Statistical Source Expansion for Question Answering

Nico Schlaefer
School of Computer Science
Carnegie Mellon University
nico@cs.cmu.edu

Jennifer Chu-Carroll
IBM T.J. Watson
Research Center
jencc@us.ibm.com

Eric Nyberg
School of Computer Science
Carnegie Mellon University
ehn@cs.cmu.edu

James Fan
IBM T.J. Watson
Research Center
fanj@us.ibm.com

Wlodek Zadrozny
IBM T.J. Watson
Research Center
wlodz@us.ibm.com

David Ferrucci
IBM T.J. Watson
Research Center
ferrucci@us.ibm.com

ABSTRACT

A source expansion algorithm automatically extends a given text corpus with related content from large external sources such as the Web. The expanded corpus is not intended for human consumption but can be used in question answering (QA) and other information retrieval or extraction tasks to find more relevant information and supporting evidence. We propose an algorithm that extends a corpus of seed documents with web content, using a statistical model to select text passages that are both relevant to the topics of the seeds and complement existing information.

In an evaluation on 1,500 hand-labeled web pages, our algorithm ranked text passages by relevance with 81% MAP, compared to 43% when relying on web search engine ranks alone and 75% when using a multi-document summarization algorithm. Applied to QA, the proposed method yields consistent and significant performance gains. We evaluated the impact of source expansion on over 6,000 questions from the Jeopardy! quiz show and TREC evaluations using Watson, a state-of-the-art QA system. Accuracy increased from 66% to 71% on Jeopardy! questions and from 59% to 64% on TREC questions.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Algorithms, Experimentation, Performance

1. INTRODUCTION

In question answering (QA), it is common to locally store and index document collections that provide good coverage of the information required for a given question domain. For instance, encyclopedias and dictionaries are useful sources for answering trivia questions, and newswire corpora provide relevant information about politics and the economy. However, these resources may not contain the answers to all questions, and answers may be hard to find if there is little redundancy in the sources. To improve the coverage of local sources and to facilitate the extraction and validation of answers, the sources can be expanded automatically with related information from large external text corpora such as the Web. In this paper, we present a fully implemented statistical approach to source expansion and compare it to several baselines for selecting relevant source content. We further demonstrate that source expansion consistently and significantly improves QA performance on large datasets drawn from the Jeopardy! quiz show and TREC evaluations.

We decompose the source expansion (SE) task into a four-stage process. Given a corpus of seed documents, our approach (1) retrieves, for each seed, related documents from the Web or other external sources, (2) extracts paragraph- or sentence-length text nuggets from the retrieved documents, (3) estimates the relevance of the nuggets with regard to the seed document using a statistical model, and (4) compiles a new pseudo-document from the most relevant text. The pseudo-documents can be indexed along with the seeds, yielding a larger corpus with increased coverage and reformulations of existing information. Note that this expansion is performed only once in a preprocessing step and does not require specific knowledge of the questions that will be asked at QA runtime.

SE is an effective method for improving QA performance because it addresses the most common types of failures in state-of-the-art systems:

1. *Source failures*, i.e. the knowledge sources do not contain the information sought by a question. SE helps by adding more relevant content.

2. *Search and candidate extraction failures*, i.e. the system cannot retrieve or extract a correct answer, often because of insufficient keyword overlap with the question. SE helps by adding paraphrases of existing information.

3. *Answer selection failures*, i.e. the answer is outscored by a wrong answer, often because of insufficient supporting evidence in the sources or because the relevant search results ranked too low. SE again helps by adding reformulations and increasing redundancy.

CIKM'11, October 24–28, 2011, Glasgow, Scotland, UK.

In contrast, by performing query expansion or using pseudo-relevance feedback (PRF) in the search phase of the QA system, one can (at most) address failures of types 2 and 3. In practice, these approaches may introduce noise and hurt performance [28]. Often, they are only applied as a fallback solution if an initial query yields low recall [19, 3].

Source expansion performed as a preprocessing step can also be preferable over live web searches at QA runtime. While web search engines must be used as black boxes, local sources can be indexed with open-source IR systems such as Indri[1] or Lucene[2], which allow full control over the retrieval model. Local sources can also be preprocessed and annotated with syntactic and semantic information, which can be leveraged to formulate queries that better describe the information need expressed in a question [30, 6]. Furthermore, in applications where speed and availability matter, where the knowledge sources contain confidential data or restricted-domain knowledge, or where a self-contained system is required, a live web search may be infeasible. Local sources also guarantee reproducible results, whereas the Web and search engines change constantly. Finally, local sources can be consolidated e.g. by merging related content into a single document and removing noise.

Source expansion has many potential applications beyond QA. For instance, the pseudo-documents generated through SE could be added to the document representations of the seeds to improve retrieval accuracy of a traditional document retrieval system. The expanded documents could also be used by a relation extraction algorithm to validate instances of relations found in the seeds and to find additional instances. This paper, however, focuses on QA as one example of a task where SE helps consistently. We show that SE significantly improves the performance of Watson [17], one of the most effective QA systems to date, yielding gains of 4.2%–8.6% in search recall and 7.6%–12.9% in QA accuracy on large datasets of Jeopardy! and TREC questions.

2. RELATED WORK

Source expansion is related to a number of NLP tasks, including content-based information filtering, topic tracking, multi-document summarization and definitional QA. The problem of extracting text nuggets from documents that relate to a given topic is perhaps most similar to multi-document summarization (e.g. [18, 27]) but differs in important ways: (1) the extraction is guided by the content of a seed document which is used to model topicality, (2) while we avoid lexical redundancy, semantically redundant text that phrases the same information differently is desirable, and (3) the generated summaries need not be coherent as they are not intended for human consumption.

Recent research on definitional QA has led to various algorithms for compiling relevant texts on topics such as people, organizations or events (e.g. [7, 32]). However, these algorithms generate answers to individual questions from existing sources at QA runtime, whereas the proposed method compiles new source material that can be used by a QA system to answer a wide range of questions with little computational overhead.

QA systems often use the Web as a large, redundant knowledge source [14, 16], but it has been noted that there are situations where a local search is preferable [13, 22]. Clarke et al. [13] evaluated the performance impact of locally indexed web crawls on TREC QA data. They found that large crawls of over 50 GB were required to outperform the 3 GB reference corpus used in TREC, and that performance degraded if the crawl exceeded about 500 GB. Our approach improves on earlier work by using statistical models to reduce the size of the retrieved web data by two orders of magnitude and to filter out noise that may hurt QA performance.

