Statistical Source Expansion for Question Answering

Nico Schlaefer

CMU-LTI-11-019

Language Technologies Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Eric Nyberg (Chair)
Jamie Callan
Jaime Carbonell
Jennifer Chu-Carroll (IBM T.J. Watson Research Center)

Submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Language and Information Technologies

This research was supported by IBM Ph.D. Fellowships in the 2009–10 and 2010–11 academic years, and by IBM Open Collaboration Agreement #W0652159.

Abstract

A source expansion algorithm automatically extends a given text corpus with related information from large, unstructured sources. While the expanded corpus is not intended for human consumption, it can be leveraged in question answering (QA) and other information retrieval or extraction tasks to find more relevant knowledge and to gather additional evidence for evaluating hypotheses. In this thesis, we propose a novel algorithm that expands a collection of seed documents by (1) retrieving related content from the Web or other large external sources, (2) extracting self-contained text nuggets from the related content, (3) estimating the relevance of the text nuggets with regard to the topics of the seed documents using a statistical model, and (4) compiling new pseudo-documents from nuggets that are relevant and complement existing information.

In an intrinsic evaluation on a dataset comprising 1,500 hand-labeled web pages, the most effective statistical relevance model ranked text nuggets by relevance with 81% MAP, compared to 43% when relying on rankings generated by a web search engine, and 75% when using a multi-document summarization algorithm. These differences are statistically significant and result in noticeable gains in search performance in a task-based evaluation on QA datasets. The statistical models use a comprehensive set of features to predict the topicality and quality of text nuggets based on topic models built from seed content, search engine rankings and surface characteristics of the retrieved text. Linear models that evaluate text nuggets individually are compared to a sequential model that estimates their relevance given the surrounding nuggets. The sequential model leverages features derived from text segmentation algorithms to dynamically predict transitions between relevant and irrelevant passages. It slightly outperforms the best linear model while using fewer parameters and requiring less training time. In addition, we demonstrate that active learning reduces the amount of labeled data required to fit a relevance model by two orders of magnitude with little loss in ranking performance. This facilitates the adaptation of the source expansion algorithm to new knowledge domains and applications.

Applied to the QA task, the proposed method yields consistent and statistically significant performance gains across different datasets, seed corpora and retrieval strategies. We evaluated the impact of source expansion on search performance and end-to-end accuracy using Watson and the OpenEphyra QA system, and datasets comprising over 6,500 questions from the Jeopardy! quiz show and TREC evaluations. By expanding various seed corpora with web search results, we were able to improve the QA accuracy of Watson from 66% to 71% on regular Jeopardy! questions, from 45% to 51% on Final Jeopardy! questions and from 59% to 64% on TREC factoid questions. We also show that the source expansion approach can be adapted to extract relevant content from locally stored sources without requiring a search engine, and that this method yields similar performance gains. When combined with the approach that uses web search results, Watson's accuracy further increases to 72% on regular Jeopardy! data, 54% on Final Jeopardy! and 67% on TREC questions.

Acknowledgements

First of all, I would like to thank my advisor Eric Nyberg for his support and guidance throughout my studies at Carnegie Mellon. From the beginning, Eric placed great confidence in me, allowing me to explore new research directions and develop my own research objectives. I also deeply appreciate his generosity and his readiness to share his experience and to give helpful advice whenever needed. Without his support, this work would not have been possible. I am also very grateful to Jennifer Chu-Carroll, who has been my mentor at IBM Research for over three years. I have been fortunate to work with one of the most experienced researchers in the field of question answering, and was able to learn a lot from her about her area of expertise and about conducting rigorous scientific research. Furthermore, I would like to thank Jamie Callan for his thoughtful comments and honest reflections on my work. Jamie made many helpful suggestions that resulted in additional experiments and analyses and ultimately improved the quality of this work considerably. I am also thankful to Jaime Carbonell for sharing his vast experience and knowledge of machine learning and language technologies with me. Despite his extremely busy schedule, Jaime took the time to study my work carefully and provide valuable feedback and guidance.

Much of my thesis research would not have been possible without the help of the DeepQA group at IBM. In addition to Jennifer's continuous support, I had the pleasure of working closely with James Fan and Wlodek Zadrozny during my summer internships at IBM Research. James and Wlodek contributed many interesting ideas and new insights that led to significant improvements of our method. In addition, I feel indebted to many other IBMers who helped build Watson, which served as a testbed for most of the experiments in this thesis. I am particularly grateful to Eric Brown, Pablo Duboue, Edward Epstein, David Ferrucci, David Gondek, Adam Lally, Michael McCord, J. William Murdock, John Prager, Marshall Schor, and Dafna Sheinwald. Furthermore, I greatly appreciate the help I received from Karen Ingraffea and Matthew Mulholland with various data annotation tasks. Karen and Matt have been instrumental in creating a dataset for statistical relevance modeling, and they helped evaluate Watson's answers to thousands of questions to obtain unbiased estimates of the performance impact of our approach.

I am also extremely grateful to my parents, Stefan and Maria Schläfer, for their constant support and trust. They always encouraged me to make my own choices and follow my own interests, while supporting me in every way they could and helping me obtain the best education possible. Finally, I would like to thank my wife, Sachiko Miyahara, for her encouragement and understanding during these busy years. Sachiko was always ready to provide advice and support, and never complained when I was busy working towards yet another deadline. I am very fortunate to have met her in Pittsburgh.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In Section 1.1 we motivate the need for source expansion techniques using automatic question answering as a sample application. Our statistical approach to source expansion is introduced in Section 1.2 and the contributions of our work are summarized in Section 1.3. Finally, we outline the structure of this thesis in Section 1.4.

1.1 Motivation

Source expansion has an important application in question answering (QA), the task of retrieving accurate answers to natural language questions from knowledge sources. In QA, it is common to locally store and index text 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 it may be difficult to find the answers even if they are supported by the sources. For example, the terminology used to describe the answer in the text corpus may be different from the keywords in the question, relevant information may be distributed over multiple documents, or additional inference may be required to derive the answer. To improve the coverage of local sources and to facilitate the extraction and validation of answers, the sources can be expanded automatically with additional related information and paraphrases of existing information found in large, external text corpora such as the Web. This thesis presents a fully implemented statistical approach that performs this *source expansion* task effectively, and compares different strategies for obtaining and selecting relevant information to be added to a given seed corpus. We further demonstrate that source expansion consistently and significantly improves QA performance on large datasets drawn from the Jeopardy! quiz show and TREC evaluations.

Source expansion (SE) is an effective strategy for improving QA performance because it addresses the most common types of failures that occur in state-of-the-art QA systems:

1. *Source failures*, i.e. the sources do not contain the information sought by a question, or they mention it in a different context and do not justify it as the

correct answer to the given question. SE can address these failures by adding more relevant content to the sources.

2. *Search and candidate extraction failures*, i.e. the sources contain and justify the answer, but the QA system is unable to retrieve relevant text or to extract the answer from the search results. Often, these failures are due to insufficient keyword overlap between questions and passages that contain the answers. SE can help by adding paraphrases of information that is already present in the knowledge sources.

3. *Answer selection failures*, i.e. the correct answer is extracted as a candidate but it is not selected as the most probable answer. This frequently happens if there is insufficient supporting evidence in the sources or if the answer was found in a low-ranking search result. Again, SE can address these failures by adding reformulations and increasing semantic redundancy in the sources, yielding more relevant search results for candidate answer extraction and scoring.

In contrast, when performing query expansion or using pseudo-relevance feedback (PRF) in the search phase of a QA system, we can (at the most) address failures of types 2 and 3 since we do not increase the coverage of the sources. In practice, these techniques can add noise to the queries and typically have high variance [Collins-Thompson and Callan, 2007]. In question answering, query expansion and PRF often do not improve the search results much and in some experiments even hurt system performance [Pizzato et al., 2006]. Furthermore, even if these methods improve average performance in a batch evaluation on a large dataset, they sometimes perform poorly on seemingly easy questions, which may be unacceptable when interacting with real users as this affects the user's trust in the reliability of the system. In addition, these techniques lead to more complex queries that may be more expensive to evaluate at QA runtime, and PRF even requires an additional search step for each question. For these reasons, query expansion and PRF are often only applied as fallback solutions if an initial query yields low recall [Harabagiu et al., 2001, Attardi et al., 2001].

Source expansion applied to the information sources of a question answering system in a preprocessing step can also be preferable over live web searches at QA runtime. While web search engines typically must be used as black boxes, locally stored sources can be indexed with open-source IR systems such as Indri[1] or Lucene[2], which allow full control over the retrieval model and search results. These IR systems offer rich query languages, supporting term weights, proximity constraints, field restrictions and approximate matching. 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 [Tiedemann, 2005, Bilotti et al., 2007]. Furthermore, in applications where speed and availability matter, a live web search and subsequent retrieval of web pages may not be a viable alternative. Live web searches may also be infeasible if the knowledge sources contain

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

confidential information (e.g. a company intranet) or restricted-domain knowledge (e.g. instructions for products), or if a self-contained system is required (e.g. in areas with intermittent Internet access, or for the IBM Jeopardy! challenge). Local sources further guarantee reproducible evaluation results, whereas the Web and the algorithms and indices used by web search engines change constantly. Finally, local sources can be consolidated e.g. by merging related content into a single document and by removing noise.

Our source expansion method has many potential applications beyond QA. For instance, the related content gathered through SE could be added to the representations of the documents in a seed corpus to improve retrieval accuracy of a traditional document retrieval system. The expanded documents could also be leveraged by a relation extraction algorithm to validate instances of relations found in the seed corpus and to identify additional instances. Furthermore, a system for recognizing textual entailment (RTE) could use paraphrases obtained through source expansion as additional evidence to determine whether a given statement is entailed by the information in a text corpus. In this thesis, however, we focus on an intrinsic evaluation of different strategies for selecting relevant information for source expansion (Chapter 5) and demonstrate the effectiveness of our approach in task-based evaluations on QA data (Chapter 6). We show that SE significantly and consistently improves the performance of Watson [Ferrucci et al., 2010], 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 sets of questions drawn from the Jeopardy! TV show and TREC evaluations. Similarly, our method improves search recall of the OpenEphyra[3] open-source QA system by 4.0%–13.7% on large TREC datasets. Suggestions for other applications of source expansion are made when we discuss areas for future research at the end of the thesis (Chapter 9).

1.2 Approach

The input to our source expansion (SE) algorithm is a topic-oriented seed corpus, i.e. a document collection in which each document contains information about a distinct topic. Each of these seed documents can be expanded with additional information about the same topic and reformulations of information that is already covered in the following four-stage pipeline (further described in Chapter 4):

1. Retrieve content that is related to the topic of the seed document from a large external source (e.g. the Web or a locally stored text corpus).

2. Extract self-contained nuggets of text from the retrieved content (e.g. paragraphs or sentences).

3. Estimate the relevance of the text nuggets with regard to the topic of the seed document using a statistical model.

[3] `http://sourceforge.net/projects/openephyra/`

4. Compile a new pseudo-document from the most relevant nuggets, excluding lexically redundant text.

This procedure can be repeated for each document in the seed corpus, yielding an expanded corpus with increased coverage and semantic redundancy. The expanded corpus can be leveraged as an additional source by an information retrieval or extraction system. In question answering, the corpus can be indexed and used to retrieve content that is related to a given question for answer extraction, or to gather additional supporting evidence for candidate answers in an answer selection phase. Note that the expansion is performed only once in a preprocessing step and that the generated sources can then be used to answer a wide range of questions that are related to the topics in the seed corpus. Our SE method does not require specific knowledge of the questions that will be posed to a QA system at runtime.

We found that it is not always beneficial or computationally feasible to expand all documents in a seed corpus. Thus we developed strategies for selecting topics that are most likely to be useful for a given QA task based on different measures of popularity (Section 6.2). For example, when expanding the encyclopedia Wikipedia, we focused on articles that are frequently referenced by other Wikipedia articles and are thus likely to contain information of common interest. In experiments with Wiktionary, an online dictionary, we expanded entries about terms that are frequently mentioned in a large English text corpus.

Crucial to the effectiveness of source expansion is the statistical relevance estimation approach (step 3 in the pipeline). We developed a variety of relevance features, ranging from surface characteristics of the extracted text nuggets (e.g. the positions of nuggets in the source documents and whether they contain many known words or trigrams) to search-related features (e.g. the search ranks of the source documents) to topicality features that take the content of the seed documents into account (e.g. using language models, *tf-idf* term weights and a multi-document summarization algorithm). Statistical models trained on a large dataset of manually annotated web pages can be used to effectively score text nuggets and separate relevant from irrelevant text (Chapter 5). However, we will show that similar results can be achieved with substantially less annotation effort through active learning techniques (Section 8.1). Linear model fitting methods such as logistic regression are well-suited for this task, but we were able to attain slightly higher performance with a sequential model that estimates the relevance of text nuggets given the surrounding text in their source documents (Section 8.2).

In most experiments, we focused on expanding topic-oriented sources such as encyclopedias and dictionaries with related content from web search results. However, our approach can also leverage unstructured sources in which there exists no one-to-one correspondence between documents and topics that are suitable for SE, such as locally stored web crawls and newswire corpora. We will show that these document collections can be used to gather related content for source expansion without relying on a web search engine (Section 7.1). In addition, we will outline how an unstructured corpus can be transformed into a topic-oriented source by identifying topics that have high coverage in the corpus and building pseudo-documents from relevant information

about these topics. The pseudo-documents can then be used as a seed corpus and can be expanded with the four-stage SE pipeline for topic-oriented sources described above (Section 7.2).

