Exact Phrases in Information Retrieval for Question Answering

Svetlana Stoyanchev, and Young Chol Song, and William Lahti
Department of Computer Science
Stony Brook University
Stony Brook, NY 11794-4400
svetastenchikova, nskystars, william.lahti @gmail.com

Abstract

Question answering (QA) is the task of finding a concise answer to a natural language question. The first stage of QA involves information retrieval. Therefore, performance of an information retrieval subsystem serves as an upper bound for the performance of a QA system. In this work we use phrases automatically identified from questions as exact match constituents to search queries. Our results show an improvement over baseline on several document and sentence retrieval measures on the WEB dataset. We get a 20% relative improvement in MRR for sentence extraction on the WEB dataset when using automatically generated phrases and a further 9.5% relative improvement when using manually annotated phrases. Surprisingly, a separate experiment on the indexed AQUAINT dataset showed no effect on IR performance of using exact phrases.

1 Introduction

Question answering can be viewed as a sophisticated information retrieval (IR) task where a system automatically generates a search query from a natural language question and finds a concise answer from a set of documents. In the open-domain factoid question answering task systems answer general questions like *Who is the creator of The Daily Show?*, or *When was Mozart born?*. A variety of approaches to question answering have been investigated in TREC competitions in the last decade from (Vorhees and Harman, 1999) to (Dang et al., 2006). Most existing question answering systems add question analysis, sentence retrieval and answer extraction components to an IR system.

Since information retrieval is the first stage of question answering, its performance is an upper bound on the overall question answering system's performance. IR performance depends on the quality of document indexing and query construction. Question answering systems create a search query automatically from a user's question, through various levels of sophistication. The simplest way of creating a query is to treat the words in the question as the terms in the query. Some question answering systems (Srihari and Li, 1999) apply linguistic processing to the question, identifying named entities and other query-relevant phrases. Others (Hovy et al., 2001b) use ontologies to expand query terms with synonyms and hypernyms.

IR system recall is very important for question answering. If no correct answers are present in a document, no further processing will be able to find an answer. IR system precision and ranking of candidate passages can also affect question answering performance. If a sentence without a correct answer is ranked highly, answer extraction may extract incorrect answers from these erroneous candidates. Collins-Thompson *et al.* (2004) show that there is a consistent relationship between the quality of document retrieval and the overall performance of question answering systems.

In this work we evaluate the use of *exact phrases* from a question in document and passage retrieval. First, we analyze how different parts of a question contribute to the performance of the sentence extraction stage of question answering. We ana-

Coling 2008: Proceedings of the 2nd workshop on Information Retrieval for Question Answering (IR4QA), pages 9–16
Manchester, UK. August 2008

lyze the match between linguistic constituents of different types in questions and sentences containing candidate answers. For this analysis, we use a set of questions and answers from the TREC 2006 competition as a *gold standard*.

Second, we evaluate the performance of document retrieval in our *StoQA* question answering system. We compare the performance of document retrieval from the Web and from an indexed collection of documents using different methods of query construction, and identify the optimal algorithm for query construction in our system as well as its limitations.

Third, we evaluate passage extraction from a set of documents. We analyze how the specificity of a query affects sentence extraction.

The rest of the paper is organized as follows: In Section 2, we summarize recent approaches to question answering. In Section 3, we describe the dataset used in this experiment. In Section 5, we describe our method and data analysis. In Section 4, we outline the architecture of our question answering system. In Section 6, we describe our experiments and present our results. We summarize in Section 7.

2 Related Work

Information retrieval (IR) for question answering consists of 2 steps: document retrieval and passage retrieval.

Approaches to passage retrieval include simple word overlap (Light et al., 2001), density-based passage retrieval (Clarke et al., 2000), retrieval based on the inverse document frequency (IDF) of matched and mismatched words (Ittycheriah et al., 2001), cosine similarity between a question and a passage (Llopis and Vicedo, 2001), passage/sentence ranking by weighting different features (Lee and others, 2001), stemming and morphological query expansion (2004), and voting between different retrieval methods (Tellex et al., 2003). As in previous approaches, we use words and phrases from a question for passage extraction and experiment with using exactly matched phrases in addition to words. Similarly to Lee (2001), we assign weights to sentences in retrieved documents according to the number of matched constituents.