Balasubramanian and Cucerzan [4] propose an algorithm for generating documents about given topics from web content. The usefulness of sentences extracted from web pages is determined with aspect models built from web search logs. These aspect models consist of terms that frequently co-occur with a given topic or related topics in the query logs. In contrast, our SE method leverages the content of existing seed corpora to model topicality. While query logs may be hard to come by, particularly when starting out in a new domain (e.g. medical or legal search), suitable seed corpora are often readily available (e.g. medical encyclopedias, legal dictionaries).

A variety of techniques have been proposed for expanding text documents to facilitate their retrieval if the queries do not exactly match the terminology used in those documents. For example, documents can be augmented with terms extracted from related documents retrieved from the same collection or an auxiliary corpus. It has been shown that document expansion improves retrieval performance on various tasks, including spoken document retrieval [29], cross-lingual IR [24] and topic tracking [23]. When indexing a collection of web pages, the anchor texts associated with hyperlinks provide high-level descriptions of the linked documents and can be added to their index representations [15]. Similarly, in a scientific corpus, terms that appear in the vicinity of citations can be associated with the referenced articles [8]. Conversely, the content of linked or cited documents can be propagated to the documents that contain the references [25].

Similarly to query expansion and PRF, these document expansion techniques are unlikely to help if the information sought by an IR system is missing in the corpus (failure type 1 in the introduction). In contrast, the proposed SE method augments a seed corpus with additional information that can be used by applications such as QA, increasing the amount of useful content several-fold. SE can also help if the expanded sources are used directly by an information extraction system (e.g. for relation extraction) without performing a search.

3. APPROACH

The input of the source expansion (SE) algorithm is a collection of documents in which each document contains information about a distinct topic. We refer to each of the documents as a *seed document* or simply *seed*, and to the entire collection as the *seed corpus*. Examples of pre-existing seed corpora that are suitable for SE are encyclopedias (such as Wikipedia) and dictionaries (such as Wiktionary). For each seed, a new pseudo-document is generated that contains related information retrieved from large external resources. By expanding the seed documents, we gather additional relevant content about their topics, as well as paraphrases of information that was already covered. A question answering

[1] http://www.lemurproject.org/indri/
[2] http://lucene.apache.org/

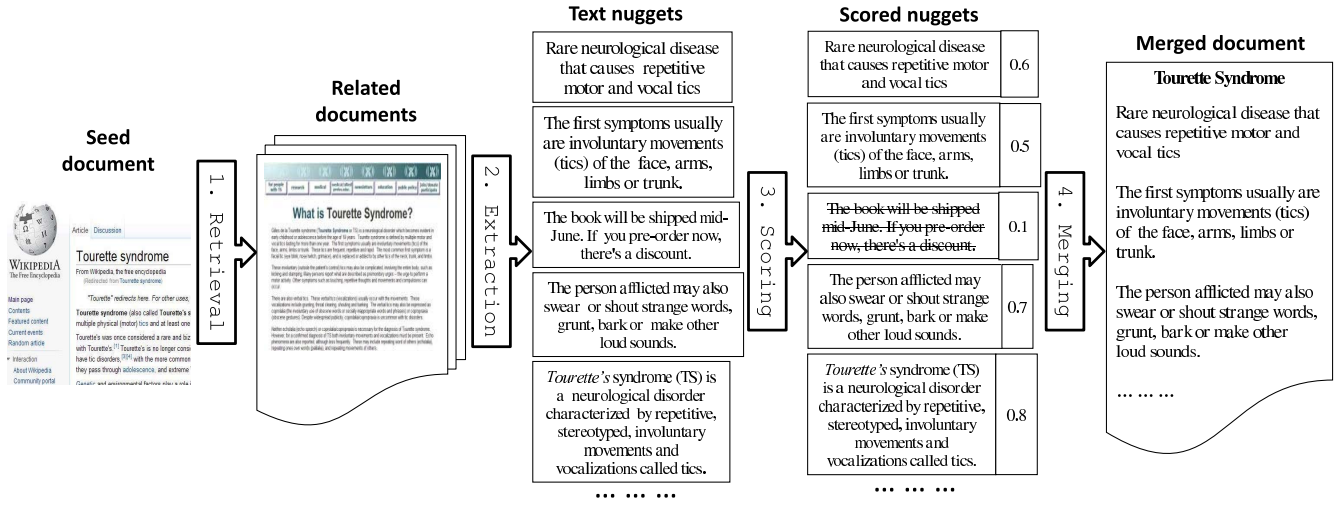

Figure 1: Four-stage source expansion pipeline processing the Wikipedia seed about *Tourette Syndrome*.

system can benefit both from the increased coverage and the added semantic redundancy.

Seed documents are expanded in a four-stage pipeline, illustrated in Figure 1 using the Wikipedia article about *Tourette Syndrome* as an example. For each seed, the SE system retrieves related documents from an external source (*retrieval* stage in Figure 1). We used the Web as a large source of related content for all experiments reported in this paper. The retrieved documents are divided into paragraph- or sentence-length text nuggets (*extraction* stage), and their relevance to the topic of the seed is estimated using a statistical model (*scoring* stage). Finally, a pseudo-document is compiled from the most relevant nuggets (*merging* stage). Note that this expansion is performed as a separate preprocessing step, and the expanded sources can then be used in conjunction with the seed corpora by a QA system and other information retrieval or extraction applications.

3.1 Retrieval

For each seed, the retrieval component generates a query, performs a Yahoo! search for related content, and fetches up to 100 web pages linked from the highest ranking search results. In experiments with Wikipedia and Wiktionary, we used the document titles as queries since they already provide fully disambiguated descriptions of the topics. However, queries can also be generated from documents that do not have descriptive titles by extracting topical terms from their bodies based on markup information or using statistical techniques.

3.2 Extraction

The extraction component splits the retrieved web documents into paragraph-length text nuggets. For HTML documents, structural markup can be used to determine boundaries. Typical text nuggets are HTML paragraphs, list items or table cells. They range in length from short fragments (e.g. *born: 1968*) to narratives of multiple sentences. Since in the merging stage individual nuggets are selected and added to the pseudo-document, the text nuggets should ideally be self-contained (i.e. meaningful on their own) and either entirely relevant or irrelevant. Nuggets that are delimited by structural markup are mostly self-contained, but are often only partially relevant. Thus we further split the nuggets into sentences and experimented with both markup-based nuggets and sentence-level nuggets. In Section 4.2 we present evaluation results for both granularities.