1.3 Contributions

We propose a novel algorithm for expanding text corpora with relevant information from the Web and other large document collections using statistical models for relevance estimation. The source expansion approach comprises techniques for selecting useful seed topics based on popularity estimates, methods for acquiring related content from external resources, and a framework for selecting relevant text using statistically motivated features. Our implementation is sufficiently robust and efficient to support the expansion of text corpora comprising millions of documents. We also give examples of how this method can be adapted to different types of seed documents and information needs.

To support the annotation of high-quality training data for the relevance estimation task and to facilitate the adaptation of our method to new knowledge domains and applications, we developed an annotation interface and a set of guidelines that aid annotators in making consistent decisions. We applied this methodology to generate a dataset comprising 1,500 web pages and over 160,000 hand-labeled text nuggets with manual relevance judgments. Different baselines and statistical models are evaluated intrinsically on the dataset, and their effectiveness for ranking text by relevance is compared. The baselines include rankings generated by a web search engine and an effective multi-document summarization algorithm. Linear models fitted with logistic regression are used to score text nuggets based on a variety of relevance features, and a sequential model combines these features with a second set of transition features to dynamically predict boundaries between relevant and irrelevant text. The statistical models consistently outperform the baselines, and they yield statistically significant improvements in average precision. We also show that active learning can reduce the amount of labeled data needed to fit statistical models by two orders of magnitude with only a small loss in relevance estimation performance.

Source expansion is also applied to the question answering task, using Watson and the OpenEphyra QA system as testbeds. Its impact is evaluated on questions from the Jeopardy! TV show and factoid questions from TREC evaluations [Dang et al., 2007]. In these experiments, we expand about 500,000 seed documents from encyclopedias and a dictionary with related text from web search results. For each QA task, search performance and end-to-end results are evaluated on datasets of over 3,000 questions. Our approach yields consistent and statistically significant improvements over baselines without source expansion, improving QA accuracy by 7.6%–12.9%. We also demonstrate that seed documents can be expanded with related information extracted from a locally stored text corpus without requiring a search engine. The impact of this extraction-based method on QA performance is comparable to the source expansion approach that uses web searches. When combining the two methods, QA accuracy increases by 9.4%–19.8% on Jeopardy! and TREC datasets.

1.4 Outline

This thesis is organized as follows. In Chapter 2 we give an overview of established research areas and existing algorithms that are related to our work on statistical source expansion. Chapter 3 introduces a canonical QA pipeline and describes QA tasks, performance metrics and systems that served as the basis for a task-based evaluation of SE. In Chapter 4 we discuss our SE method in detail, give examples of expanded documents generated with our approach, and illustrate how they help improve QA performance. In Chapter 5 we present intrinsic evaluation results for our relevance models and baselines, and in Chapter 6 we apply statistical SE to the question answering task. Chapter 7 describes how the SE approach can be extended to leverage unstructured document collections as seed corpora or as sources of related content for SE. In Chapter 8 we discuss how active learning can reduce the need for labeled training data when fitting statistical models for relevance estimation, and we introduce sequential models that estimate the relevance of text nuggets in the context of surrounding nuggets. Finally, we summarize our findings and propose directions for future research in Chapter 9.

Chapter 2

Related Work

In this chapter we compare source expansion to related disciplines of natural language processing that influenced our work (Section 2.1). We further give an overview of previous approaches for constructing local sources that can be used by QA systems (Section 2.2) and related work on document expansion for information retrieval (Section 2.3). In Section 2.4 we describe a summarization algorithm that was adapted to estimate the relevance of text passages for source expansion. The chapter concludes with an overview of text segmentation approaches that motivated features used in a sequential model for relevance estimation (Section 2.5).

2.1 Relations to Established Research Areas

We identified ties between our statistical source expansion approach and various established NLP tasks, including multi-document summarization, definitional question answering, content-based information filtering, and topic detection and tracking.

The problem of extracting text nuggets that relate to a given topic from documents is perhaps most similar to multi-document summarization (e.g. Goldstein et al. [1999, 2000], Nenkova et al. [2006]), but it differs in the following important ways:

1. The selection of relevant text for source expansion is guided by the content of a seed document. For instance, we have designed features that evaluate the topicality of text nuggets using term weights and topic language models estimated from seed content. In multi-document summarization, on the other hand, seed documents are usually not available and instead the relevance of text passages is estimated based on much shorter query strings. In our experiments, topicality features that use the seed content are combined with other relevance features that are based on web search results and the surface forms of text nuggets in a statistical model. By leveraging the seeds, we were able to substantially improve relevance estimation performance. However, we will show that even if the seed documents are sparse or of low quality, or if no seed content is available at all, our approach for selecting relevant text nuggets still outperforms several strong baselines.

2. While we avoid lexical redundancy by removing text nuggets that are near duplicates of other nuggets or seed content, semantically redundant text that phrases the same information differently is desirable. For instance, QA systems benefit from semantic redundancy as it becomes more likely that a relevant passage in the sources closely matches the keywords in a question and can be retrieved in the search phase. Furthermore, multiple supporting text passages can be leveraged to more accurately estimate the confidence in a candidate answer. Similarly, semantic redundancy can improve the recall and precision in other information retrieval and extraction tasks. Multi-document summarization, on the other hand, aims at compiling succinct summaries consisting of text passages that add novel information. Thus, relevant text that rephrases important information but adds little new content may not be included in the summary.

3. The pseudo-documents generated through SE are not intended for human consumption. Thus, we are not concerned with generating coherent paragraphs of text that appear natural to a human reader. This allows us to remove irrelevant content from the pseudo-documents more aggressively. For instance, we can drop text nuggets if their confidence estimates fall below a threshold even if the remaining text appears out of context to a human. As the threshold can be adjusted to include more or less relevant text, we can fully control the size of the expanded sources.

In this thesis, we compare our statistical relevance estimation approach to a summarization algorithm. We chose the maximal marginal relevance (MMR) algorithm introduced by Carbonell and Goldstein [1998] because of its effectiveness and relative simplicity. The MMR algorithm generates a summary by iteratively selecting text passages that are relevant to a query and add novel information that is not yet covered by previously selected passages. Additional details about the algorithm are provided in Section 2.4. MMR is effective for identifying a diverse sample of relevant text, but the statistical method is more suitable for selecting relevant and semantically redundant information for source expansion. Furthermore, we found that the performance of the statistical model can be improved by incorporating a feature that is based on a variation of the MMR algorithm.

Recent research on definitional QA has led to various algorithms for compiling relevant texts on topics such as people, organizations or events. For instance, Blair-Goldensohn et al. [2004] describe a hybrid approach to answering definitional questions that combines knowledge-based and statistical methods. Weischedel et al. [2004] answer biographical questions of the form *Who is X?* by automatically extracting linguistic constructs such as appositives and propositions from sentences mentioning a person. More recently, various strategies for definitional QA have been proposed in the proceedings of the Text REtrieval Conference [Dang et al., 2007]. Kaisser et al. [2006] present a simple yet effective web reinforcement approach which scores candidate sentences based on how frequently their keywords occur in web search results. Qiu et al. [2007] extract sentences from the sources and rank them with a statistical model that combines syntactic features, retrieval scores and language models.

However, while all these systems generate answers to individual questions from existing sources at QA runtime, source expansion is concerned with the construction of new source material in a preprocessing step. The new sources can then be used by a QA system to answer a wide range of questions from different domains with little computational overhead at runtime. An efficient definitional QA algorithm may also be applied offline to compile summaries about a large set of pre-defined topics, but this would require an approach for identifying topics that are relevant to a given QA domain without knowing the questions yet. Our statistical SE approach relies on a seed corpus to select the topics, assuming that the corpus is at least partially relevant and mentions topics that are central to the QA domain. In addition, as discussed previously, the SE approach benefits from existing seed content when determining the relevance of candidate text, whereas a definitional QA algorithm only relies on a short topic string given in the question.

2.2 Construction of Local Sources

QA systems often use the Web as a large, redundant information source [Clarke et al., 2001, Dumais et al., 2002], but it has also been noted that there are situations where a local search is preferable [Clarke et al., 2002, Katz et al., 2003]. Clarke et al. [2002] analyzed the impact of a locally indexed web crawl on QA performance using a TREC dataset. The crawler was seeded with the home pages of educational institutions and retrieved linked web pages in breadth-first order. The authors found that over 50 GB of web content were required to outperform the 3 GB reference corpus used in TREC, and that performance actually degraded if the crawl exceeded about 500 GB. Our proposed method 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 [2009] propose an algorithm for generating documents comprising useful information about given topics from web data. The usefulness of sentences extracted from web pages is determined by using aspect models built from query logs of the Bing[1] search engine. These aspect models consist of words that frequently co-occur with a given topic or related topics in the query logs. The approach is used to compile biographical information about people from Wikipedia's "Living people" category, but it appears to be applicable to other types of topics as well. The authors note that the topic pages generated with their approach are often preferable over the search result pages created by Bing. A key difference to the proposed SE approach lies in the generation of topic models for selecting useful content. Instead of relying on query logs, we leverage the content of existing seed corpora to model topicality. While comprehensive query logs may be hard to come by, particularly when starting out in a new domain, there already exist seed corpora for many knowledge domains. For instance, Wikipedia articles can be used as seeds when developing a QA system for trivia questions, a medical encyclopedia could serve as a starting point for a QA system that answers questions about diseases and cures,

[1] `http://www.bing.com/`

and a legal dictionary can be used for answering questions about the law. Furthermore, while the query-based aspect modeling approach is evaluated on a relatively small dataset of 300 topics, we were able to apply our source expansion approach efficiently to more than 400,000 topics and provide intrinsic evaluation results as well as extrinsic results on multiple QA datasets comprising over 3,000 questions each.

For the Jeopardy! QA challenge, we initially followed a manual source acquisition approach in which the information sources used by the Watson QA system were augmented with resources that were expected to fill in gaps in the coverage of the Jeopardy! domain identified in an error analysis. However, we found that statistical SE substantially outperforms this intuitive approach to building source material, and it also requires less manual effort. Furthermore, when we manually augmented the knowledge sources with text corpora that improved QA performance on a particular question set, we often did not observe the same performance gains when switching to an unseen test collection. This was particularly true when adding small or specialized sources that only addressed a few questions in a development set. Statistical SE, on the other hand, does not depend on any particular dataset and is therefore less prone to overfitting. It does, however, rely on a relevant seed corpus that at least touches on the most important topics. Also note that the manual and automatic approaches to growing source material are not mutually exclusive. Given a collection of manually acquired sources, statistical SE can be performed using these sources as seeds to gather additional related content.

2.3 Document Expansion

In this section we will discuss a variety of techniques for expanding text documents to facilitate their retrieval given queries that do not exactly match the terminology used in those documents. For instance, documents can automatically be augmented with terms and phrases extracted from related documents in the same collection or an auxiliary corpus. Some methods also re-weight existing content words based on their frequencies in related text, or substitute terms with similar or more general concepts that appear in the query. When indexing a collection of web pages, the anchor text associated with hyperlinks provides high-level descriptions of the linked documents and can be added to their index representations. Similarly, in a collection of scientific publications, terms that appear in the vicinity of citations can be associated with the referenced articles. Conversely, the content of linked or cited documents can be propagated to the documents that contain the references.

Research on document expansion dates back at least to the early 1980s, when O'Connor [1982] leveraged citation text in a corpus of chemistry articles to improve search recall, and the field has since evolved into a diverse and well-established area of research. It has been shown that document expansion can improve retrieval performance on various tasks, including spoken document retrieval (e.g. Singhal et al. [1998], Johnson et al. [1999]), cross-lingual information retrieval (e.g. Darwish and Oard [2002], Li and Meng [2003]), topic tracking (e.g. Levow and Oard [2002]), and web search (e.g. Brin and Page [1998], Craswell et al. [2001]). In the following, we

present a sample of this work, distinguishing between techniques for collections of independent documents that do not explicitly reference one another (Section 2.3.1) and methods that rely on citations or hyperlinks between the documents (Section 2.3.2). We further discuss how the proposed statistical source expansion approach extends this related work by providing applications such as QA with additional relevant content that cannot be found in the original corpus (Section 2.3.3).

2.3.1 Expansion of Independent Documents

Singhal et al. developed a document expansion approach that improves the performance of a speech retrieval system [Singhal et al., 1998, Singhal and Pereira, 1999]. Their algorithm is evaluated on a dataset from the Spoken Document Retrieval (SDR) task at TREC 7 [Garofolo et al., 1998]. Collections of text documents are generated by transcribing spoken news stories using automatic speech recognition (ASR), and the documents are indexed and searched with a traditional vector space retrieval system. However, since ASR is error-prone and frequently fails to recognize important content words, document retrieval over automatic speech transcriptions is generally less effective than the retrieval of written or manually transcribed documents. To compensate for recognition errors, the transcriptions are automatically expanded with terms extracted from other documents. Using the original transcription as a query, related documents are retrieved from a corpus of newspaper articles that covers a similar time period. Relevant terms are then extracted and added to the transcription and the weights of existing terms are adjusted using Rocchio's relevance feedback algorithm [Rocchio, 1971]. To avoid adding noise to the transcription, the expansion can optionally be restricted to terms that also appear in the word lattice generated by the ASR system and that are thus acoustically similar to the words that were actually recognized. This conservative extension of the algorithm mainly compensates for speech recognition failures, rather than addressing vocabulary mismatches between queries and spoken documents.