Systems vary in the size of retrieved passages. Some systems identify multi-sentence and variable size passages (Ittycheriah et al., 2001; Clarke et al., 2000). An optimal passage size may depend on the method of answer extraction. We use single sentence extraction because our system's semantic role labeling-based answer extraction functions on individual sentences.

White and Sutcliffe (2004) performed a manual analysis of questions and answers for 50 of the TREC questions. The authors computed frequency of terms matching exactly, with morphological, or semantic variation between a question and a answer passage. In this work we perform a similar analysis automatically. We compare frequencies of phrases and words matching between a question and candidate sentences.

Query expansion has been investigated in systems described in (Hovy et al., 2001a; Harabagiu et al., 2006). They use WordNet (Miller, 1995) for query expansion, and incorporate semantic roles in the answer extraction process. In this experiment we do not expand query terms.

Corpus pre-processing and encoding information useful for retrieval was shown to improve document retrieval (Katz and Lin, 2003; Harabagiu et al., 2006; Chu-Carroll et al., 2006). In our approach we evaluate linguistic question processing technique which does not require corpus preprocessing.

Statistical machine translation model is used for information retrieval by (Murdock and Croft, 2005). The model estimates probability of a question given an answer and is trained on <question, candidate sentence> pairs. It capturing synonymy and grammar transformations using a statistical model.

3 Data

In this work we evaluate our question answering system on two datasets: the AQUAINT corpus, a 3 gigabyte collection of news documents used in the TREC 2006 competition; and the Web.

We use questions from TREC, a yearly question answering competition. We use a subset of questions with non-empty answers [1] from the TREC 2006 dataset [2]. The dataset provides a list of matching documents from the AQUAINT corpus and correct answers for each question. The dataset contains 387 questions; the AQUAINT corpus contains an average of 3.5 documents per ques-

[1] The questions where an answer was not in the dataset were not used in this analysis

[2] http://trec.nist.gov/data/qa/t2006_qadata.html

tion that contain the correct answer to that question. Using *correct answers* we find the *correct sentences* from the *matching documents*. We use this information as a gold standard for the IR task.

We index the documents in the AQUAINT corpus using the Lucene (Apache, 2004 2008) engine on the document level. We evaluate document retrieval using *gold standard* documents from the AQUAINT corpus. We evaluate sentence extraction on both AQUAINT and the Web automatically using regular expressions for correct answers provided by TREC.

In our experiments we use manually and automatically created phrases. Our automatically created phrases were obtained by extracting noun, verb and prepositional phrases and named entities from the question dataset using then NLTK (Bird et al., 2008) and Lingpipe (Carpenter and Baldwin, 2008) tools. Our manually created phrases were obtained by hand-correcting these automatic annotations (e.g. to remove extraneous words and phrases and add missed words and phrases from the questions).

4 System

For the experiments in this paper we use the *StoQA* system. This system employs a pipeline architecture with three main stages as illustrated in Figure 1: question analysis, document and sentence extraction (IR), and answer extraction. After the user poses a question, it is analyzed. Target named entities and semantic roles are determined. A query is constructed, tailored to the search tools in use. Sentences containing target terms are then extracted from the documents retrieved by the query. The candidate sentences are processed to identify and extract candidate answers, which are presented to the user.

We use the NLTK toolkit (Bird et al., 2008) for question analysis and can add terms to search queries using WordNet (Miller, 1995). Our system can currently retrieve documents from either the Web (using the Yahoo search API (Yahoo!, 2008)), or the AQUAINT corpus (Graff, 2002) (through the Lucene indexer and search engine (Apache, 2004 2008)). When using Lucene, we can assign different weights to different types of search term (e.g. less weight to terms than to named entities added to a query) (cf. (Lee and others, 2001)).

We currently have two modules for answer extraction, which can be used separately or together. Candidate sentences can be tagged with named entity information using the Lydia system (Lloyd et al., 2005). The tagged word/phrase matching the target named entity type most frequently found is chosen as the answer. Our system can also extract answers through semantic role labeling, using the SRL toolkit from (Punyakanok et al., 2008). In this case, the tagged word/phrase matching the target semantic role most frequently found is chosen as the answer.