3.3 Scoring

The core component of our SE approach is a statistical model that scores the extracted text nuggets based on their relevance to the topic of the seed document. A large dataset of manually labeled nuggets described in Section 4.1 was used to fit the relevance model. We experimented with various features that estimate the topicality or textual quality of nuggets and are thus predictive of their relevance. In Table 1 we briefly describe and motivate each feature, and specify its type (*topicality*, *search* or *surface* feature), its range and whether it is generated at the level of *documents* or individual *nuggets*. The most predictive topicality features (*TopicLRSeed*, *TFIDFSeed* and *MMR*) utilize the body of the seed document, which provides useful information about the topicality of text nuggets that is not available in definitional QA or summarization tasks. For the features that are based on language models (*TopicLRSeed* and *TopicLRNuggets*), we used simple unigrams with Good-Turing discounting since often not much data is available to fit the topic models.

We first fitted a logistic regression (LR) model [1] using these features to estimate the relevance of each text nugget independently. However, we found that text nuggets should not be evaluated independently since they are more likely to be relevant if the surrounding text is relevant. Thus we relaxed the independence assumption by adding features of adjacent nuggets to the LR model. More precisely, in addition to the original relevance features, we added the nugget-level features (i.e. all features except *DocRank*) of the previous nugget and the next nugget, and fitted a model using this larger feature set. In Section 4.2 we show that this simple extension improves nugget ranking performance.

3.4 Merging

The merging component ranks the text nuggets by their relevance scores in descending order. A filter reduces lexical redundancy by removing nuggets whose keywords are sub-

Table 1: Features for estimating the relevance of text nuggets to the topic of a seed document.

Feature (Type)	Description (Range, Level)	Motivation
TopicLRSeed *(Topicality)*	Likelihood ratio of the nugget estimated with topic and background language models. The topic model is estimated from the seed document, the background model from a large sample of Wikipedia articles. *(continuous, per nugget)*	Nuggets with large likelihood ratios are often thematically related to the seed.
TopicLRNuggets *(Topicality)*	Similar to *TopicLRSeed*, but the topic model is estimated from the nuggets that were retrieved for the given topic, and the background model from a large sample of nuggets retrieved for different topics. *(continuous, per nugget)*	More stable than *TopicLRSeed* for short seed documents.
TFIDFSeed *(Topicality)*	Average *tf.idf* score of the tokens in the nugget. The *tf* scores are estimated from the seed document, the *idf* scores from a large sample of Wikipedia articles. *(continuous, per nugget)*	Tokens with high *tf.idf* scores are often central to the topic.
TFIDFNuggets *(Topicality)*	Similar to *TFIDFSeed*, but the *tf* scores are estimated from the nuggets that were retrieved for the given topic, and the *idf* scores from a large sample of nuggets retrieved for different topics. *(continuous, per nugget)*	More stable than *TFIDFSeed* for short seed documents.
MMR *(Topicality)*	Relevance score from the maximal marginal relevance (MMR) summarization algorithm with $\lambda = 1$ [10, 18]. Originally used as a baseline, then adopted as an additional feature. See Section 4.2 for details. *(continuous, per nugget)*	Nuggets with large MMR scores have a similar word distribution as the seed and are thus likely to be topical.
QueryTerms *(Topicality)*	Fraction of the terms in the query used in the retrieval stage that occur in the text nugget. Terms are weighted with *idf* scores that are normalized to sum to 1. *(continuous, per nugget)*	Nuggets that contain rare query terms are more likely to be topical.
3PPronoun *(Topicality)*	Whether the nugget contains a third person pronoun. *(binary, per nugget)*	Third person pronouns often refer to the topic.
DocRank *(Search)*	Yahoo! search rank of the web page the nugget was extracted from. *(discrete, per document)*	Documents with high ranks usually contain more relevant text.
AbstractCoverage *(Search)*	Fraction of the tokens in the Yahoo! search abstract covered by the nugget. *(continuous, per nugget)*	The abstract is the search engine's view of the most relevant text.
Known3Grams *(Surface)*	Fraction of token 3-grams found in a dictionary of known 3-grams. *(continuous, per nugget)*	High-quality English text should contain many known 3-grams.
NuggetLength *(Surface)*	Length of the nugget measured in tokens. *(discrete, per nugget)*	Very short nuggets are usually not relevant.
NuggetOffset *(Surface)*	Offset of the nugget in the document measured in nuggets. *(discrete, per nugget)*	Text at the beginning of a document is often more relevant.
TypeTokenRatio *(Surface)*	Ratio of the number of distinct tokens over the total number of tokens in the nugget. *(continuous, per nugget)*	Penalizes nuggets that repeat topical terms over and over.
SpecialCharRatio *(Surface)*	Ratio of the number of special characters over the total number of characters in the nugget. *(continuous, per nugget)*	Penalizes nuggets that are not well-formed English text.

sumed by higher ranking nuggets or the seed. Nuggets are also dropped if their relevance scores are below an absolute threshold, or if the total character length of all nuggets exceeds a threshold that is relative to the length of the seed. In Section 5.2 we describe how thresholds were chosen in our experiments. The remaining nuggets are compiled into a pseudo-document that can be indexed and searched along with the seed.

3.5 Examples

To better understand how source expansion improves QA performance, consider the TREC question *What is the name of the rare neurological disease with symptoms such as: involuntary movements (tics), swearing, and incoherent vocalizations (grunts, shouts, etc.)?* The expanded version of the Wikipedia article about *Tourette syndrome* in Figure 1 contains the following nuggets, which originated from different web pages and jointly almost perfectly cover the question terms (underlined):

- *Rare neurological disease that causes repetitive motor and vocal tics*
- *The first symptoms usually are involuntary movements (tics) of the face, arms, limbs or trunk.*
- *Tourette's syndrome is a neurological disorder characterized by repetitive, stereotyped, involuntary movements and vocalizations called tics.*
- *The person afflicted may also swear or shout strange words, grunt, bark or make other loud sounds.*

This example supports the claim in Section 1 that SE helps consolidate sources by merging related content into a single document. The expanded document is retrieved by Watson, enabling it to correctly answer the question. Note that this article was expanded along with many other Wikipedia articles in a preprocessing step, without knowing yet which articles will be relevant for answering questions.