The authors demonstrate that document retrieval from the expanded transcriptions consistently outperforms retrieval from the original transcriptions without expansion. This result holds even when expanding gold standard transcriptions generated by humans. Furthermore, if a reasonably accurate ASR system is available, document expansion performed on top of automatic transcriptions can be more effective than retrieval from human-generated transcriptions that were not expanded. Thus the document expansion approach does not only help in the presence of speech recognition errors, but may also improve traditional document retrieval from written text collections. If the queries are expanded using pseudo-relevance feedback, document expansion is still beneficial but the gains in retrieval performance tend to be smaller. This document expansion approach relies on the availability of a similar text corpus to retrieve related documents and is less effective if an unrelated document collection or the speech transcriptions are used instead. Singhal et al. further observe that if a word already appears frequently in the original transcribed document, it is often also selected by the document expansion algorithm and may be overweighted as a result. By preventing excessive increases in word frequencies, the algorithm could

be further improved in the TREC 8 SDR evaluation [Singhal et al., 1999].

Darwish and Oard [2002] performed document expansion in the English-Arabic cross-lingual information retrieval (CLIR) evaluation at TREC 11 [Oard and Gey, 2002]. In the CLIR task, an Arabic newswire corpus was searched for documents that relate to topics in English. As relevant documents often only contain a fraction of the terms in the topic descriptions, each document was enriched with related terms from other documents in the same collection by applying the following procedure:

1. Extract terms with high *tf-idf* scores from the document, and use them as a query to retrieve related documents.

2. Merge the retrieved documents and extract high *tf-idf* terms from the combined document.

3. Add the extracted terms to the original document.

Unfortunately, the contribution of this expansion to overall system performance was not evaluated, and it is unclear whether the algorithm helped.

Li and Meng [2003] show that document expansion can improve search performance in cross-lingual spoken document retrieval. In their evaluation, English newswire documents are used as queries to retrieve relevant broadcast news stories in Mandarin. The queries are translated to Chinese and the broadcast news are transcribed with an ASR system. Li and Meng deploy a vector space retrieval model, using syllable bigrams as the basic indexing unit. The speech transcriptions are used as queries to retrieve related documents from a Chinese newswire corpus, and are expanded with terms extracted from those documents. This expansion approach yields consistent gains in mean average precision[2] on a test set of 195 queries.

Similarly, Levow and Oard [2002] used document expansion for cross-lingual topic detection and tracking (TDT) in the TDT-3 evaluation [Graff et al., 1999]. Given English newswire articles as training data, the task was to retrieve Mandarin news stories that cover the same topics. The collection of Mandarin documents used in TDT-3 comprises both newswire documents and automatically transcribed broadcast news. While Li and Meng [2003] translate English queries to Chinese and perform document expansion in the target language, Levow and Oard translate the Chinese news and expand the translations using an English newswire corpus. The document expansion improves the effectiveness of the topic tracking system on both newswire articles and broadcast news, with a slightly larger gain on the (presumably noisier) speech transcriptions.

Tao et al. [2006] apply document expansion to information retrieval using language models. Their retrieval engine ranks the documents in a collection by the KL divergence between language models estimated from the query and the documents. To mitigate the problem of data sparsity when estimating document language models, each model is interpolated with the language models of similar documents in the collection. Cosine similarities are used to identify the most similar documents and to weight those documents. This approach consistently improves retrieval accuracy

[2]See Section 3.3 for a definition of mean average precision (MAP).

on different TREC datasets. The largest gains are realized when expanding short documents that would otherwise yield sparse language models.

Billerbeck and Zobel [2005] compare the efficiency and effectiveness of document expansion and query expansion using pseudo-relevance feedback. The experiments are based on topic sets and newswire corpora from TREC evaluations. Two alternative methods for document expansion are evaluated:

- Each document in the collection is used as a query to retrieve similar documents, related terms are extracted from the retrieved documents, and the original document is augmented with these terms. This approach is similar to the expansion algorithms used by Singhal et al. [1998, 1999], Darwish and Oard [2002], Li and Meng [2003] and Levow and Oard [2002], but the implementation details vary.

- Each term in the vocabulary is expanded with related terms using pseudo-relevance feedback, the expanded query is used to rank the documents in the collection, and the original term is added to the highest ranking documents. This technique is intended to augment documents that are related to a given term with that term if it is not already present in the documents.

The authors observe that query expansion with pseudo-relevance feedback can be computationally intensive at retrieval time, whereas both document expansion methods are performed at indexing time with little computational overhead during retrieval. However, they also found that their document expansion approaches often do not improve retrieval effectiveness significantly, whereas query expansion yields more consistent gains in their experiments. These results clearly do not confirm the previously discussed observations based on SDR, CLIR and TDT datasets, where document expansion yielded consistent and substantial gains. It is possible that the expansion techniques evaluated by Billerbeck and Zobel are less effective, or the impact of document expansion may depend on the dataset.

Zhang et al. [2002] observe that queries often describe a topic in general terms, whereas relevant documents use more specific terminology. For instance, a query may contain the concept *computer* while a document in the corpus specifically mentions subtypes such as *server*, *PC* or *laptop*. In this scenario, query expansion by adding more specific concepts may be ineffective if the query terms have too many irrelevant hyponyms. As a more viable alternative, the authors propose a document expansion algorithm that replaces terms in the corpus with more general concepts. At retrieval time, nouns in the documents are substituted with hypernyms or their synonyms in WordNet [Fellbaum, 1998] if these terms occur in the query. In contrast to previously discussed algorithms, this method cannot be applied at indexing time since it depends on the query, and it focuses on replacing existing terms in the documents instead of adding new terms or manipulating weights. In the TREC 2002 Novelty track [Harman, 2002], the approach outperformed a query expansion algorithm that augmented query terms with their hyponyms. However, this technique may be computationally intensive, and it is unclear whether the hypernym expansion was performed for the entire collection or only for documents that ranked high in an initial search.

2.3.2 Link and Citation Analysis

Another common form of document expansion leverages links and citations in a text corpus to enrich documents with related text from documents that reference them or are referenced by them. For example, when searching the Web or a local collection of HTML documents (e.g. the intranet of a company) the anchor texts associated with hyperlinks can be added to the representations of the target pages to enhance them with high-level descriptions of their content. The use of anchor text for web page retrieval dates back to the WWW Worm, one of the earliest web search engines developed in 1993 [McBryan, 1994]. During indexing, a web crawler recorded for each URL the anchor texts of links pointing to it, as well as the titles and URLs of the linking pages. This meta-data could then be searched independently or in conjunction with other indexed information, and the users could navigate to the linking web pages. Brin and Page [1998] adopted the idea of leveraging anchor texts for document retrieval in the Google search engine. They noted that often anchor texts are more descriptive of the linked pages than the text on those pages and thus associated the anchor texts with the destination pages. This approach also enabled Google to retrieve files that do not contain plain text, such as images and programs, and pages that were not downloaded by their web crawler.

Craswell et al. [2001] demonstrate that anchor text can be used effectively to retrieve documents from both a general TREC web collection and a crawl of a university website. While previous TREC evaluations primarily focused on subject search (e.g. find documents about *text categorization*), Craswell et al. concentrate on the task of finding sites given their names (e.g. find the home page of *Carnegie Mellon*). Anchor texts can be useful for site finding since they often contain the names of the linked web pages. In contrast, the target pages themselves may not contain their names in textual form, or this information can be outweighed by other content. The authors found retrieval based on anchor text to be significantly more effective than retrieval using the actual content of the documents. In their experiments, a relevant search result could be retrieved in first place more than twice as often from an index in which each document is represented by the anchor text of incoming links than from a conventional document index.

Instead of using anchor text that is associated with incoming links, Marchiori [1997] leverages outgoing links to improve document rankings generated by web search engines. The author noted that linking a web page is similar to directly incorporating the target content into the web page since a visitor can access this content by following the link. However, rather than adding the content of linked pages to the source document, the relevance of each page with regard to a given query is evaluated independently and the relevance scores are combined, giving less weight to linked pages. This method was applied as a post-processing step to refine the relevance scores and document rankings generated by several web search engines. In a user study, the rankings that were adjusted based on outgoing links were preferred over the original rankings from the search engines.

Similarly to hyperlinks, citations in a corpus of scientific publications can be leveraged for document expansion. Here the text surrounding a citation of a related

article can be regarded as a summary of the most important information in this article, or information that is deemed important in the context of the citing article. However, while the recognition of anchor text is straightforward, it can be difficult to accurately select relevant text that should be associated with citations. O'Connor [1982] used citing statements in a corpus of chemical journal papers to improve search recall. A rule-based approach was developed for selecting sentences in the vicinity of a citation and other text that can be associated with the referenced article, such as the citing article's title and the heading that precedes the citation. The execution of these rules was simulated manually because the corpus was not available in electronic form. As a baseline, O'Connor used an index that combined manually assigned index terms with words that were automatically extracted from a digital database of abstracts. By expanding this index with words from citing statements, recall of a Boolean search was improved from 50% to 70%. Note, however, that the expansion also increased the number of articles matching Boolean queries, and thus it may have hurt precision.

Bradshaw [2003] observes that it is hard to automatically extract relevant terms from a single citing statement, but if a publication is referenced multiple times, the references can be compared and terms that are used repeatedly can be given more weight. The author proposes a retrieval model that assigns high scores to documents that are frequently cited using many of the terms in a query. This model is aimed at retrieving publications that are both relevant to a query and influential in their field of research. The approach is compared to a vector space model that computes cosine similarities between the query and the content of documents. In an evaluation using subject queries against a collection of scientific publications from CiteSeer[3], the citation-based method retrieved on average 1.66 more relevant articles among the first 10 results than the vector space approach. Bradshaw concludes that retrieval using citation text is more robust because a query term that repeatedly appears in references to a document is usually central to its topic, whereas the document itself may frequently contain terms that are unrelated to the topic.

2.3.3 Comparison to Source Expansion

All of these document expansion methods have in common that they improve retrieval accuracy on an existing text collection. We have seen consistent gains in document ranking performance (in recent experiments with vector space models and language models) and search recall (in early studies using a Boolean retrieval model). This was accomplished by augmenting document representations with related terms and phrases found in the same collection or an auxiliary corpus, and by re-weighting or replacing existing terms. As a result, the retrieval system can more accurately evaluate the relevance of documents to a given query. Document expansion can be particularly helpful if the query terms do not match the terminology used in relevant documents. This is often the case for sparse queries containing few useful keywords, and short documents that have low coverage and redundancy. However, these techniques are less suitable for enhancing the search results with additional content that can be

[3]Recently replaced by CiteSeer^X: `http://citeseerx.ist.psu.edu/`

leveraged by another application. For instance, individual terms and phrases that are added to the documents are of limited use to a question answering system for answer extraction or as supporting evidence for answer validation since their context is not preserved. Also, many of the discussed algorithms expand documents with related terms found in the same collection, rather than adding content from external sources that is not already included in the collection. Similarly to query expansion and pseudo-relevance feedback, document expansion techniques are unlikely to help if the relevant information sought by a QA system is missing in the corpus (failure type 1 in Section 1.1).

Like document expansion, our statistical source expansion approach also addresses vocabulary mismatches between queries and relevant content in a document collection. By augmenting sources with reformulations of existing information, source expansion increases the likelihood that at least some relevant text closely resembles the query and can be retrieved. Similarly to the document expansion techniques for independent documents in Section 2.3.1, our method does not rely on explicit references between related documents. Source expansion is different from the above techniques in that it also adds large amounts of useful content to a seed corpus. In our experiments, local corpora are expanded several-fold with related text passages from much larger auxiliary collections, such as the Web. As a result, source expansion not only improves document rankings but it also increases the amount of relevant content in the search results. Semantic redundancy in the retrieved text facilitates the extraction of relevant information and the validation of this information by a QA system. In addition to increasing redundancy, the proposed method adds new information that was not present in the original sources and could therefore not be found even with perfect retrieval. In other words, if the answer to a question is not in the original corpus, source expansion may add it to the sources, but document expansion techniques are unlikely to help. Finally, source expansion is not only helpful for applications that use an information retrieval system to search the text collections, but the added content can also be leveraged directly for information extraction without performing a search. For instance, a relation extraction algorithm could benefit from the increased coverage and semantic redundancy of an expanded text corpus.

While the previously described document expansion algorithms add words or phrases to existing documents, our method processes source corpora as a whole and generates new sources that can be indexed and searched independently. The proposed approach encompasses techniques for identifying topics in a text corpus and selecting relevant topics for which new pseudo-documents are generated. It can even be applied to unstructured sources that are not divided into documents about distinct topics. For these reasons, we refer to this new method as *source expansion* as opposed to *document expansion* throughout the thesis.

2.4 Maximal Marginal Relevance

The maximal marginal relevance (MMR) algorithm is an intuitive and effective approach for single- and multi-document summarization that was introduced in its most

basic form by Carbonell and Goldstein [1998] and later extended with additional features by Goldstein et al. [2000]. Given a query string that describes an information need and a set of candidate text passages, the algorithm iteratively selects passages that are relevant to the query while avoiding redundant information. We adopt a variation of the MMR algorithm as a baseline for estimating the relevance of text nuggets in the relevance ranking step of the source expansion system (Sections 5.2–5.4), and we also derive a feature for our statistical relevance models from this method.