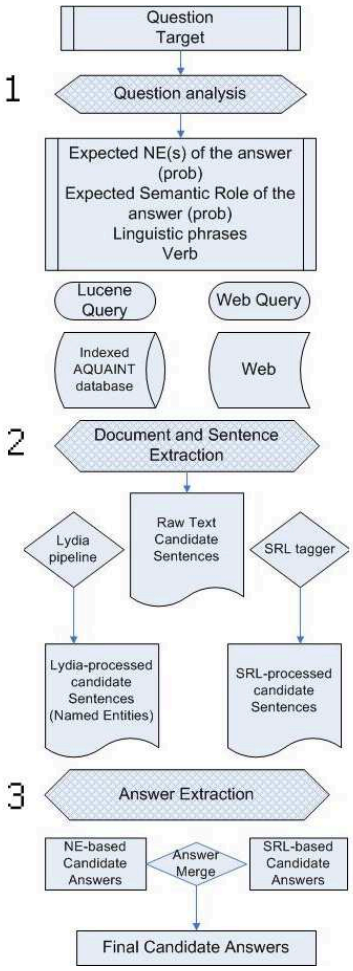

Figure 1: Architecutre of our question answering system

5 Method

5.1 Motivation

Question answering is an engineering-intensive task. System performance improves as more sophisticated techniques are applied to data processing. For example, the IR stage in question answering is shown to improve with the help of techniques like predictive annotations and relation extraction; matching of semantic and syntactic re-

Target Question	United Nations What was the number of member nations of the U.N. in 2000?
Named Entity	U.N., United Nations
Phrases	"member nations of the U.N."
Converted Q-phrase	"member nations of the U.N. in 2000"
Baseline Query	was the number of member nations of the U.N. in 2000 United Nations
Lucene Query with phrases and NE	was the number of member nations of the U.N. in 2000 "United Nations", "member nations of the u.n."
Cascaded web query	
query1	"member nations of the U.N. in 2000" AND (United Nations)
query2	"member nations of the u.n." AND (United Nations)
query3	(number of member nations of the U.N. in 2000) AND (United Nations)
query4	(United Nations)

Table 1: Question processing example: terms of a query

lations in a question and a candidate sentence are known to improve overall QA system performance (Prager et al., 2000; Stenchikova et al., 2006; Katz and Lin, 2003; Harabagiu et al., 2006; Chu-Carroll et al., 2006).

In this work we analyze less resource expensive techniques, such as chunking and named entity detection, for IR in question answering. Linguistic analysis in our system is applied to questions and to candidate sentences only. There is no need for annotation of all documents to be indexed, so our techniques can be applied to IR on large datasets such as the Web.

Intuitively, using phrases in query construction may improve retrieval precision. For example, if we search for *In what year did the movie win academy awards?* using a disjunction of words as our query we may match irrelevant documents about the military *academy* or Nobel prize *awards*. However, if we use the phrase *"academy awards"* as one of the query terms, documents with this term will receive a higher ranking. A counterargument for using phrases is that *academy* and *awards* are highly correlated and therefore the documents that contain both will be more highly ranked. We hypothesize that for phrases where constituents are not highly correlated, exact phrase extraction will give more benefit.

5.2 Search Query

We process each TREC question and target [3] to identify named entities. Often, the target is a complete named entity (NE), however, in some of the TREC questions the target contains a named entity, e.g. *tourists massacred at Luxor in 1997*, or *1991 eruption of Mount Pinatubo* with named entities *Luxor* and *Mount Pinatubo*. For the TREC question *What was the number of member nations of the U.N. in 2000?*, the identified constituents and automatically constructed query are shown in Table 1. **Named entities** are identified using Lingpipe (Carpenter and Baldwin, 2008), which identifies named entities of type *organization, location* and *person*. **Phrases** are identified automatically using the NLTK toolkit (Bird et al., 2008). We extract noun phrases, verb phrases and prepositional phrases. The rules for identifying phrases are mined from a dataset of manually annotated parse trees (Judge et al., 2006) [4]. **Converted Q-phrases** are heuristically created phrases that paraphrase the question in declarative form using a small set of rules. The rules match a question to a pattern and transform the question using linguistic information. For example, one rule matches *Who is|was NOUN|PRONOUN VBD* and converts it to *NOUN|PRONOUN is|was VBD*. [5]

[3] The TREC dataset also provides a *target topic* for each questions, and we include it in the query.

[4] The test questions are not in this dataset.

[5] Q-phrase is extracted only for who/when/where questions. We used a set of 6 transformation patterns in this experiment.