Now consider the TREC question *When were the first postage stamps issued in the United States?* The article on *Postage stamp* in our copy of Wikipedia only mentions the year (*1847*) and the relevant passage does not match the question well enough to be retrieved by Watson. However, the expanded version of the article contains the snippet *on*

<DOC>
<TITLE>Postage stamp</TITLE>
<TEXT>
<NUGGET SCORE="0.9942" URL="http://www.answers.com/topic/postage-stamp">Postage stamp, government stamp affixed to mail to indicate payment of postage. The term includes stamps printed or embossed on postcards and envelopes as well as the adhesive labels. The use of adhesive postage stamps was advocated by Sir Rowland Hill; it was adopted in Great Britain in 1839. Zuerich (Switzerland) and Brazil issued stamps in 1843, and by 1850 the custom had spread throughout the world. Although the postmasters of several cities had previously issued provisional stamps, the first U.S. official issue was in 1847. [...]</NUGGET>
<NUGGET SCORE="0.9718" URL="http://www.answers.com/topic/postage-stamp">The postage stamp is a relatively modern invention, first proposed in 1837 when Sir Rowland Hill, an English teacher and tax reformer, published a seminal pamphlet entitled Post Office Reform: Its Importance and Practicability. Among other reforms, Hill's treatise advocated that the English cease basing postal rates on the distance a letter traveled and collecting fees upon delivery. Instead, he argued, they should assess fees based on weight and require prepayment in the form of stamps. [...]</NUGGET>
<NUGGET SCORE="0.9512" URL="http://www.edinformatics.com/inventions_inventors/postage_stamp.htm">The adhesive postage stamp and the uniform postage rate were devised in Great Britain by James Chalmers around 1834. The same ideas were brought forward by Lovrenc Kosir, a Slovenian postal clerk at the Viennese court in 1835, but did not meet a favorable response. [...]</NUGGET>
[...]
<NUGGET SCORE="0.3312" URL="http://www.postalmuseum.si.edu/resources/6a2p_1847s.html">This summer the National Postal Museum celebrates the anniversary of a watershed event in America's postal history. One hundred and fifty years ago, on July 1, 1847, the first federal United States postage stamps were issued in New York City. [...]</NUGGET>
[...]
</TEXT>
</DOC>

Figure 2: Pseudo-document for the Wikipedia seed article about *Postage stamp* (abbreviated).

July 1, 1847, the first federal United States postage stamps were issued, which perfectly matches the question and enables Watson to retrieve the complete answer. An abbreviated version of the pseudo-document that includes the snippet is shown in Figure 2. This example illustrates how SE adds new information to a corpus, addressing source failures (type 1 in Section 1), and how reformulations facilitate the retrieval of relevant text, mitigating search failures (type 2).

4. INTRINSIC EVALUATION

4.1 Dataset

A large dataset of manually annotated web pages was created to evaluate different relevance estimation strategies. For a sample of 15 Wikipedia articles about people, things and events, we retrieved 100 web pages each and presented them to human annotators with instructions to select relevant substrings. The annotators were given guidelines in the form of a checklist for making consistent decisions. A nugget was considered relevant if any of its tokens were selected by an annotator. Table 2 lists the topics along with the total number of extracted markup-based nuggets and the number of nuggets labeled as relevant. The agreement between annotators was high, with κ scores of 0.9085 to 0.9379.

Table 2: Summary of relevance estimation dataset.

Topic	# Nuggets	# Relevant
Amy Van Dyken	8,502	969 (11.4%)
Anne Frank	11,360	830 (7.3%)
Berlin Wall	10,280	1,300 (12.6%)
Fort Boise	8,756	73 (0.8%)
Harry Blackmun	9,750	641 (6.6%)
Iran-Iraq War	15,193	1,929 (12.7%)
Jenny Toomey	10,789	145 (1.3%)
John Rolfe	10,490	283 (2.7%)
José de San Martín	11,984	206 (1.7%)
Karriem Riggins	10,120	105 (1.0%)
Lincoln assassination	11,632	542 (4.7%)
Mother Teresa	11,507	1,383 (12.0%)
Ross Powless	13,410	97 (0.7%)
Vasco Núñez de Balboa	7,939	508 (6.4%)
XYZ Affair	12,726	210 (1.7%)
All	**164,438**	**9,221 (5.6%)**

4.2 Experiments

Using this dataset, we evaluated different relevance models through 15-fold cross-validation (one fold for each topic). The nuggets were ranked by relevance estimates in descending order, and performance was measured in terms of mean average precision (*MAP*) and micro-averaged precision-recall curves. Precision and recall were computed at the level of tokens rather than text nuggets so that results are comparable across different nugget granularities. For instance, precision is the percentage of tokens up to a given cutoff point in the ranking that were annotated as relevant. We present evaluation results on sentence-level nuggets (*Sentence*) and markup-based nuggets (*Markup*) for the independent logistic regression model (*LR Independent*), the model with features of adjacent instances (*LR Adjacent*) and these baselines:

- *Random.* Nuggets are ranked randomly. We report the expected performance if each token is treated as an independent nugget.
- *Round Robin.* Selects the first nugget from all documents, followed by the second nugget, and so forth, assuming that relevant text is more concentrated at the top of documents.

Table 3: MAP of baselines and statistical models.

Model	MAP	
	Sentence	Markup
Random	28.48%	28.48%
Round Robin	32.59%	35.78%
Search Rank	42.67%	42.67%
MMR	61.46%	74.75%
LR Independent	71.95%	79.69%
LR Adjacent	**77.19%**	**80.59%**

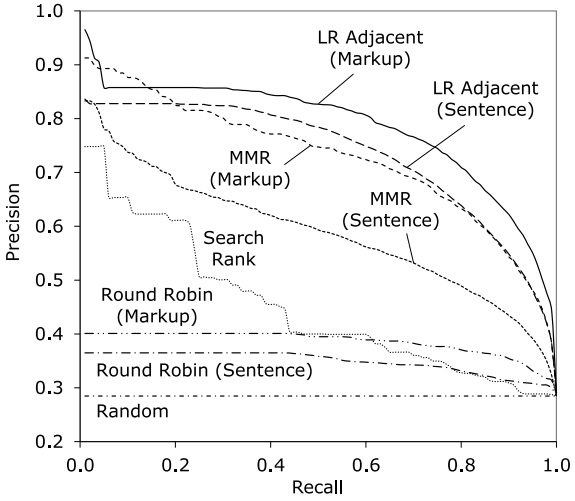

Figure 3: Precision-recall curves for baselines and statistical models.

- *Search Rank.* Preserves the ranking of the documents by the search engine and the order of the text nuggets within the documents.
- *MMR.* The maximal marginal relevance (MMR) algorithm [10, 18] is used to rank the text nuggets. MMR is one of the most effective algorithms for multi-document summarization. It iteratively selects text passages with high "marginal relevance", i.e. passages that are relevant to a query and add novel information. We used the full seed documents as queries and set the parameter $\lambda = 1$, i.e. the algorithm selects the most relevant nuggets regardless of their novelty. This setup worked best in a preliminary study.