A summary is generated by incrementally selecting text passages that maximize a measure of *relevant novelty*. Carbonell and Goldstein applied the MMR algorithm to sentences, but one can also use other units of text. We will show that for source expansion it is generally more effective to rank longer, paragraph-length text passages. In each step, the algorithm selects a passage that is similar to the query and thus likely to be *relevant*, but at the same time dissimilar from previously selected passages and therefore likely to contain *novel* information. More formally, let Q be the query, R the set of candidate passages, and S the subset of those passages that were already selected in previous iterations. Then a passage P_{sel} is selected such that

$$P_{sel} = \arg \max_{P \in R \setminus S} \left[\lambda \, CosSim(P, Q) - (1 - \lambda) \max_{P' \in S} CosSim(P, P') \right].$$

The objective function is a weighted average of a relevance term and a novelty term, and the parameter $\lambda \in [0, 1]$ controls the tradeoff between these components. If $\lambda = 1$ then the most relevant passages are selected regardless of their redundancy, and if $\lambda = 0$ then the algorithm selects a maximally diverse sample of text. In source expansion we focus on finding relevant information, and we include reformulations of previously selected text since they may facilitate answer extraction and validation in question answering. In addition, we need not worry about lexically redundant content or even exact duplicates since such text is filtered out in the final merging phase of the SE system. Thus values of λ close to 1 can be expected to be most effective for our application.

Goldstein et al. [2000] compute term weight vectors for the query and candidate text passages, and measure the cosine similarity between these vectors. For instance, if V is the vocabulary of all terms, then the cosine similarity between a candidate passage P and the query Q is defined as

$$CosSim(P, Q) = \frac{\sum_{t \in V} w_P(t) \; w_Q(t)}{\sqrt{\sum_{t \in V} w_P(t)^2 \; \sum_{t \in V} w_Q(t)^2}},$$

where $w_P(t)$ and $w_Q(t)$ are the weights of the term t in passage P and in query Q, respectively. Goldstein et al. remove stopwords and stem the remaining tokens, and they use `lnn` term weights (i.e. **l**ogarithmic term frequency, **n**o document frequency, **n**o normalization). For example, $w_P(t) = 1 + \log(f_P(t))$ where $f_P(t)$ is the frequency count of term t in passage P. For additional information about `lnn` term weighting and other weighting schemes that are commonly used in information retrieval, we

refer the reader to Manning et al. [2008]. In our implementation of MMR, individual characters and tokens that appear on a list of about 250 function words are removed, the Porter stemmer [Porter, 1980] is applied to the remaining tokens, and the same term weighting scheme is used.

The choice of query Q is crucial to the performance of the MMR algorithm. Carbonell and Goldstein achieve strong results using topic descriptions that consist of about 100–150 tokens as queries. These descriptions were compiled manually and are of high quality, and they tend to repeat important information. When applying MMR to select text that is related to the topic of a seed document for source expansion, we found that the approach is most effective if the entire seed is used as a query. If instead only the seed title is used, or if the seeds are artificially degraded by removing text or adding noise, performance degrades substantially (cf. Sections 5.3 and 5.4). Thus the algorithm appears to work best if the queries are long, and redundancy in the queries may help reinforce important information and give more weight to key terms in the cosine similarity calculations.

The MMR algorithm terminates once a given number of passages has been selected, or when the compression ratio reaches a threshold. The compression ratio is defined as $|S| \,/\, |R|$, where $|\cdot|$ could e.g. be the total character length or the number of text passages in a set. However, it can be difficult to choose a threshold when evaluating the algorithm because the ideal summary length is subjective and should be determined based on the application and user preferences. Depending on the cutoff point, precision and recall of the generated summaries can differ widely. In addition, it may not be effective to use a fixed threshold because different summary lengths may be optimal depending on the topic and the amount of relevant content in the set of candidate passages. For these reasons, we do not set a threshold in our experiments, but instead we generate a complete ranking of all candidate passages. Ranking performance is evaluated using mean average precision (MAP) as a single aggregate measure that takes into account the quality of the ranking at different cutoff points (see Section 3.3 for a definition of MAP).

Both MMR and source expansion are extraction-based methods, i.e. they compile documents from existing text passages (e.g. sentences or paragraphs) instead of generating new text. The two approaches also have in common that they use domain-independent techniques and focus on statistical processing rather than natural language understanding. The MMR algorithm and the relevance estimation component in our SE system also use similar amounts of related content and are designed to achieve similar compression ratios. Goldstein et al. [2000] report that they condensed 200 newswire articles down to 0.3% of their original length measured in characters. We typically apply our relevance models to about 100 web pages per seed topic and, depending on the length of the seed, reduce their size to about 0.1–0.6%.

The efficiency of the MMR algorithm depends on the choice of the parameter λ. If only the relevance component is used (i.e. $\lambda = 1$) then the computation time is linear in the number of text nuggets since each text nugget is compared to the query only once. In practice, it takes less than 1 second per seed document to rank the text nuggets by relevance in this special case. However, if the novelty of text nuggets is taken into account (i.e. $\lambda < 1$) then nuggets must be selected incrementally based on

their similarity to the previously selected nuggets. Consequently, each nugget must be compared to all other nuggets, and the runtime is quadratic in the number of nuggets. We found that it takes about 1–6 minutes to rank the text nuggets for a single seed topic in this general case. As a comparison, all other features used in our statistical relevance models can be computed in less than 2 seconds. The experiments were run on a server with a 3 GHz Xeon CPU and 32 GB RAM.

When using a parameter value of $\lambda = 1$, the MMR algorithm simply ranks the text passages by their cosine similarity to a given query. This special case provides a strong baseline for relevance estimation that outperforms other single-method ranking strategies by a wide margin. For instance, this approach ranks a large set of manually labeled text nuggets with 75% MAP, compared to 43% MAP when using rankings generated by a web search engine. Furthermore, while this method is outperformed by a statistical model that combines different relevance estimation strategies, it can be integrated into the model to further increase its ranking performance. By adding cosine similarities as an additional feature, we were able to improve MAP by 2.5–3.5 percentage points compared to a model that does not use this feature.

2.5 Sequential Models for Text Segmentation

In the nugget formation and scoring phases of the SE system, relevant text can be identified and separated from irrelevant text using sequential models (see Section 8.2). This problem is related to text segmentation and layout analysis, and some of the features used in our models are based on ideas drawn from these research areas.

Particularly relevant for our work was the TextTiling algorithm [Hearst, 1997] and a statistical approach for text segmentation [Beeferman et al., 1999]. The TextTiling algorithm counts word frequencies in adjacent text windows and uses the cosine similarity between the word frequency vectors as a measure of lexical coherence. The coherence of a document can be visualized by plotting the cosine similarity at different positions in the text, and transitions between topics are predicted at "valleys", i.e. areas of low similarity surrounded by more coherent text. Beeferman et al. [1999] combine language modeling features and trigger words in an exponential model to predict boundaries between news articles. They estimate the likelihood of a sentence under long-range and short-range language models trained on previous sentences, and assume a document boundary to be more likely at sentence boundaries where the likelihood ratio is small, i.e. the context captured by the long-range model does not help in predicting the next sentence. Binary trigger word features increase or decrease the probability of a document boundary if a given word occurs nearby.

We implemented several variations of the TextTiling algorithm to predict transitions between relevant and irrelevant text based on lexical coherence. The statistical models proposed by Beeferman et al. were less applicable to nugget formation and scoring since they were tailored to the segmentation of whole newswire documents. The language modeling features were designed to separate longer units of text, and we found it hard to identify trigger words that indicate transitions between topics in our much noisier and heterogeneous web data. However, we attempted to incorporate

the idea of trigger words in a more generic way using transition language models.

In Section 8.2 we describe a sequential relevance model that leverages these transition features, and we compare it to linear models that select relevant text passages without taking the lexical coherence of their source documents into account.

Chapter 3

Fundamentals

In this chapter we introduce a canonical QA pipeline and discuss how additional knowledge sources generated through statistical source expansion can be integrated, and how they can improve system performance (Section 3.1). We also give an overview of QA tasks on which we evaluate the impact of source expansion (Section 3.2) and performance metrics adopted in our experiments (Section 3.3). Finally, we describe existing QA systems that are used in our task-based evaluation (Section 3.4).[1]

3.1 Pipeline for Question Answering

Although a variety of architectures have been adopted by question answering systems, the vast majority of systems are based on a core pipeline of components for question analysis, query generation, search, candidate answer generation and answer scoring [Hickl et al., 2007, Shen et al., 2007, Schlaefer et al., 2007, Ferrucci et al., 2010]. The *question analysis* component derives syntactic and semantic information from the question, using techniques such as answer type classification, syntactic and semantic parsing, and named entity recognition. In the *query generation* stage, this information is transformed into a set of search queries, often with some degree of query expansion, which are passed to the *search* component to retrieve relevant content from a collection of *knowledge sources*. The search results are processed by the *candidate generation* component, which extracts or generates candidate answers of the desired granularity (e.g. factoid answers or definitional phrases). The *answer scoring* component estimates confidence scores for the candidate answers, ranks the candidates by confidence and often merges similar candidates. At this stage, the knowledge sources can be reused to retrieve supporting evidence for the most promising candidates. The final output is a list of answers ranked by confidence estimates.

Figure 3.1 illustrates this canonical architecture and shows how a question might be processed by a typical implementation. The input to the QA system is the question *Which computer scientist invented the smiley?* given in textual form. In this example, the question analysis component determines that the question seeks an answer of the

[1] The discussions of question answering pipelines, the TREC QA task and performance metrics were adapted from Schlaefer and Chu-Carroll [2012].

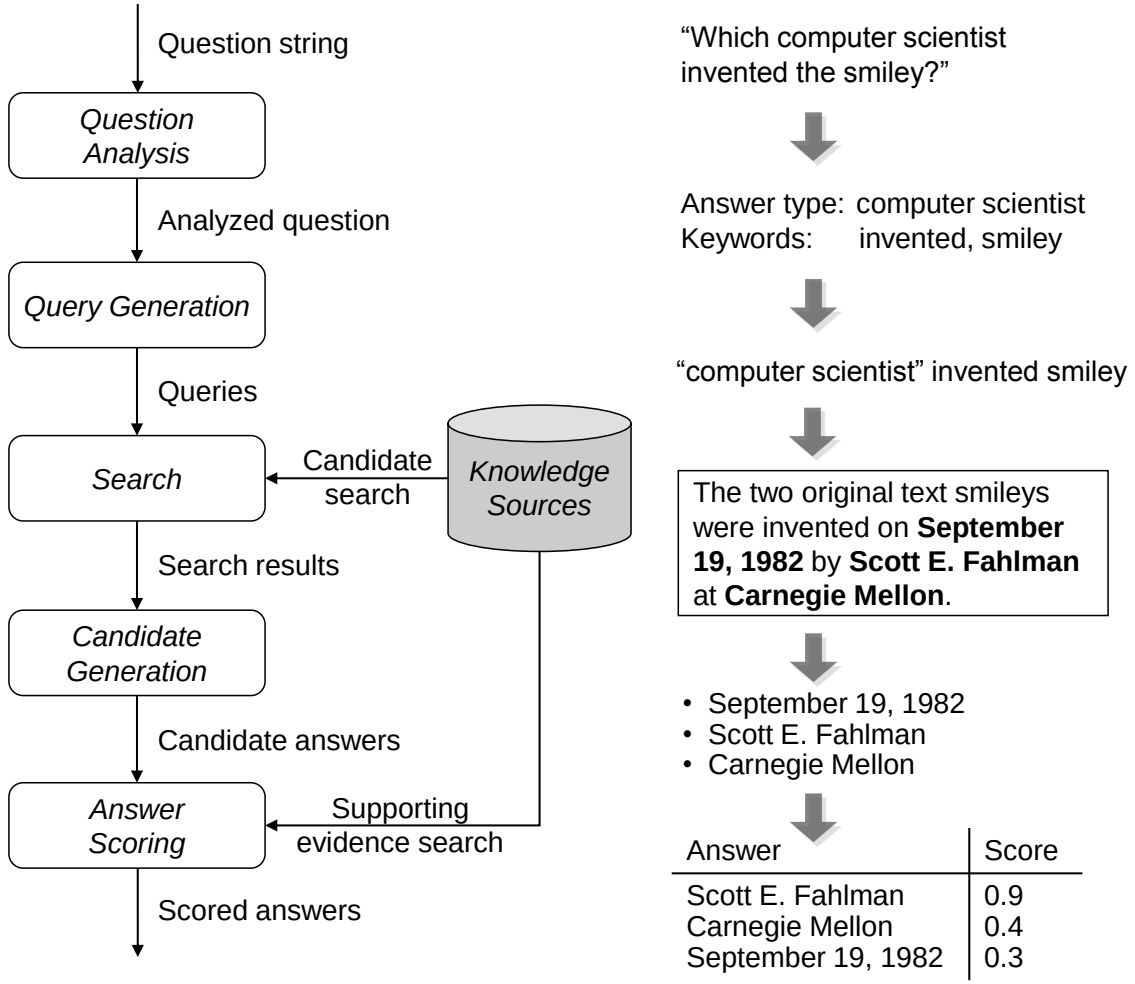


Figure 3.1: Canonical question answering architecture (left) and sample question processed in the pipeline (right). This illustration was adopted from Schlaefer and Chu-Carroll [2012].

type *computer scientist*, and it extracts the additional keywords *invented* and *smiley*. The query generation component constructs a search engine query from the answer type and the additional keywords. Given this query, the search component retrieves passages from a text corpus (say, the Web), such as the one shown in the example. In the candidate generation stage, entities are extracted as candidate answers. Finally, an answer scoring component estimates a confidence score for each candidate, using features such as the retrieval rank, the number of occurrences of a candidate in the search results, and whether it matches the predicted answer type. The highest scoring candidate, *Scott E. Fahlman*, is returned as the most probable answer. For additional details on QA pipelines and algorithms that are commonly used in each of the pipeline stages, we refer the reader to Schlaefer and Chu-Carroll [2012].