Named Entities	Phrases
great pyramids; frank sinatra; mt. pinatubo; miss america; manchester united; clinton administration	capacity of the ballpark; groath rate; security council; tufts university endowment; family members; terrorist organization

Table 2: Automatically identified named entities and phrases

A q-phrase represents how a simple answer is expected to appear, e. g. a **q-phrase** for the question *When was Mozart born?* is *Mozart was born.* We expect a low probability of encountering a **q-phrase** in retrieved documents, but a high probability of co-occurrence of q-phrases phrase with correct answers.

In our basic system (baseline), words (trivial query constituents) from question and target form the query. In the experimental system, the query is created from a combination of words, quoted exact phrases, and quoted named entities. Table 2 shows some examples of phrases and named entities used in queries. The goal of our analysis is to evaluate whether non-trivial query constituents can improve document and sentence extraction.

We use a back-off mechanism with both of our IR subsystems to improve document extraction. The Lucene API allows the user to create arbitrarily long queries and assign a weight to each query constituent. We experiment with assigning different weights based on the type of a query constituent. Assigning a higher weight to phrase constituents increases the scores for documents matching a phrase, but if no phrase matches are found documents matching lower-scored constituents will be returned.

The query construction system for the Web first produces a query containing only **converted q-phrases** which have low recall and high precision (query 1 in table 1). If this query returns less than 20 results, it then constructs a query using **phrases** (query 2 in table 1), if this returns less than 20 results, queries without exact phrases (queries 3 and 4) are used. Every query contains a conjunction with the question *target* to increase precision for the cases where the *target* is excluded from **converted q-phrase** or an **exact phrase**.

For both our IR subsystems we return a maximum of 20 documents. We chose this relatively low number of documents because our answer extraction algorithm relies on semantic tagging of candidate sentences, which is a relatively time-consuming operation.

The text from each retrieved documents is split into sentences using Lingpipe. The same sentence extraction algorithm is used for the output from both IR subsystems (AQUAINT/Lucene and Web/Yahoo). The sentence extraction algorithm assigns a score to each sentence according to the number of matched terms it contains.

5.3 Analysis of Constituents

For our analysis of the impact of different linguistic constituent types on document retrieval we use the TREC 2006 dataset which consists of questions, documents containing answers to each question, and *supporting sentences*, sentences from these documents that contain the answer to each question.

Table 3 shows the number of times each constituent type appears in a *supporting sentence* and the proportion of *supporting sentences* containing each constituent type (sent w/answer column). The "All Sentences" column shows the number of constituents in all sentences of candidate documents. The *precision* column displays the chance that a given sentence is a *supporting sentence* if a constituent of a particular type is present in it. *Converted q-phrase* has the highest precision, followed by phrases, verbs, and named entities. Words have the highest chance of occurrence in a *supporting sentence* (.907), but they also have a high chance of occurrence in a document (.745).

This analysis supports our hypothesis that using exact phrases may improve the performance of information retrieval for question answering.

6 Experiment

In these experiments we look at the impact of using exact phrases on the performance of the document retrieval and sentence extraction stages of question answering. We use our *StoQA* question answering system. Questions are analyzed as described in the previous section. For document retrieval we use the back-off method described in the previous sec-

	sent w/ answer		all sentences		precision
	num	proportion	num	proportion	
Named Entity	907	0.320	4873	0.122	.18
Phrases	350	0.123	1072	0.027	.34
Verbs	396	0.140	1399	0.035	.28
Q-Phrases	11	0.004	15	0.00038	.73
Words	2573	0.907	29576	0.745	.086
Total Sentences	2836		39688		

Table 3: Query constituents in sentences of correct documents

	avg doc recall	avg doc MRR	overall doc recall	avg sent MRR	overall sent recall	avg corr sent in top 1	avg corr sent in top 10	avg corr sent in top 50
IR with Lucene on AQUAINT dataset								
baseline (words disjunction from target and question)	0.530	0.631	0.756	0.314	0.627	0.223	1.202	3.464
baseline + auto phrases	0.514	0.617	0.741	0.332	0.653	0.236	1.269	3.759
words + auto NEs & phrases	0.501	0.604	0.736	0.316	0.653	0.220	1.228	3.705
baseline + manual phrases	0.506	0.621	0.738	0.291	0.609	0.199	1.231	3.378
words + manual NEs & phrases	0.510	0.625	0.738	0.294	0.609	0.202	1.244	3.368
IR with Yahoo API on WEB								
baseline words disjunction	-	-	-	0.183	0.570	0.101	0.821	2.316
cascaded using auto phrases	-	-	-	0.220	0.604	0.140	0.956	2.725
cascaded using manual phrases	-	-	-	0.241	0.614	0.155	1.065	3.016

Table 4: Document retrieval evaluation.

tion. We performed the experiments using first automatically generated phrases, and then manually corrected phrases.