Table 3 shows MAP scores for all ranking strategies and Figure 3 illustrates their performance in terms of precision-recall curves. Note that *Random* breaks down nuggets into tokens and *Search Rank* preserves the original order of the nuggets, and thus these strategies do not depend on the size of the nuggets. Among the baselines, MMR is clearly most effective, followed with some distance by the search engine rankings. The LR models outperform the baselines, with the only exception that MMR-based rankings have higher precision on markup-based nuggets at very low recall levels (too low to generate expanded documents of reasonable length). The independent nugget scoring model performs worse than the model with features of adjacent instances and has been omitted in the precision-recall plot for ease of presentation. The results also indicate that it is more effective to rank the longer markup-based text nuggets, even though they are often only partially relevant. This is because some features, such as the likelihood ratios and average *tf.idf* scores, are less reliable for short nuggets.

When adopting the source expansion approach, one could initially use the *MMR* baseline to rank text nuggets since it is easy to implement and does not require training data. Its performance (74.75% MAP on markup-based nuggets) comes relatively close to the best statistical model (80.59%), but the baseline solely relies on the seed content and is only effective if the seeds are long and of high quality. Other features that are predictive on their own include *TopicLRSeed* (69.40%) and *AbstractCoverage* (50.93%).

Table 4: Summary of QA datasets.

Regular Jeopardy!	Final Jeopardy!	TREC 8–12 Factoid
3,508 questions (66 episodes)	788 questions (1 per episode)	2,137 questions (no *NIL* questions)

5. APPLICATION TO QA

5.1 Datasets

The source expansion approach was evaluated on large datasets of Jeopardy! and TREC questions. A summary of the datasets is given in Table 4.

Jeopardy! questions and their answers were retrieved from J! Archive[3]. Most of these questions ask for factoid answers that can be extracted from text corpora, but there are also puzzles, word plays and puns that require additional processing and specialized inference. In each Jeopardy! episode, human contestants compete in answering up to 60 regular questions (sometimes not all questions are revealed) and one Final Jeopardy! question. Most questions in Jeopardy! only have a single correct answer. In the following experiments we use both regular Jeopardy! and Final Jeopardy! datasets. Final Jeopardy! questions are usually harder, both for human players and Watson, which will be reflected in the results. Questions with audio or visual clues were excluded.

For experiments with TREC datasets, we used independent factoid questions from the TREC 8–12 evaluations [31]. The questions and corresponding answer keys were obtained from the NIST website[4]. *NIL* questions without known answers were removed. TREC questions often have more than one acceptable answer, and the answer keys only cover answers that were found in the reference corpora used in the evaluations. When we evaluated end-to-end QA results, assessors who were not involved in the development of the approach judged the top answers returned by Watson and extended the answer keys with additional correct answers. However, when measuring search performance, we did not further extend the answer keys because it was impractical to judge all search results manually.

5.2 Sources

We expanded two sources that are both useful for Jeopardy! and TREC question answering: Wikipedia and the online dictionary Wiktionary[5]. Wikipedia, in particular, has

[3]http://www.j-archive.com/
[4]http://trec.nist.gov/data/qamain.html
[5]http://www.wiktionary.org/

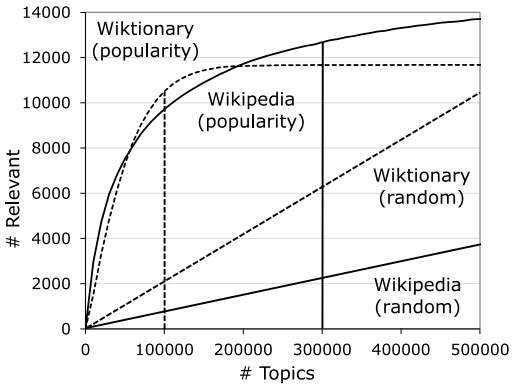

Figure 4: Relevance of Wikipedia and Wiktionary seeds for Jeopardy! when ranked by popularity or randomly.

proven to be a valuable resource for a variety of natural language processing tasks [9, 26] and has been used successfully in QA [2, 21]. The sources differ in two significant ways that affect SE: (1) Wiktionary entries are on average much shorter than Wikipedia articles (780 vs. 3,600 characters), and (2) Wiktionary entries are often about common terms. The shorter Wiktionary seeds render some topicality features less effective. Web queries for common terms yield more noise, which we alleviated by adding the keyword "define" to all queries and dropping search results that did not contain the topic in their title or URL.

Our cleansed copies of Wikipedia and Wiktionary contain about 2.1 million and 600,000 documents, respectively. To reduce computational costs and to avoid adding noise, we focused on expanding seeds that we deemed most relevant for Jeopardy! and TREC. Questions in both QA tasks are often about topics of common interest, such as famous people or well-known events, and thus SE should be prioritized to cover the most *popular* seeds first. For Wikipedia, in which rich hyperlink information is available, we defined the popularity of an article as the number of references from other articles in Wikipedia. Since Wiktionary contains only sparse hyperlink information, we estimated the popularity of a Wiktionary entry based on its frequency in a large collection of English documents across a variety of topics and genre. We sorted the seeds by popularity in descending order and plotted the number of seeds versus the number of those seeds that are *relevant* for the Jeopardy! task. A seed was considered relevant if its title was the answer to one of 32,000 Jeopardy! questions selected randomly from episodes that are not part of our test sets. This is an approximation of relevance based on the potential impact of a seed if only document titles are retrieved. The relevance curves for both sources, shown in Figure 4, have decreasing slopes, indicating that popularity-based rankings outperform random rankings of the seeds, which are illustrated by straight lines. We also performed this analysis for all factoid questions in TREC 8–15 and obtained very similar curves.

Based on these relevance curves, we chose to expand the top 300,000 Wikipedia articles and the top 100,000 Wiktionary entries. The size of the expanded pseudo-documents was restricted to at most 10 times the length of the seeds, and nuggets with relevance scores below 0.1 were dropped. We analyzed a sample of the pseudo-documents and chose parameter values that resulted in reasonable cutoff points. That is, with these cutoff values the final pseudo-documents contained a large number of relevant nuggets but little noise. Later we also performed QA experiments using different parameter settings and found that the precise thresholds have very little impact on QA performance. In general, more lenient thresholds yield slightly better results at the expense of larger expanded sources and thus longer QA runtime.

Table 5: Sizes of seed and expanded corpora.