When applying source expansion to the question answering task, the knowledge sources are augmented with additional information, which can be utilized both by the search component and by the answer scoring component in a QA pipeline. Thus,

SE has the potential of improving both search performance and answer scoring performance:

- The QA system may retrieve relevant content for additional questions from the expanded sources, which directly improves search recall.
- Even if for a given question relevant text was already found in the original sources, SE may help retrieve more relevant results at higher ranks. This facilitates the extraction of the correct answer from the search results. It may also improve answer scoring performance if the frequencies of candidate answers in the search results and their search ranks are used as features.
- During the answer scoring phase, some QA systems retrieve supporting evidence for individual candidate answers from the knowledge sources. For instance, a system may issue a query for each candidate consisting of question keywords and the candidate itself [Magnini et al., 2002, Ko et al., 2010, Ferrucci et al., 2010]. The search results are then compared to the question on a lexical, syntactic or semantic level, and the candidate is scored according to the degree of similarity. Expanded sources may yield additional supporting evidence and thus facilitate answer scoring.

3.2 Question Answering Tasks

We evaluated our statistical source expansion approach on two question answering tasks: (1) factoid questions from annual QA evaluations conducted by the Text REtrieval Conference (TREC), and (2) questions from the Jeopardy! TV game show. In this section, we give an overview of these tasks and discuss differences that affect the impact of SE.

3.2.1 Text REtrieval Conference (TREC)

The Text REtrieval Conference (TREC) conducted annual evaluations of English question answering systems in the years 1999–2007 (TREC 8–16) [Dang et al., 2007]. This evaluation forum has been one of the major drivers of English QA research, and the questions and corresponding answer keys produced in the evaluations have become standard test collections. Initially, TREC focused on factoid questions, but in later years list, definitional and relationship questions were added to the evaluations. While earlier test sets (TREC 8–12) consisted of independent and self-contained questions, in more recent evaluations (TREC 13–16) questions were grouped into series with a common topic and contained coreferences to the topic, preceding questions and answers to those questions. Table 3.1 illustrates common types of independent TREC questions, and Table 3.2 gives an example of a question series with coreferences. Note that in the question series, the noun phrases *the 1999 Game* and *the game* refer to the topic *1999 Baseball All-Star Game*, and the noun phrase *the ballpark* in *What is the seating capacity of the ballpark?* refers to the answer of the previous question.

Type	Sample Questions
Factoid	Who was the first American in space? *(TREC 8, Question 21)* Where is the Valley of the Kings? *(TREC 9, Question 249)*
List	Name 20 countries that produce coffee. *(TREC 10 list task, Question 1)*
Definitional	Who is Aaron Copland? *(TREC 12 main task, Question 1901)* What is a golden parachute? *(TREC 12 main task, Question 1905)*

Table 3.1: Independent factoid, list and definitional questions in TREC.

Type	Question
Factoid	In what city was the 1999 All-Star Game held?
Factoid	In what city was the 1999 Game originally scheduled?
List	List the official sponsors of the game.
Factoid	What is the name of the ballpark where the game was played?
Factoid	What is the seating capacity of the ballpark?
Factoid	What was the date of the 1999 All-Star Game?
Factoid	Who was the Most Valuable Player (MVP) of the game?

Table 3.2: Question series about the TREC topic *1999 Baseball All-Star Game* (TREC 15, Target 161).

In TREC evaluations, participating systems were required to retrieve answers from reference corpora including collections of newspaper articles and a large crawl of blog websites. The systems further had to support each answer with a document from these sources that contains the answer and justifies it. Thus, even though the systems were allowed to leverage additional sources such as the Web for candidate generation and answer scoring, the justification of the final answers had to come from the sources used in the evaluations. The answers submitted by the participants were judged manually by human assessors.

In this thesis, statistical source expansion is evaluated on factoid questions, leaving other question types such as list and definitional questions as promising directions for future work. For factoid questions, the correct answers found in the evaluations were later compiled into regular expressions, which we used as answer keys for automatic evaluations.[2] Note, however, that these answer keys are often incomplete since they only cover correct answers found by participants and assessors in the reference corpora used in the evaluations [Dang et al., 2007]. When evaluating end-to-end QA performance, we often extended the answer keys with additional correct answers found in

[2]The answer keys are available at `http://trec.nist.gov/data/qamain.html`.

Category	Jeopardy! Clue
Y TO K	Synonym for "tug" or "jerk" that was a nickname for a Northerner (*Answer:* Yank)
A GIRL'S BEST FRIEND	South Africa's diamond industry began in 1867 after a child found a diamond beside this "colorful" river (*Answer:* Orange)
ZOOLOGY	This behavior in gorillas can express exuberance or intimidate (*Answer:* Chest-beating)
"F" IN MATH	A number written as a quotient; examples are 2/3, 1/2 and 1/4 (*Answer:* Fraction)
TONY WINNERS	This redhead won four Tonys in the 50's (*Answer:* Gwen Verdon)
1946	Of 25%, 50%, or 75%, the percentage of veterans in the Sept. 1946 enrollment at Harvard (*Answer:* 75%)
FARAWAY PLACES	1 of the 3 countries bordering Sudan whose names start with an "E" (*Answers:* Ethiopia, Eritrea, Egypt)
ANAGRAMS THAT MAKE SENSE	He was a poet, actor & dramatist: I'LL MAKE A WISE PHRASE (*Answer:* William Shakespeare)
CROSSWORD CLUES "L"	A cradle song (7) (*Answer:* Lullaby)

Table 3.3: Examples of Jeopardy! categories and clues.

our experiments to more accurately estimate actual system performance. When we evaluated intermediate search results, however, it was not feasible to manually judge all retrieved passages and documents, and thus we did not further extend the answer keys. As a result, we systematically underestimate true search performance, but this does not affect the validity of our comparative evaluations.

3.2.2 Jeopardy!

The second question answering task on which we evaluated our source expansion approach is the Jeopardy! TV quiz show. The Jeopardy! task mainly consists of factoid questions whose answers can be found in text corpora such as encyclopedias or dictionaries, but also includes puzzle questions that require additional computation and inference. Jeopardy! questions (also called *clues*) are given in the form of statements rather than in interrogative form, and are associated with a category that may provide additional useful or even essential information. A game show is divided into two rounds of regular Jeopardy! questions followed by a single Final Jeopardy! question. In each round, the contestants are asked 30 questions from 6 different categories. Thus there are up to 61 questions in total (in some games not all questions are revealed). Final Jeopardy! questions are usually more difficult, and contestants are given more time to answer. Examples of Jeopardy! categories and questions are shown in Table 3.3.

Datasets consisting of past Jeopardy! questions and their answers were retrieved from J! Archive[3]. The answers were augmented manually with additional correct answers found by IBM's Watson QA system (see Section 3.4.2) to allow for more accurate automatic evaluations of system performance. Since the answer keys were updated regularly and since questions in Jeopardy! are usually designed to only have a single correct answer, the automatic evaluation results reported for this task closely reflect true system performance.

The Jeopardy! challenge differs from factoid questions in TREC in several important ways that affect the potential impact of the proposed source expansion approach [Ferrucci et al., 2009]:

- In the TREC evaluations, the participants were given a reference corpus that contained the answers to most of the questions. If the answer was not in this corpus, the QA systems were expected to detect this and return no answer. For the Jeopardy! challenge, there is no requirement to find the answers in a particular information source. Any text corpora that contain useful information can be used as sources of candidate answers, with the only restriction that the QA system must be self-contained and cannot consult external resources such as the Web during a Jeopardy! match. Therefore, in order to achieve competitive performance, it is essential to develop an approach for identifying and acquiring relevant source content ahead of time. This can be a manual source acquisition methodology or an automated process such as statistical source expansion.

- Jeopardy! questions usually test general knowledge, and therefore the answers can be found in or deduced from sources such as encyclopedias (e.g. Wikipedia) and dictionaries (e.g. Wiktionary). These sources are ideal for automatic source expansion since each document is about a known, distinct topic that is specified by an unambiguous title. On the other hand, TREC questions sometimes seek very specific information that can only be found in a given reference corpus. The sources used in TREC mainly consist of newspaper articles, whose topics are often not distinct and may be unknown.

- Compared to TREC questions, Jeopardy! clues are often more complex and much longer, containing non-vital information that may be intended as a hint for the contestants, or to entertain or educate the audience. As a consequence, queries generated from Jeopardy! clues generally contain more key terms and are more constrained than TREC queries. Thus they match fewer text passages and large, redundant sources may be required to achieve high search recall. At the same time, more constrained queries are usually less sensitive to noise that may be introduced through automatic source expansion in addition to relevant information. This makes the Jeopardy! task an ideal test case for the source expansion approach.

- Jeopardy! contestants can choose not to answer a question in the regular Jeopardy! rounds if they do not know the answer or are not sufficiently confident,

[3] `http://www.j-archive.com/`

and thus reliable confidence estimation for candidate answers is of vital importance. Source expansion can provide additional supporting evidence for answer scoring, which may improve the reliability of the confidence estimates and their consistency across questions. However, the contestants must always answer the Final Jeopardy! question, even if they have low confidence.

- The contestants in Jeopardy! must compete for the buzzer, i.e. the first player to press a button gets to answer a question. Only if that player is wrong or fails to answer within 5 seconds will the other contestants get their chance to answer. Thus, in order to compete with strong human players, a QA system must be capable of answering rapidly. Since the retrieval of relevant text for candidate answer extraction and supporting evidence for answer scoring are among the most expensive processing steps in a QA pipeline, and the efficiency of the searches is directly related to the size of the sources, it is essential to avoid adding noise during source expansion. Of course, noise should also be removed from the sources because it can affect the quality and relevance of the search results. This is the primary objective of using statistical models for relevance estimation (see Section 4.3).

3.3 Performance Metrics

When we evaluate the impact of SE on factoid QA performance in Chapter 6, we report *QA accuracy*, the percentage of questions answered correctly by a system. More formally, let n be the number of questions in a test set and c the number of questions answered correctly by the top-ranked candidate returned by the system. Then accuracy is defined as

$$Accuracy = \frac{c}{n}.$$

Accuracy is also a common metric for evaluating classifiers, where *classification accuracy* is defined as the percentage of instances that are assigned the correct label.

While QA accuracy is based on only the answers with the highest confidence, the *mean reciprocal rank* (MRR) metric also gives partial credit to correct answers at lower ranks. For each question q_i in a test set ($i = 1, ..., n$) let r_i be the rank of the first correct answer in the list of candidates generated by a QA system for that question, if one has been found. Then the MRR is computed as follows:

$$MRR = \frac{1}{n} \sum_{i=1}^{n} \begin{cases} \frac{1}{r_i} & \text{if a correct answer was found,} \\ 0 & \text{otherwise.} \end{cases}$$

The MRR is commonly based on only the top answers up to a fixed rank. For instance, MRR@10 only takes the 10 highest ranked answers for each question into account. This metric is useful for capturing the system's ability to extract the correct answer while rewarding systems that are able to rank that answer higher up in the answer list. Note that the MRR is not only useful for evaluating final answers but can also be

computed for intermediate search results retrieved by a QA system, such as passages or documents, if relevance judgments for these results are available.

The performance of a QA system on list questions is often measured in terms of *F-scores*. Let t_i be the total number of correct answers to the i-th list question in a test set, r_i the number of answers returned by a QA system for that question, and c_i the number of those answers that are correct. Further, let precision and recall on the i-th question be defined as follows:

$$Precision_i = \frac{c_i}{r_i} \quad \text{and} \quad Recall_i = \frac{c_i}{t_i}.$$

The F-score is a weighted harmonic mean of precision and recall:

$$F_i(\beta) = \frac{(\beta^2 + 1) \times Precision_i \times Recall_i}{\beta^2 \times Precision_i + Recall_i}.$$

In this formula the weight parameter β determines the relative importance of precision and recall. The larger β, the more weight is given to recall, that is, the more important it becomes to find all correct answers and the less important to avoid incorrect answers. If $\beta = 1$, precision and recall are equally important. The overall performance of a QA system on a set of list questions can now be defined as the arithmetic mean of the F-scores:

$$F(\beta) = \frac{1}{n} \sum_{i=1}^{n} F_i(\beta).$$

F-scores are also used for evaluating definition questions and other types of questions with complex answers. In TREC evaluations, the assessors compiled lists of information nuggets they considered to be vital or acceptable parts of an answer. Recall and precision were defined based on the coverage of those nuggets in the answer produced by a system and the overall length of the answer [Dang et al., 2007].