For document retrieval we report: 1) average recall, 2) average mean reciprocal ranking (MRR), and 3) overall document recall. Each question has a document retrieval recall score which is the proportion of documents identified from all correct documents for this question. The *average recall* is the individual recall averaged over all questions. MRR is the inverse index of the first correct document. For example, if the first correct document appears second, the MRR score will be 1/2. MRR is computed for each question and averaged over all questions. *Overall document recall* is the percentage of questions for which at least one correct document was retrieved. This measure indicates the upper bound on the QA system.

For sentence retrieval we report 1) average sentence MRR, 2) overall sentence recall, 3) average precision of the first sentence, 4) number of correct candidate sentences in the top 10 results, and 5) number of correct candidate sentences in the top 50 results [6].

Table 4 shows our experimental results. First, we evaluate the performance of document retrieval on the indexed AQUAINT dataset. Average document recall for our baseline system is 0.53, indicating that on average half of the correct documents are retrieved. Average document MRR is .631, meaning that on average the first correct document appears first or second. Overall document recall indicates that 75.6% of queries contain a correct document among the retrieved documents. Average sentence recall is lower than document recall indicating that some proportion of correct answers is not retrieved using our heuristic sentence extraction algorithm. The average sentence MRR is .314 indicating that the first correct sentence is approximately third on the list. With

[6] Although the number of documents is 20, multiple sentences may be extracted from each document.

the AQUAINT dataset, we notice no improvement with exact phrases.

Next, we evaluate sentence retrieval from the WEB. There is no *gold standard* for the WEB dataset so we do not report document retrieval scores. Sentence scores on the WEB dataset are lower than on the AQUAINT dataset [7].

Using back-off retrieval with automatically created phrases and named entities, we see an improvement over the baseline system performance for each of the sentence measures on the WEB dataset. Average sentence MRR increases 20% from .183 in the baseline to .220 in the experimental system. With manually created phrases MRR improves a further 9.5% to .241. This indicates that information retrieval on the WEB dataset can benefit from a better quality of chunker and from a properly converted question phrase. It also shows that the improvement is not due to simply matching random substrings from a question, but that linguistic information is useful in constructing the exact match phrases. Precision of automatically detected phrases is affected by errors during automatic part-of-speech tagging of questions. An example of an error due to POS tagging is the identification of a phrase *was Rowling born* due to a failure to identify that *born* is a *verb*.

Our results emphasize the difference between the two datasets. AQUAINT dataset is a collection of a large set of news documents, while WEB is a much larger resource of information from a variety of sources. It is reasonable to assume that on average there are much fewer documents with query words in AQUAINT corpus than on the WEB. Proportion of *correct documents* from all retrieved WEB documents on average is likely to be lower than this proportion in documents retrieved from AQUAINT. When using words on a query to AQUAINT dataset, most of the *correct documents* are returned in the top matches. Our results indicate that over 50% of *correct documents* are retrieved in the top 20 results. Results in table 3 indicate that exactly matched phrases from a question are more precise predictors of presence of an answer. Using exact matched phrases in a WEB query allows a search engine to give higher rank to more relevant documents and increases likelihood of these documents in the top 20 matches.

Although overall performance on the WEB dataset is lower than on AQUAINT, there is a potential for improvement by using a larger set of documents and improving our sentence extraction heuristics.

[7] Our decision to use only 20 documents may be a factor.

7 Conclusion and Future Work

In this paper we present a document retrieval experiment on a question answering system. We evaluate the use of named entities and of noun, verb, and prepositional phrases as exact match phrases in a document retrieval query. Our results indicate that using phrases extracted from questions improves IR performance on WEB data. Surprisingly, we find no positive effect of using phrases on a smaller closed set of data.

Our data analysis shows that linguistic phrases are more accurate indicators for candidate sentences than words. In future work we plan to evaluate how phrase type (noun vs. verb vs. preposition) affects IR performance.

Acknowledgment

We would like to thank professor Amanda Stent for suggestions about experiments and proofreading the paper. We would like to thank the reviewers for useful comments.

References

Apache. 2004-2008. Lucene. http://lucene.apache.org/java/docs/index.html.