Source	# Docs	Size
Wikipedia	2,114,707	7.0 GB
Expanded Wikipedia	100,000	4.8 GB
	200,000	7.6 GB
	300,000	9.8 GB
Wiktionary	565,335	420 MB
Expanded Wiktionary	100,000	590 MB

The text nuggets were scored using the model with features of adjacent instances, trained on all 15 topics. Overall, we processed about 40 million web pages, totaling about 2 TB of web data and 4 billion text nuggets. The expansion of each seed took around 15–45s and was dominated by the retrieval of web pages from different hosts. These numbers illustrate the necessity of a highly efficient and robust implementation. The sizes of the seed corpora and expanded sources are given in Table 5. The SE algorithm condensed the web data by two orders of magnitude, yielding a corpus that can be indexed and searched on a single node.

5.3 QA Search Experiments

We evaluated the impact of SE on search performance using Watson [17], a state-of-the-art QA system. As baselines, we used (1) Wikipedia, (2) Wiktionary and (3) a large collection of existing sources that we manually identified to be relevant for Jeopardy! and TREC based on an iterative error analysis. The collection (subsequently referred to as *All Sources*) comprises 23 GB of text, including encyclopedias such as Wikipedia and World Book, dictionaries such as Wiktionary, thesauri, newswire sources such as the AQUAINT corpus and a New York Times archive, literature and other sources of trivia knowledge [12]. We compare the Wikipedia and Wiktionary baselines to configurations that include the corresponding expanded sources. The collection of all sources is compared to a corpus to which only expanded versions of Wikipedia and Wiktionary were added.

The sources were indexed and searched with both Indri and Lucene, and the results were pooled. Queries were generated with Watson's retrieval component [11]. Two search strategies were evaluated: (1) we fetched a total of 20 *passages* and aligned them with sentence boundaries, and (2) we retrieved 50 documents in total and used their *titles* as results. The title searches target questions asking for an entity that matches a given description, such as the Jeopardy! question *From the Latin for "evening", it's a service of Evening Worship*[6] (answer: *vesper*) or the TREC question *What is the fear of lightning called?* (answer: *astraphobia*). Here source expansion helps by generating more complete descriptions of such terms that are more likely to match the question. Search results were judged as relevant if any token sequence matched the answer key. Performance was

[6]Jeopardy! questions are given in the form of statements.

Table 6: Search recall on Wikipedia, Wiktionary and all sources for Jeopardy! and TREC. For each setup, we show the percentage gain and the number of questions gained/lost. All improvements are significant with $p < .001$ based on a one-sided sign test.

Sources	Regular Jeopardy!			Final Jeopardy!			TREC 8–12		
	Passages	Titles	Total	Passages	Titles	Total	Passages	Titles	Total
Wikipedia	74.54%	65.19%	81.33%	52.54%	44.92%	63.32%	72.30%	49.42%	76.74%
Expansion	80.05%	71.86%	86.23%	59.39%	55.84%	72.21%	79.50%	52.32%	82.17%
% Gain	*+7.4%*	*+10.2%*	*+6.0%*	*+13.0%*	*+24.3%*	*+14.0%*	*+10.0%*	*+5.9%*	*+7.1%*
# Gain/Loss	*+280/-87*	*+293/-59*	*+211/-39*	*+77/-23*	*+104/-18*	*+88/-18*	*+203/-49*	*+150/-88*	*+157/-41*
Wiktionary	21.84%	18.93%	30.39%	8.12%	8.12%	13.32%	23.12%	15.96%	29.15%
Expansion	42.47%	32.47%	51.20%	19.67%	18.27%	27.79%	47.92%	26.02%	52.46%
% Gain	*+94.5%*	*+71.5%*	*68.5%*	*+142.2%*	*+125.0%*	*+108.6%*	*+107.3%*	*+63.0%*	*+80.0%*
# Gain/Loss	*+856/-132*	*+558/-83*	*+852/-122*	*+110/-19*	*+93/-13*	*+138/-24*	*+596/-66*	*+269/-54*	*+556/-58*
All Sources	78.48%	71.07%	**85.55%**	57.11%	52.54%	**69.54%**	75.76%	51.90%	**79.64%**
Expansion	82.38%	76.54%	**89.17%**	62.94%	61.42%	**75.51%**	80.30%	54.33%	**83.29%**
% Gain	*+5.0%*	*+7.7%*	***+4.2%***	*+10.2%*	*+16.9%*	***+8.6%***	*+6.0%*	*+4.7%*	***+4.6%***
# Gain/Loss	*+255/-118*	*+248/-56*	***+171/-44***	*+82/-36*	*+85/-15*	***+73/-26***	*+158/-61*	*+128/-76*	***+119/-41***

measured in search recall, the percentage of questions with relevant results. Search recall is an important metric in QA as it constitutes an upper bound on end-to-end accuracy.

Table 6 shows the impact of source expansion on search recall when retrieving only titles or passages and when combining these search results. We also indicate for each setup the number of questions gained and lost and the percentage gain. The performance on regular Jeopardy! is higher than on Final Jeopardy! since the regular questions are generally easier to process and ask about less obscure facts. The TREC performance numbers are lower than the regular Jeopardy! results because the answer keys were incomplete and because most of our development effort focused on Jeopardy! questions. On all datasets, SE yields consistent and significant ($p < .001$) improvements over the baselines, independently of the search strategy used. Even if a seed corpus with reasonable coverage for a QA task exists, such as Wikipedia for Jeopardy! and TREC, SE improves performance substantially. If a corpus with lower coverage is expanded, such as Wiktionary, very large gains can be realized. Compared to the strongest baseline, SE improves total search recall by 4.2% on regular Jeopardy! questions, 8.6% on Final Jeopardy! and 4.6% on TREC questions. The improvements are significant, even though of all the sources used in the baseline only Wikipedia and Wiktionary (one third of the collection) were expanded. It can also be seen that relatively few questions are hurt (e.g. 171 regular Jeopardy! questions are gained but only 44 out of 3,508 are lost). This distinguishes source expansion from most query expansion techniques: SE adds noise to the sources in addition to relevant content, but that is less likely to cause a search to fail than irrelevant keywords added to the query.

The recall curves in Figure 5 illustrate that SE improves search recall independently of the hit list length on both Jeopardy! and TREC questions. For these experiments we used only Indri and performed only one title search and one passage search to have a single hit list for each strategy. The gains in passage search recall are larger on Jeopardy! than on TREC questions because there is more headroom in Jeopardy! search performance. Passage recall on the TREC dataset approaches 90% even though the answer keys are often incomplete. Title searches are less effective for TREC since the questions targeted by this strategy are relatively infrequent in this dataset. The results also show that when adding the expanded sources, fewer titles or passages yield the same search recall as longer hit lists without source expansion. Decreasing the hit list lengths can be worthwhile since this reduces the number of candidate answers and improves the efficiency and effectiveness of answer selection.