When comparing rankings of text nuggets generated with different relevance estimation strategies in Chapter 5, we compute precision and recall at different cutoff points. Given rankings of n different sets of text nuggets, let $c_i(r)$ be the number of relevant instances among the top r instances in the i-th ranking ($i = 1, ..., n$), and t_i the total number of relevant instances in that ranking. Then precision and recall at rank r are defined as follows:

$$Precision@r = \frac{1}{n} \sum_{i=1}^{n} \frac{c_i(r)}{r} \quad \text{and} \quad Recall@r = \frac{1}{n} \sum_{i=1}^{n} \frac{c_i(r)}{t_i}.$$

A *precision-recall curve* is obtained by plotting *Precision@r* versus *Recall@r* for different values of r. It illustrates the tradeoff between the two metrics, and can be used to determine the attainable precision/recall if a certain recall/precision is required. Recall is also a useful measure for evaluating the search results retrieved by a QA system (*search recall*) or the candidate answers extracted from the results (*candidate recall*). Search recall is defined as the percentage of questions for which relevant results were

found, and candidate recall is the percentage of questions for which correct answers were extracted. These measures indicate the maximum performance achievable by downstream components in a QA system and thus can be regarded as upper bounds on end-to-end QA accuracy.

Finally, we evaluate rankings of text nuggets in terms of *mean average precision* (MAP), defined as

$$MAP = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{t_i} \sum_{j=1}^{t_i} \frac{c_i(r_{i,j})}{r_{i,j}},$$

where $r_{i,j}$ is the rank of the j-th relevant text nugget in the i-th ranking. MAP is a popular means of summarizing the quality of rankings generated by information retrieval systems in a single performance metric.

3.4 Question Answering Systems

We give a brief overview of two existing question answering systems that we utilized to evaluate the impact of source expansion on QA performance. The Ephyra system (Section 3.4.1) was used to evaluate search performance on TREC datasets and the Watson system (Section 3.4.2) served as the basis for search experiments and end-to-end QA evaluations on both Jeopardy! questions and TREC data.

3.4.1 Ephyra and OpenEphyra

The Ephyra question answering system [Schlaefer et al., 2006, 2007] developed at Carnegie Mellon University and previously at Universität Karlsruhe is a modular and extensible framework that supports the integration of different QA algorithms and knowledge resources. The primary design goals were to enable experiments with different pipeline setups and to ensure the reusability of individual components. Ephyra's pipeline includes stages for question analysis, query generation, search and answer selection, and thus closely resembles the sample pipeline outlined in Section 3.1. Its current setup combines the following answer extraction approaches:

1. *Answer type based extraction.* During the question analysis phase, a classifier predicts the answer type of the question. For instance, the question may seek the name of a *person*, a *location* or a *date*. Named entity recognizers are applied to text passages retrieved from the knowledge sources to extract candidate answers of the expected type.

2. *Pattern learning and matching.* In an offline learning phase, surface patterns are generated that can be used to recognize various types of relations in text passages. For instance, one set of patterns extracts the *birthdate* of a person, while a different set matches passages naming the *leader* of an organization. During question answering, the questions are categorized by a classification component, and an appropriate set of surface patterns is applied to extract candidate answers from the sources.

		Factoid Accuracy	**List** $F(1)$	**Definition** $F(3)$	**Overall** Per-series
TREC 15	Ephyra	0.196	0.096	0.150	0.143
	Median	0.186	0.087	0.125	0.134
TREC 16	Ephyra	0.206	0.140	0.189	0.181
	Median	0.131	0.085	0.118	0.108

Table 3.4: Evaluation results for Ephyra in the TREC 15 and 16 evaluations.

3. *Semantic parsing.* A semantic parser and ontologies are used to generate shallow semantic representations of questions and sentences retrieved from the sources. The semantic representations are essentially predicate-argument structures augmented with named entity information and related terms. Candidate answers are extracted from sentences that match the semantic structure of the question.

Ephyra has been evaluated in the TREC 15 and 16 question answering tracks [Dang et al., 2006, 2007]. In Table 3.4 we show evaluation results for different question types and compare Ephyra to the median over all TREC participants. More recently, Ephyra was released as open source software to the research community. The open source version OpenEphyra is lightweight and platform-independent, and can be downloaded from SourceForge[4]. OpenEphyra has been used by other researchers as a framework for experiments with new QA algorithms [Bilotti and Nyberg, 2008, Banerjee and Han, 2009], and it served as the basis for participations in cross-lingual QA evaluations [van Zaanen, 2008, Dornescu et al., 2008]. In an internal evaluation on 444 factoid questions from the TREC 11 dataset (excluding questions with no known answers), OpenEphyra had an accuracy of 49%.

3.4.2 Watson and the DeepQA Architecture

Watson [Ferrucci et al., 2010] is a state-of-the-art question answering system developed at IBM Research for the Jeopardy! challenge. The system is based on IBM's DeepQA architecture, a flexible QA framework that facilitates the development of general and reusable QA technologies. The goal of the DeepQA effort is to achieve high-performing results across a wide range of question types and domains that are reproducible by the research community.

Figure 3.2 gives an overview of the DeepQA architecture. DeepQA differs from common TREC systems such as Ephyra in many respects [Ferrucci et al., 2010], but two characteristics are particularly relevant in the context of automatic source expansion:

- During the answer scoring phase, DeepQA retrieves supporting evidence for each candidate that passes an initial filtering stage. The system searches the knowledge sources for text passages that contain both question terms and the

[4] `http://sourceforge.net/projects/openephyra/`

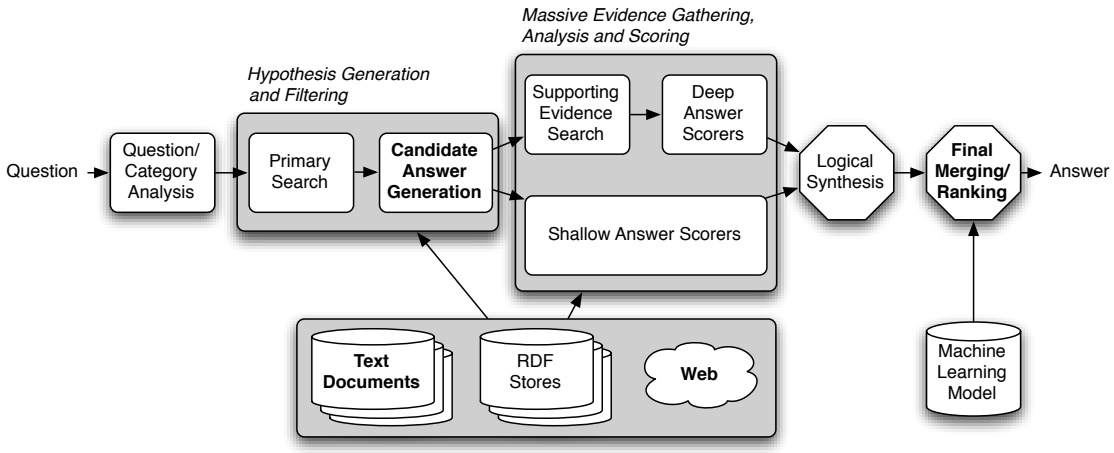


Figure 3.2: DeepQA architecture adopted by Watson.

candidate answer, and computes various features that reflect the lexical, syntactic and semantic similarity of the question and retrieved text. If the sources contain passages that closely resemble the question and contain the candidate answer, the candidate is more likely to be correct. The effectiveness of these features depends on the quality of the retrieved supporting evidence, which can be improved by expanding the sources with additional information and increasing semantic redundancy.

- DeepQA's answer scoring is based on a comprehensive probabilistic model that makes use of more than a hundred deep and shallow features of candidate answers. This model can effectively and reliably rank large numbers of candidates extracted from a variety of resources, including potentially more noisy sources generated through statistical source expansion.

To demonstrate the effectiveness of the question answering technologies that were developed as part of the DeepQA effort, IBM decided to take on the best human contestants in a Jeopardy! match. To compete at the human champion level, a QA system must be capable of answering most Jeopardy! questions with high precision, and to do so within just a few seconds. In Figure 3.3 we illustrate how human players fared in past Jeopardy! episodes. Each of the gray points visualizes the performance of the winner of a Jeopardy! game with three contestants. We refer to these points as the *Winners Cloud*. It can be seen that on average the winners were fast enough to acquire 40–50% of all questions, and that they answered those questions with 85–95% precision. The black points illustrate the performance of Ken Jennings, one of the most successful Jeopardy! players of all time and winner of 74 episodes in a row.

Figure 3.3 also illustrates how Watson performed at different stages in the development process. Watson estimates confidence scores for each of its candidate answers, and based on its confidence in the top candidate decides whether to answer a question. By adjusting the confidence threshold we can control how many questions Watson attempts to answer. Since the confidence estimates are quite reliable, Watson typi-

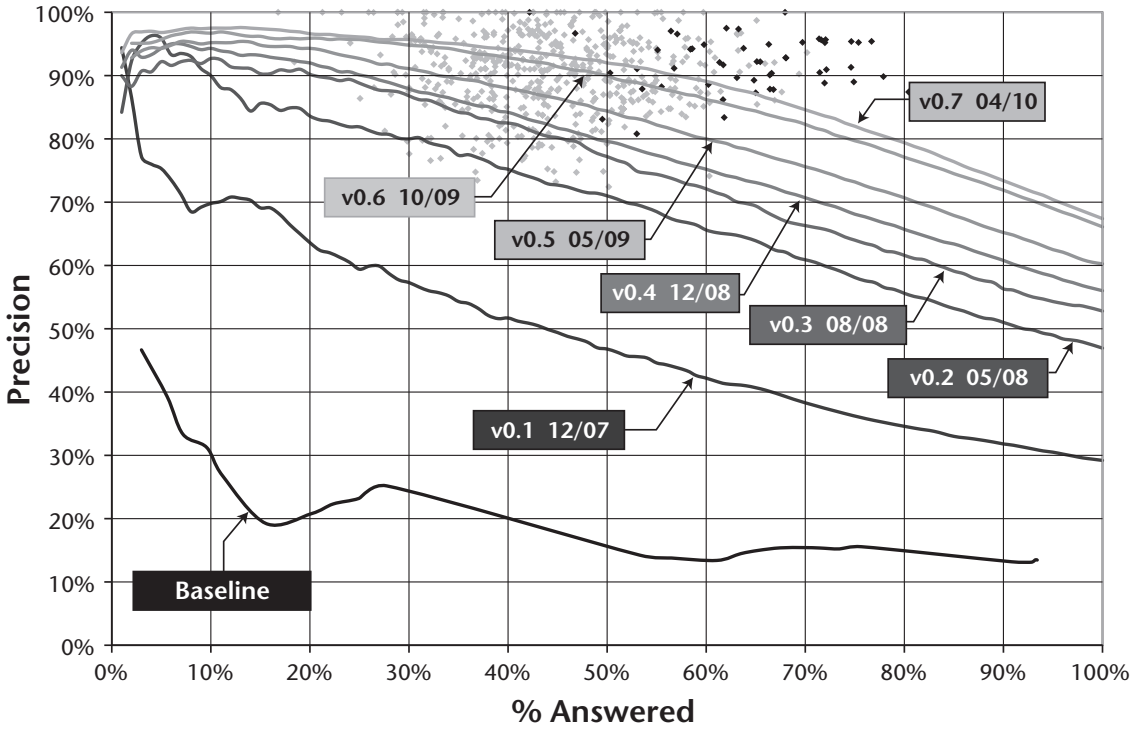


Figure 3.3: Performance of Watson and human contestants at the Jeopardy! task (adopted from Ferrucci et al. [2010]).

cally has higher precision if it only attempts questions with high-confidence answers. Thus Watson's performance can be illustrated as a line that, for the most part, has a decreasing slope. In 2007, a baseline system comprising QA technologies developed previously at IBM for the TREC task fell far short of the performance required to compete with human players. Over the following years performance improved continuously, and by 2010 Watson's precision curve cut through the upper end of the Winners Cloud. This means that Watson was now effective enough to win against most human Jeopardy! players, including some of the strongest contestants. However, Watson's response time had also increased as more and more algorithms were integrated in its processing pipeline, and it now took up to 2 hours to answer a question using a single CPU. To achieve the speedup necessary to compete with humans, the system was deployed on massively parallel hardware. Running on a high-speed cluster with 2,880 processing cores and 15 terabytes of RAM, Watson can answer more than 85% of the Jeopardy! questions within 5 seconds.

At the beginning of 2011, Watson played against two of the strongest Jeopardy! players of all time, Ken Jennings and Brad Rutter, and it won the match. In this thesis, we use a development version of Watson that is very similar to the live system that was deployed in this match. However, while the live system minimizes the response time by parallelizing the execution of each individual question, the development system is optimized to achieve high throughput by processing many questions in parallel and running each question in a single thread. Both versions include the statistical source expansion approach described in this thesis.