Bilotti, M., B. Katz, and J. Lin. 2004. What works better for question answering: Stemming or morphological query expansion? In *Proc. SIGIR*.

Bird, S., E. Loper, and E. Klein. 2008. Natural Language ToolKit (NLTK). http://nltk.org/index.php/Main_Page.

Carpenter, B. and B. Baldwin. 2008. Lingpipe. http://alias-i.com/lingpipe/index.html.

Chu-Carroll, J., J. Prager, K. Czuba, D. Ferrucci, and P. Duboue. 2006. Semantic search via XML fragments: a high-precision approach to IR. In *Proc. SIGIR*.

Clarke, C., G. Cormack, D. Kisman, and T. Lynam. 2000. Question answering by passage selection (multitext experiments for TREC-9). In *Proc. TREC*.

Collins-Thompson, K., J. Callan, E. Terra, and C. L.A. Clarke. 2004. The effect of document retrieval quality on factoid question answering performance. In *Proc. SIGIR*.

Dang, H., J. Lin, and D. Kelly. 2006. Overview of the TREC 2006 question answering track. In *Proc. TREC*.

Graff, D. 2002. The AQUAINT corpus of English news text. Technical report, Linguistic Data Consortium, Philadelphia, PA, USA.

Harabagiu, S., A. Hickl, J. Williams, J. Bensley, K. Roberts, Y. Shi, and B. Rink. 2006. Question answering with LCC's CHAUCER at TREC 2006. In *Proc. TREC*.

Hovy, E., L. Gerber, U. Hermjakob, M. Junk, and C.-Y. Lin. 2001a. Question answering in Webclopedia. In *Proc. TREC*.

Hovy, E., U. Hermjakob, and C.-Y. Lin. 2001b. The use of external knowledge in factoid QA. In *Proc. TREC*.

Ittycheriah, A., M. Franz, and S. Roukos. 2001. IBM's statistical question answering system – TREC-10. In *Proc. TREC*.

Judge, J., A. Cahill, and J. van Genabith. 2006. QuestionBank: Creating a corpus of parse-annotated questions. In *Proc. ACL*.

Katz, B. and J. Lin. 2003. Selectively using relations to improve precision in question answering. In *Proc. of the EACL Workshop on Natural Language Processing for Question Answering*.

Lee, G. G. et al. 2001. SiteQ: Engineering high performance QA system using lexico-semantic pattern matching and shallow NLP. In *Proc. TREC*.

Light, M., G. S. Mann, E. Riloff, and E. Breck. 2001. Analyses for elucidating current question answering technology. *Journal of Natural Language Engineering*, 7(4).

Llopis, F. and J. L. Vicedo. 2001. IR-n: A passage retrieval system at CLEF-2001. In *Proc. of the Second Workshop of the Cross-Language Evaluation Forum (CLEF 2001)*.

Lloyd, L., D. Kechagias, and S. Skiena. 2005. Lydia: A system for large-scale news analysis. In *Proc. SPIRE*, pages 161–166.

Miller, George A. 1995. WordNet: a lexical database for english. *Communications of the ACM*, 38(11).

Murdock, V. and W. B. Croft. 2005. Simple translation models for sentence retrieval in factoid question answering. In *Proc. SIGIR*.

Prager, J., E. Brown, and A. Coden. 2000. Question-answering by predictive annotation. In *ACM SIGIR*. QA -to site.

Punyakanok, V., D. Roth, and W. Yih. 2008. The importance of syntactic parsing and inference in semantic role labeling. *Computational Linguistics*, 34(2).

Srihari, R. and W. Li. 1999. Information extraction supported question answering. In *Proc. TREC*.

Stenchikova, S., D. Hakkani-Tur, and G. Tur. 2006. QASR: Question answering using semantic roles for speech interface. In *Proc. ICSLP-Interspeech 2006*.

Tellex, S., B. Katz, J. Lin, A. Fernandes, and G. Marton. 2003. Quantitative evaluation of passage retrieval algorithms for question answering. In *Proc. SIGIR*.

Vorhees, V. and D. Harman. 1999. Overview of the eighth Text REtrieval Conference (TREC-8). In *"Proc. TREC"*.

White, K. and R. Sutcliffe. 2004. Seeking an upper bound to sentence level retrieval in question answering. In *Proc. SIGIR*.

Yahoo!, Inc. 2008. Yahoo! search API. http://developer.yahoo.com/search/.