrase. The relative pronoun indicates that the answer also "ran for mayor of New York City". Phrase matching procedures (detailed in Section 6) perform the verification using the answer hypotheses and the primary document matches, but the verification is not limited to primary document matches.

It can happen that a pertinent phrase relation is not present in the primary document matches although it can be confirmed elsewhere in the encyclopedia. This is because too few words are involved in the relation in comparison to other phrase matches, so the appropriate sentence does not rank high enough to be in the selected primary document matches. It is also possible that the appropriate information is not expressed in any primary document match and depends only on the answer hypothesis. This is the case with one heuristic that the system uses to try and verify that a noun phrase represents a person's name. The heuristic involves looking for an article that has the noun phrase in its title; thus if the article does not share any phrases with the question, it would not be part of any primary document match.

Secondary queries are used as an alternative means to confirm phrase relations. A secondary query may consist of solely an answer hypothesis (as for the heuristic just mentioned) or it may also include other question phrases such as the question's type phrase. To find out whether an answer hypothesis is a "novelist", the two phrases are included in a query and a search yields a list of relevant articles. Sentences which contain co-occurrences are called secondary document matches. The system analyzes secondary document matches to see if answer hypotheses can be validated as instances of the type phrase via lexico-syntactic patterns.

The best matching phrase
for this question is: **Mailer, Norman**

The following documents were most relevant:

Document Title: **Mailer, Norman**
Relevant Text:

- "The Armies of the Night (1968), a personal narrative of the 1967 peace march on the Pentagon, won **Mailer** the **Pulitzer Prize** and the National Book Award."
- "In 1969 **Mailer** ran unsuccessfully as an independent candidate for **mayor** of **New York City**."

Document Title: novel
Relevant Text:

- "Among contemporary American **novelists**, Saul Bellow, John Dos Passos, John Hawkes, Joseph Heller, **Norman Mailer**, Bernard Malamud, Thomas Pynchon, and J. D. Salinger have reached wide audiences."

Next best: Edith Wharton, William Faulkner

Figure 3: Example Output

4.3 System Output

For the given question the system produces the output shown in Figure 3. The presentation is different from extant IR systems. Answer hypotheses are shown to the user to focus his attention on likely answers and how they relate to other phrases in the question. The text presented is not necessarily from documents that have high similarity scores, but those which confirm phrase relations that lend evidence for an answer. This behaviour is readily understood by users, even though they have not been involved in the tedious intermediate work done by the system.

In Figure 3, the first two sentences are from primary document matches. The last sentence confirming Norman Mailer as a novelist is a secondary document match. It was confirmed by a lexico-syntactic pattern which identifies the answer hypothesis as being in a list-inclusion relationship with the type phrase.

We next consider this approach in contrast to a common alternative, vector-space search. Vector-space search using full-length documents is not as well suited to the task. For the example question, a search was done using a typical similarity measure and the bag of content words of the question. The most relevant document (about Norman Mailer) was ranked 37th. Somewhat better results could be expected if sentence or paragraph level matching was done (cf. [Salton and Buckley, 1991]). However the resulting text matches do not have the benefit of being correlated in terms of a particular answer and they muddle information for different answer hypotheses.

5 Primary Query Construction

This section describes how phrases from a question are translated into boolean queries with proximity constraints. These are passed to an IR system which searches the encyclopedia and returns a list of matching documents (or *hits*). The following functionality is assumed of the IR system:

1. The boolean AND of terms, denoted here as: $[term_1, term_2, ...term_n]$
2. Proximity of a strict sequence of terms, separated by up to p other terms denoted here as: $\{p\ term_1, term_2, ...term_n\}$
3. Proximity of an unordered list of terms, separated by up to p other terms denoted here as: $(p\ term_1, term_2, ...term_n)$

The overall process is again illustrated via an example question:

"Who shot President Lincoln ?"

The question is first tagged and the noun phrases and main verbs are found. In the above case the only noun phrase is *President Lincoln* and the main verb is *shot*. Boolean terms are next constructed from the phrases. At the outset a strict ordering is imposed on the component words of phrases. For the preceding question, the first query is:

{0 president lincoln}

The IR system is given this boolean query and searches for documents that match. Depending on the number of hits, new boolean queries may be generated with the purpose of:

1. Refining the ranking of the documents.
2. Reducing the number of hits (Narrowing).
3. Increasing the number of hits (Broadening).

Iterative broadening and narrowing has been investigated for the situation where phrase structure is not considered [Salton *et al.*, 1983].

5.1 Narrowing

Items (1) and (2) above are performed by using title phrases (Section 3.1) rather than the noun phrases, or by adding extra query terms such as the main verbs and performing a new search in the encyclopedia. Including the main verb in the example gives:

[{0 president lincoln} shot]

Narrowing is done to try to reduce the number of hits. It also involves reducing the co-occurrence scope of terms in the query and constrains phrases to be closer together (and thus indirectly there is a higher probability of them being in some syntactic relation with each other). A sequence of queries with increasingly smaller scope are made, until there are fewer hits than some predetermined threshold. A narrowed version for the previous example is shown below:

(10 {0 president lincoln} shot)

5.2 Broadening

Broadening is done to try and increase the number of hits for a boolean query. It is achieved in three ways:

1. Increasing the co-occurrence scope of words within phrases, while jointly dropping the requirement for strict ordering of the words. E.g. (5 president lincoln) would match the phrase "President Abraham Lincoln". A sequence of queries with increasingly larger scope are made until some threshold on either the proximity or resulting number of hits is reached.
2. Dropping one or more whole phrases from the boolean query. Query terms, each corresponding to a phrase, are dropped to get more hits. It is efficient to drop them in an order that corresponds to decreasing number of overall occurrences in the encyclopedia.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.