With source expansion, the best-performing version of Watson that was available at the time of the competition in early 2011 achieves 71% accuracy on a random sample of 3,508 regular Jeopardy! questions, and 51% accuracy on 788 randomly selected Final Jeopardy! questions. If only questions with high-confidence answers are attempted in the regular Jeopardy! rounds, the system has a precision of 88% while answering about 70% of the questions. On a set of 444 factoid questions from the TREC 11 evaluation, Watson has an accuracy of 64% when only utilizing local corpora that were expanded with web data in a prepocessing step, and 69% when performing live web searches in addition to local searches. As a comparison, OpenEphyra answers the same set of questions with 49% accuracy using live web search results. Without source expansion, Watson's performance is significantly lower. QA accuracy decreases from 71% to 66% on regular Jeopardy! questions and from 51% to 45% on Final Jeopary! questions. If Watson only attempts 70% of the regular Jeopardy! questions based on its confidence estimates, it performs at 84% precision without source expansion, compared to 88% when using expanded sources. On TREC 11, Watson's accuracy decreases from 64% to 59% if source expansion is turned off.

As with most natural language processing projects, the DeepQA research team initially realized large performance gains, but as Watson became more effective at the Jeopardy! task, the improvements grew smaller. The reason for the diminishing returns is that different algorithms often address similar issues, and the incremental performance gains of new methods can depend on the order in which they are added to a system. Source expansion was first integrated in Watson in the summer of 2009 and is responsible for much of the performance difference between Watson v0.5 and v0.6 shown in Figure 3.3. It also had the largest incremental impact on precision and accuracy among all algorithms that were added to Watson in the final years before the Jeopardy! competition. This is because the improvements through source expansion are largely orthogonal to advancements in other QA algorithms, such as question analysis, search, candidate answer extraction and answer scoring. Thus, even though Watson was already very effective at answering Jeopardy! questions, source expansion yielded substantial improvements because other system components benefited from the additional relevant source content provided with this method. For this reason, we believe that source expansion could also have a significant performance impact on other QA systems, and on other natural language processing applications that leverage textual source corpora.

Chapter 4

Source Expansion Approach

The input of the statistical source expansion (SE) algorithm is a topic-oriented corpus, i.e. a collection of documents in which each document contains information about a distinct topic. We refer to each of these documents as a *seed document* or simply *seed*, and to the entire document collection as the *seed corpus*. Examples of pre-existing seed corpora that are suitable for this SE approach are encyclopedias (such as Wikipedia) and dictionaries (such as Wiktionary). For each of the seed documents, a new pseudo-document is generated that contains related information extracted from large external resources. By expanding the seeds, we gather additional relevant information about their topics, as well as paraphrases of information that is contained in the seeds. Our goal is to increase both the coverage and the semantic redundancy of the seed corpus. By semantic redundancy we mean reformulations of existing information, as opposed to lexically similar text or even duplicates that are of no additional benefit. A question answering system can take advantage of both types of related content: it can find answers that were previously not covered in the seeds, and it can retrieve more relevant text for answer extraction and scoring. In Section 6.5 we take a closer look at the relative contributions of coverage and redundancy.

The seed documents are expanded in a four-stage pipeline, illustrated in Figure 4.1 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 4.1). We used the Web as a source of related information for the experiments reported in this thesis. The retrieved documents are split into paragraph- or sentence-length text nuggets (*extraction* stage), and their relevance with regard to the topic of the seed is estimated using a statistical model (*scoring* stage). Finally, a new pseudo-document is compiled from the most relevant nuggets (*merging* stage). Note that this expansion is performed as a separate pre-processing 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.

In Sections 4.1–4.4 we describe each of the pipeline stages in more detail, and in Section 4.5 we give examples of generated pseudo-documents that illustrate how source expansion improves question answering performance. A generalization of this topic-oriented source expansion approach to unstructured seed corpora is discussed in Chapter 7.

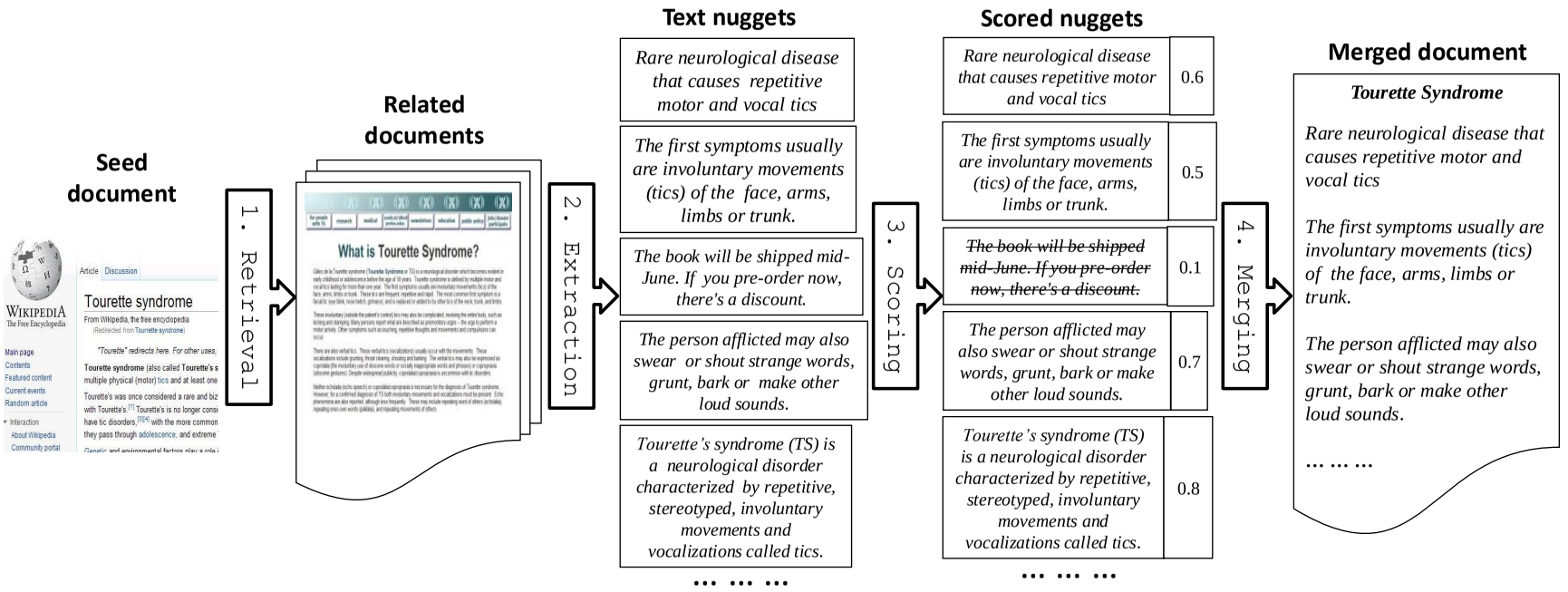


Figure 4.1: Four-stage pipeline for statistical source expansion. In this illustration, the Wikipedia article about *Tourette Syndrome* is expanded using text nuggets extracted from web pages.

4.1 Retrieval

For each seed document, the retrieval component generates a web search query, performs a Yahoo! search for related content, and fetches the web pages for the top 100 search results. The hit list length was determined by the Yahoo! search API, which allowed us to retrieve up to 100 results per query. However, we found that the web pages at the lowest ranks often do not contain much relevant information, and thus it seems unlikely that longer hit lists would significantly improve source expansion performance. On the other hand, shorter hit lists would contain less noise, but we use the search rank as one of the features for relevance estimation, and thus the statistical relevance model can discount text nuggets that were extracted from low-ranking and potentially noisier documents. Our current implementation of the SE algorithm does not support some proprietary document formats such as PDF files and Microsoft Word documents, and thus the number of search results that are actually fetched and parsed can be less than 100.

We also experimented with the retrieval of documents that are directly referenced by the seed documents, e.g. through external links in Wikipedia articles, but found the web search approach to yield more relevant text. When using web pages that are linked from Wikipedia as a source of related text in addition to Yahoo! search results, the final expanded corpus contained very few text nuggets from the linked pages. In experiments with Jeopardy! datasets, search recall did not improve significantly compared to source expansion using only web search results.

In experiments with Wikipedia and Wiktionary, we used the document titles as queries for the Yahoo! searches. This approach works well since in these sources the titles are fully disambiguated descriptions of the topics of documents. In Wikipedia, the titles of articles often contain disambiguation information in parentheses, such as *Washington (U.S. state)* or *Paul Simon (politician)*, which is included in our queries. However, queries can also be generated from topic-oriented documents that do not have descriptive titles by extracting topical terms from their bodies based on markup information or using statistical techniques.

Furthermore, the SE approach can be extended to support unstructured seed corpora in which there exists no one-to-one correspondence between documents and topics, such as newswire corpora or locally stored web crawls. In Chapter 7, we describe how such sources can be transformed into topic-oriented document collections before applying the SE pipeline discussed here. We also demonstrate that it is feasible to extract related text from a large unstructured text corpus without using a document retrieval system, and that the performance of this extraction-based method is comparable to the search-based approach. Thus the search engine is not an integral part of the SE approach, and it is not required for our method to be effective.

4.2 Extraction

The extraction component splits the retrieved documents into paragraph-length text nuggets. For HTML pages, structural markup is used to determine the boundaries.

The text nuggets are then converted to plain text by removing remaining markup and embedded source code. Typical text nuggets are HTML paragraphs, list items or table cells. They range in length from short sentence fragments (e.g. *born in 1968*) to narratives of multiple sentences. Examples can be seen in the expanded documents at the end of Section 4.5.

Since in the merging stage individual text nuggets are selected and added to the pseudo-document, the nuggets should ideally be self-contained and either entirely relevant or entirely irrelevant. To be self-contained, a nugget must not contain references to entities that are not defined within the nugget, such as pronouns or nominals referring to preceding text. This condition is mostly met by nuggets that are delimited by structural markup, since these nuggets tend to be long and can span multiple related sentences. However, long nuggets are often only partially relevant with regard to a given topic, which motivates the use of smaller sized nuggets. Thus we further split the nuggets into sentences and experiment with both markup-based nuggets and sentence-level nuggets. In Chapter 5, we present evaluation results for both nugget granularities and show that markup-based nuggets are generally preferable over individual sentences.

4.3 Scoring

The core component of our source expansion approach is a statistical model that scores the extracted text nuggets based on their relevance to the topic of the seed document. For instance, if the seed is an article about *Barack Obama's inauguration*, the goal is to assign higher relevance scores to text nuggets that precisely match this topic than to nuggets about *inaugurations* or *Barack Obama* in general, or text nuggets about completely unrelated topics and uninformative text such as ads and spam. In the following, we describe our methodology for annotating training and test data for this relevance estimation task (Section 4.3.1). We also discuss features that are predictive of the relevance of text nuggets (Section 4.3.2) and linear models for relevance estimation based on these features (Section 4.3.3). Sequential models that take the context of text nuggets into account when estimating their relevance are the subject of Section 8.2.

4.3.1 Annotation Methodology

To develop effective feature sets and learning methods for the relevance estimation task, we needed to annotate a large amount of high-quality training and test data. We selected a sample of 15 Wikipedia articles and for each of these articles retrieved 100 web pages following the approach described in Section 4.1. When selecting seeds, we focused on articles about unambiguous topics so that it could be decided with relative ease whether a given text passage is relevant to the topic. We further only considered articles that were long enough for the relevance features discussed in Section 4.3.2 to be reliable, and that were at least of marginal interest to the annotators. In addition, we attempted to select a diverse sample of articles about people, things and events.

The seed articles and retrieved web pages were presented to three human annotators, who were given instructions to (1) read the seeds to familiarize themselves with their topics, and (2) label substrings of the web pages that are relevant to the topics of the seeds. To keep the manual labeling costs to a minimum, each topic was only assigned to a single annotator. Therefore, it was important to develop detailed guidelines that enable annotators to make consistent decisions about the relevance of text passages. These guidelines were given in the form of a checklist designed to facilitate decisions in borderline cases, and they were gradually refined as we annotated more data. In the following, we list criteria that a text passage had to satisfy in order to be relevant, or could satisfy while still being considered relevant. The annotators were instructed to select substrings of the retrieved documents that are as long as possible without violating any of these criteria.

In order to be relevant, a text passage must be:

1. **Topical**, i.e. it must be about the topic defined by the seed document or about a directly related topic. For instance, if the topic is *Barack Obama*, then the text snippet *President Obama lives in the White House* is topical and the snippet *the White House is located in Washington, D.C.* is also acceptable since its topic *White House* is directly related to *Barack Obama*.

2. **Informative**, i.e. it must provide information about the topic that is of general interest. This definition is somewhat imprecise, and a good question to ask is *Could this information appear in the seed document, or a more comprehensive version of the seed?* For instance, if the seed is about a famous actor then a passage taken from his blog stating that the actor had a coffee at Starbucks in the morning is not relevant, but a passage that mentions a movie he starred in is relevant. No preference is given to more recent information.

3. **Self-contained**, i.e. it must be a paragraph, sentence, sentence fragment or phrase that is meaningful when viewed independently. For instance, if the topic is *Albert Einstein*, then the snippet *born on March 14, 1879* is self-contained, but the snippet *March 14, 1879* is not self-contained since the date is taken out of its context. The snippet may optionally contain coreferences to the topic (e.g. *he was born in Germany*), but not to other entities that are not mentioned in the snippet (e.g. *she married him in 1903*).

4. **Well-formed**, i.e. it must be a paragraph, sentence, sentence fragment or phrase in proper English. Text passages in languages other than English, passages containing source code in a script language, lists of proper names etc. are not well-formed.