Table 7: Search recall when expanding increasing numbers of Wikipedia seed articles for Jeopardy! and TREC.

	Regular J!	Final J!	TREC 8–12
Wikipedia	81.33%	63.32%	76.74%
Top 100,000	85.46%	71.32%	**80.91%**
Top 200,000	86.03%	72.08%	**80.91%**
Top 300,000	**86.09%**	**72.21%**	80.67%

Table 7 shows the impact of SE when expanding varying numbers of Wikipedia seeds. The top 100,000 seeds are responsible for most of the performance gain, which confirms that popularity-based seed selection is effective. On TREC, performance even degrades (though not significantly) if more than 200,000 seeds are expanded.

5.4 End-to-End QA Experiments

The Watson system was also used to evaluate the impact of source expansion on end-to-end QA accuracy, the percentage of questions answered correctly. Watson extracts candidate answers from the passage and title search results and scores them using a statistical model. During this answer scoring phase, additional supporting passages are retrieved for each of the top candidate answers, using a query comprising key terms from the question and the candidate itself. The supporting passages come from the same collection used in the initial searches, including the expanded sources. They are used to assess whether a candidate matches the information need expressed in the question and play a crucial role in answer scoring [17]. Thus Watson not only uses the expanded sources in the initial searches for candidate answers but also leverages them to score the candidates.

We again use the collection of manually acquired sources as a baseline and compare it to a configuration including expanded versions of Wikipedia and Wiktionary only. Watson

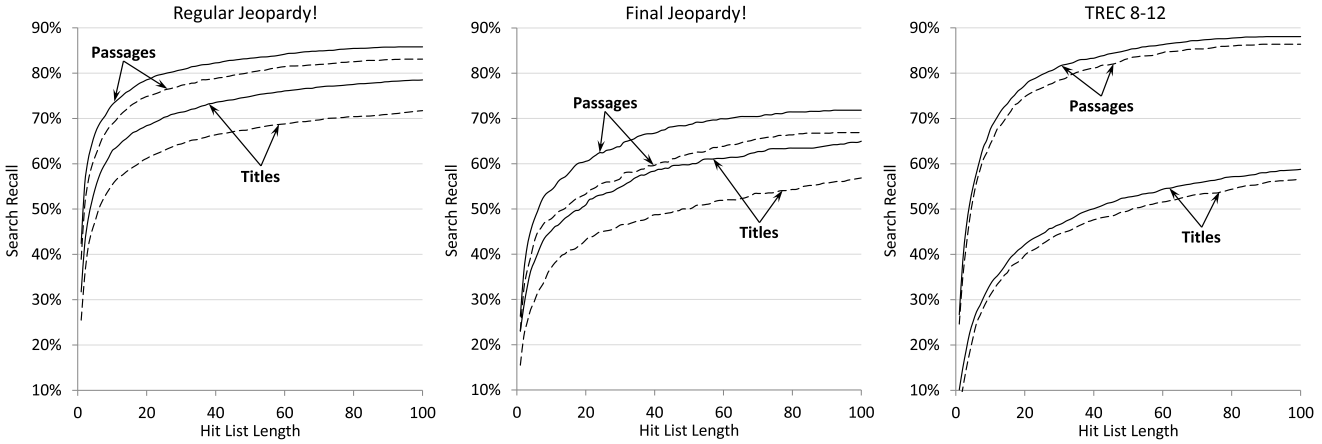

Figure 5: Search recall as a function of the hit list length for Jeopardy! and TREC. Dashed lines: all sources without expansion; solid lines: all sources with expansion of Wikipedia and Wiktionary.

Table 8: QA accuracies for Jeopardy! and TREC. For each dataset, we show the percentage gain and the number of questions gained/lost. All improvements are significant with $p < .01$ based on a one-sided sign test.

	Regular J!	Final J!	TREC 11
All Sources	66.08%	45.43%	59.46%
Expansion	71.12%	51.27%	63.96%
% Gain	*+7.6%*	*+12.9%*	*+7.6%*
# Gain/Loss	*+286/-109*	*+82/-36*	*+44/-24*

scores and ranks candidate answers using supervised models trained on question-answer pairs with relevance judgments. For the Jeopardy! task, we trained these models on separate datasets and used the same test sets as in previous experiments. For TREC, we had to reserve most data for training, leaving TREC 11 (444 questions with known answers) as an independent test set.

Table 8 shows that source expansion improves QA accuracy by 7.6% on regular Jeopardy! questions, 12.9% on Final Jeopardy! and 7.6% on TREC 11. The improvements in QA performance are significant with $p < .01$, even though only a fraction of the sources in the baseline were expanded. It is also worth noting that the baseline includes the AQUAINT corpus, which was used as the reference source in TREC 11 and thus contains the answers to all TREC questions. The improvements in accuracy exceed the gains in search recall on all datasets, which supports our claim that SE also improves answer selection (failure type 3 in Section 1).

6. CONCLUSIONS AND FUTURE WORK

We proposed a statistical approach for source expansion and evaluated its impact on large datasets of Jeopardy! and TREC questions, using different seed corpora and applying different search strategies. The proposed method yields significant gains in search performance on all datasets, improving search recall by 4.2–8.6%. These improvements were achieved *without* retrieving more text at QA runtime. Recall can always be increased through additional or longer search results, but this often comes at the expense of adding noise and hurting answer scoring efficiency and effectiveness. SE also significantly improves the overall performance of Watson, one of the most effective QA systems to date, increasing accuracy by 7.6–12.9% on Jeopardy! and TREC datasets.

It may seem that search performance always improves if more source content is added, but we found that this is not the case. For instance, just adding large web crawls is ineffective since the vast majority of web pages do not contain useful information [13]. We have shown that it is important to select relevant seed content for source expansion (Figure 4) and that performance can degrade if too many seeds are expanded (Table 7). Even if the most relevant seeds are chosen, an effective model is needed to accurately estimate the relevance of related text and avoid adding noise.

It is usually possible to make large improvements early in the development of a QA system, but as the system becomes more effective at its task the gains invariably get smaller. Often the impact of new algorithms depends on the order in which they are added because different approaches address similar issues. Yet we found that the impact of SE did not diminish as Watson was improved over time because advances in other components also helped better utilize the additional source content. For instance, more correct answers could be found in the expanded sources because of improved question analysis, search and answer extraction, and advances in answer scoring helped cope with the added noise. Because the gains from SE are largely orthogonal to other improvements, it is likely that this method will also help other QA systems and will be useful for other applications beyond QA. Since the approach is fully automated, it can be applied to new tasks with low effort.

We have begun working on sequential models for relevance estimation that predict transitions between relevant and irrelevant text using lexical coherence features derived from text segmentation algorithms [20, 5]. Preliminary results indicate that these models can outperform the logistic regression models discussed in this paper. We are also extending SE to seed corpora in which there is no one-to-one correspondence between documents and topics, such as newswire sources or web pages, by automatically detecting relevant topics and building pseudo-documents about them. Finally, SE can be applied to QA in different languages, and to other information retrieval and extraction tasks, such as document retrieval and relation extraction.

7. ACKNOWLEDGMENTS

The authors would like to thank Jamie Callan and Jaime Carbonell for helpful discussions on document expansion and multi-document summarization.

8. REFERENCES

[1] A. Agresti. *Categorical Data Analysis*. Wiley, 2002.

[2] D. Ahn, V. Jijkoun, G. Mishne, K. Müller, M. de Rijke, and S. Schlobach. Using Wikipedia at the TREC QA track. In *Proceedings of the Thirteenth Text REtrieval Conference*, 2004.

[3] G. Attardi, A. Cisternino, F. Formica, M. Simi, and A. Tommasi. PiQASso: Pisa question answering system. In *Proceedings of the Tenth Text REtrieval Conference*, 2001.

[4] N. Balasubramanian and S. Cucerzan. Automatic generation of topic pages using query-based aspect models. In *Proceedings of CIKM*, 2009.

[5] D. Beeferman, A. Berger, and J. Lafferty. Statistical models for text segmentation. *Machine Learning*, pages 177–210, 1999.

[6] M. Bilotti, P. Ogilvie, J. Callan, and E. Nyberg. Structured retrieval for question answering. In *Proceedings of SIGIR*, 2007.

[7] S. Blair-Goldensohn, K. McKeown, and A. Schlaikjer. Answering definitional questions: A hybrid approach. *New Directions In Question Answering*, 2004.

[8] S. Bradshaw. Reference directed indexing: Redeeming relevance for subject search in citation indexes. In *Proceedings of the 7th European Conference on Research and Advanced Technology for Digital Libraries*, 2003.

[9] R. Bunescu, E. Gabrilovich, and R. Mihalcea. Wikipedia and artificial intelligence: An evolving synergy. *Papers from the 2008 AAAI Workshop*, 2008.

[10] J. Carbonell and J. Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of SIGIR*, 1998.

[11] J. Chu-Carroll and J. Fan. Leveraging Wikipedia characteristics for search and candidate generation in question answering. *AAAI Conference on Artificial Intelligence*, 2011.

[12] J. Chu-Carroll, J. Fan, N. Schlaefer, and W. Zadrozny. Textual resource acquisition and engineering. *Submitted to IBM Journal of Research and Development*, 2011.

[13] C. Clarke, G. Cormack, M. Laszlo, T. Lynam, and E. Terra. The impact of corpus size on question answering performance. In *Proceedings of SIGIR*, 2002.

[14] C. Clarke, G. Cormack, and T. Lynam. Exploiting redundancy in question answering. In *Proceedings of SIGIR*, 2001.

[15] N. Craswell, D. Hawking, and S. Robertson. Effective site finding using link anchor information. In *Proceedings of SIGIR*, 2001.

[16] S. Dumais, M. Banko, E. Brill, J. Lin, and A. Ng. Web question answering: Is more always better? In *Proceedings of SIGIR*, 2002.

[17] D. Ferrucci, E. Brown, J. Chu-Carroll, J. Fan, D. Gondek, A. A. Kalyanpur, A. Lally, J. W. Murdock, E. Nyberg, J. Prager, N. Schlaefer, and C. Welty. Building Watson: An overview of the DeepQA project. *AI Magazine*, 31(3):59–79, 2010.

[18] J. Goldstein, V. Mittal, J. Carbonell, and M. Kantrowitz. Multi-document summarization by sentence extraction. In *Proceedings of the ANLP/NAACL Workshop on Automatic Summarization*, 2000.

[19] S. Harabagiu, D. Moldovan, M. Paşca, R. Mihalcea, M. Surdeanu, R. Bunescu, R. Gîrju, V. Rus, and P. Morărescu. The role of lexico-semantic feedback in open-domain textual question-answering. In *Proceedings of the 39th ACL Conference*, 2001.

[20] M. Hearst. TextTiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1):33–64, 1997.

[21] M. Kaisser. The QuALiM question answering demo: Supplementing answers with paragraphs drawn from Wikipedia. In *Proceedings of the ACL-08 HLT Demo Session*, 2008.

[22] B. Katz, J. Lin, D. Loreto, W. Hildebrandt, M. Bilotti, S. Felshin, A. Fernandes, G. Marton, and F. Mora. Integrating web-based and corpus-based techniques for question answering. In *Proceedings of the Twelfth Text REtrieval Conference*, 2003.

[23] G.-A. Levow and D. W. Oard. Signal boosting for translingual topic tracking: Document expansion and n-best translation. In *Topic Detection and Tracking: Event-based Information Organization, Chapter 9*, pages 175–195. Kluwer Academic Publishers, 2002.

[24] Y.-C. Li and H. M. Meng. Document expansion using a side collection for monolingual and cross-language spoken document retrieval. In *ISCA Workshop on Multilingual Spoken Document Retrieval*, 2003.

[25] M. Marchiori. The quest for correct information on the Web: Hyper search engines. *Computer Networks and ISDN Systems*, 29(8-13):1225–1235, 1997.

[26] O. Medelyan, C. Legg, D. Milne, and I. Witten. Mining meaning from Wikipedia. *International Journal of Human-Computer Studies*, 67(9):716–754, 2009.

[27] A. Nenkova, L. Vanderwende, and K. McKeown. A compositional context sensitive multi-document summarizer: Exploring the factors that influence summarization. In *Proceedings of SIGIR*, 2006.

[28] L. Pizzato, D. Mollá, and C. Paris. Pseudo relevance feedback using named entities for question answering. In *Proceedings of the Australasian Language Technology Workshop (ALTW)*, 2006.

[29] A. Singhal and F. Pereira. Document expansion for speech retrieval. In *Proceedings of SIGIR*, 1999.

[30] J. Tiedemann. Integrating linguistic knowledge in passage retrieval for question answering. In *Proceedings of HLT/EMNLP*, 2005.

[31] E. Voorhees. Overview of the TREC 2003 question answering track. In *Proceedings of the Twelfth Text REtrieval Conference*, 2003.

[32] R. Weischedel, J. Xu, and A. Licuanan. A hybrid approach to answering biographical questions. *New Directions In Question Answering*, 2004.