5. **Objective or a substantiated opinion.** For instance, the snippet *Obama made history because he is the first African American president* is substantiated whereas the statement *Obama made history!!!* is unsubstantiated.

6. **Correct at some point in time**, i.e. it should not provide information about the topic that is obviously wrong and has never been true in the past. However,

it is infeasible for annotators to manually verify each and every fact before selecting a text passage.

7. **Not misleading when taken out of context.** For instance, if the topic is *Barack Obama*, a passage about his father (who had the same name) can be misleading when viewed independently.

A text passage is still considered relevant even if it is:

- **Incomplete**, e.g. a sentence fragment or a phrase. For example, if the topic is *Barack Obama*, the snippet *elected president in 2008* is acceptable.
- **About a more specific topic.** For instance, if the topic is an animal species, a text snippet about a subspecies is also acceptable.
- **Ungrammatical or contains spelling errors**, if it can still be understood easily by a human reader.
- **Outdated or in the wrong tense.** For example, if the topic is *Bill Clinton*, the text passage *he is the president of the US* is outdated but acceptable.
- **Redundant**, e.g. it is an exact copy of a passage in the seed document or a passage that has been selected previously.

Note that we permitted redundant text and even exact duplicates since we are not concerned with redundancy filtering in the scoring phase. Instead, a simple filter that measures the token overlap of text nuggets and the seed effectively removes lexically redundant text in the merging phase (Section 4.4).

We developed a graphical interface that allows annotators to browse web pages and label relevant passages that are related to a given seed. In a first attempt, we split the web pages into text nuggets using HTML markup to determine nugget boundaries (see Section 4.2) and asked the annotators to label each nugget as relevant or irrelevant. However, we found that binary decisions about the relevance of paragraph-length nuggets are not ideal since (1) often only a substring of a nugget is relevant, which makes it difficult to decide whether the nugget is a positive instance, and (2) when assigning binary labels it is impossible to change the granularity of the nuggets, e.g. by segmenting them into sentences, without labeling the data again. Therefore, we extended the annotation interface to allow the annotators to select arbitrary regions of relevant text in a web page. A text nugget was then assigned a positive label if any substring of it was selected by an annotator. We also experimented with stricter conditions for relevancy, e.g. requiring at least half of the text nugget to be selected, and we attempted using the fraction of relevant text in a nugget as a continuous measure of its relevance. However, both extensions resulted in a moderate drop in nugget scoring performance.

A screenshot of the annotation GUI is shown in Figure 4.2. The interface displays a web page retrieved for the Wikipedia article on *Abraham Lincoln assassination*. Topic terms are automatically highlighted in red so that relevant sections of the web

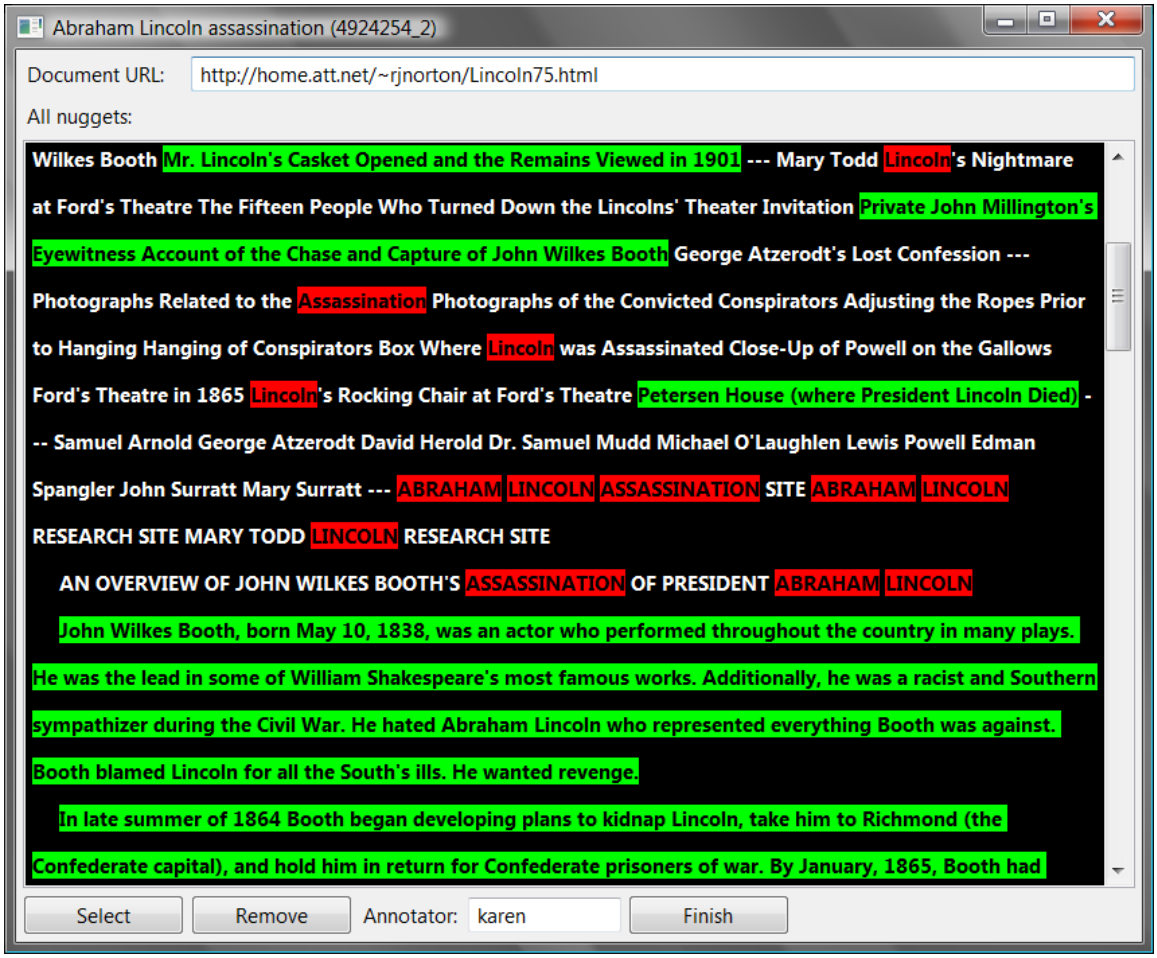


Figure 4.2: Graphical interface for the annotation of relevant content, showing a web page that is related to the topic "Abraham Lincoln assassination". Topic terms are highlighted in red, relevant text passages selected by the annotator in green.

page can be identified more quickly, and substrings that were selected as relevant by the annotator are highlighted in green. Note that even though the interface only requires the annotator to label relevant text, it implicitly assumes all other text to be irrelevant. Thus the annotators were cautioned against missing any positive instances since otherwise the data would contain false negatives.

4.3.2 Relevance Features

A dataset of text nuggets that was annotated following the above methodology and that is described in Section 5.1 was used to fit a statistical model. We experimented with various statistically or linguistically motivated features that estimate the topicality or textual quality of text nuggets and are thus predictive of their relevance. In the following, we list each of the features along with its range (*binary*, *discrete* or *continuous*), whether it is generated at the level of *documents* (and is thus the

same for all nuggets in a document) or at the level of individual *nuggets*, and a brief description and intuitive motivation.

Topicality features:

- *TopicRatioSeed* (continuous, per nugget):
  Likelihood ratio of a text nugget estimated with topic and background language models. The topic language model is estimated from the content of the seed document, the background model from a random sample of about 7,500 Wikipedia articles. We used a subset of Wikipedia to avoid excessive memory consumption. Both language models are unigrams with Good-Turing discounting [Good, 1953], a smoothing technique that assigns non-zero probabilities to words that do not occur in the training data while decreasing the probabilities of rare words that appear in the data. Nuggets with large likelihood ratios are often thematically related to the seed document.

- *TopicRatioNuggets* (continuous, per nugget):
  Similar to *TopicRatioSeed*, but the topic language model is estimated from the text nuggets that were retrieved for the given topic, and the background model from a large sample of nuggets retrieved for different topics. On average, nuggets that were retrieved for the given seed document can be expected to be more topical than nuggets retrieved for other documents, and thus the likelihood ratio is indeed a measure of topicality. This feature is more stable than *TopicRatioSeed* for short seed documents.

- *TFIDFSeed* (continuous, per nugget):
  Average *tf-idf* score of the word tokens in the text nugget. The term frequencies (*tf*) are estimated from the seed, the inverse document frequencies (*idf*) from the same sample of Wikipedia articles used for the *TopicRatioSeed* feature. Tokens with high *tf-idf* scores are often central to the topic, and thus nuggets that contain them are more likely to be topical.

- *TFIDFNuggets* (continuous, per nugget):
  Similar to *TFIDFSeed*, but the *tf* scores are estimated from text nuggets that were retrieved for the given topic, and the *idf* scores from a large sample of nuggets retrieved for different topics. This feature is more stable than *TFIDFSeed* for short seed documents.

- *CosineSim* (continuous, per nugget):
  Cosine similarity between word frequency vectors derived from the text nugget and the seed. Stopwords are removed and the remaining tokens are stemmed. This is a special case of the maximal marginal relevance (MMR) summarization algorithm [Carbonell and Goldstein, 1998, Goldstein et al., 2000], using $\lambda = 1$ (i.e. the most relevant text nuggets are selected regardless of their novelty). Additional details are given in Section 2.4. This method was originally used as a single-strategy baseline for relevance estimation, but we later adopted it as

an additional feature in our statistical models. Nuggets with large cosine scores have a similar word distribution as the seed and are thus likely to be topical.

- *QueryTerms* (continuous, per nugget):
  Fraction of the terms in the query used in the retrieval stage that occur in the text nugget. Each term is weighted with its *idf* score, which is estimated from the same sample of Wikipedia articles used for the *TopicRatioSeed* feature, and weights are normalized to sum to 1. This feature takes into account that nuggets containing query terms that describe the topic are more likely to be topical, and that more weight should be given to rare terms.

- *3rdPersonPronoun* (binary, per nugget):
  Whether the text nugget contains a third person pronoun, which may refer to the topic and thus increases the likelihood of the nugget being topical.

Search features:

- *DocumentRank* (discrete, per document):
  Retrieval rank of the document the text nugget was extracted from. Documents that are ranked high by the search engine usually contain more relevant text.

- *SearchAbstractCoverage* (continuous, per nugget):
  Fraction of the tokens in the abstract generated by the web search engine for the source document of a text nugget that are covered in the nugget. The abstract is the search engine's view of the most relevant text. Nuggets that overlap with the abstract often closely match the query and are more likely to be relevant.

Surface features:

- *KnownTokenRatio* (continuous, per nugget):
  Fraction of the tokens in the text nugget that occur in a large corpus of English text comprising web pages, encyclopedias, dictionaries, newswire corpora and literature. English text nuggets usually contain many known words.

- *Known3GramRatio* (continuous, per nugget):
  Fraction of the token 3-grams in the text nugget that occur in the same English text corpus. This feature is similar to *KnownTokenRatio*, but the tokens must also appear in proper English word order.

- *Avg3GramCount* (continuous, per nugget):
  Average frequency of the token 3-grams in the English text corpus. This feature is similar to *Known3GramRatio* but favors nuggets with 3-grams that are more frequently used in English.

- *NuggetLength* (discrete, per nugget):
  Length of the text nugget measured in tokens. Very short nuggets are usually not relevant.

- *NuggetOffset* (discrete, per nugget):
  Offset of the text nugget in its source document measured in nuggets. Text at the beginning of a document is often more relevant.

- *AvgTokenLength* (continuous, per nugget):
  Average length of the tokens in the nugget. Text in natural language on average contains longer tokens than source code snippets and other character sequences that are not useful for source expansion.

- *TypeTokenRatio* (continuous, per nugget):
  Ratio of the number of distinct tokens over the total number of tokens in the text nugget. Some nuggets have high topicality scores just because they repeat topical terms over and over, but they provide little relevant information. This feature is intended to favor nuggets that are less repetitive.

- *SpecialCharacterRatio* (continuous, per nugget):
  Ratio of the number of special characters over the total number of characters in the nugget. This feature penalizes nuggets that are not well-formed English text, which often contain a large number of special characters.

- *CapitalizationRatio* (continuous, per nugget):
  Ratio of the number of capital letters over the total number of characters in the nugget. Text that is not in English or malformed often contains more upper-case characters than well-formed English text.

- *PunctuationRatio* (continuous, per nugget):
  Ratio of the number of punctuation marks over the total number of characters in the text nugget. Natural English text contains fewer punctuation marks than source code snippets, enumerations and other non-natural text.

The topicality features *TopicRatioSeed*, *TFIDFSeed* and *CosineSim* utilize the body of the seed document, which provides useful information about the relevance of related text that is not available in definitional QA or multi-document summarization tasks. For features that rely on language models to estimate the topicality of text nuggets (*TopicRatioSeed* and *TopicRatioNuggets*), we use simple unigrams with Good-Turing discounting because often not much data is available to fit the topic models.

Note that the Yahoo! search engine is used because it offers a convenient API and the search rankings are fairly reliable. However, Yahoo! could be substituted with any other document retrieval system that returns relevant search results for a given seed corpus, and the search-related features could be replaced with similar features that are based on this retrieval system. Not all search engines return summary snippets for the results in the hit list, which are needed to compute the *SearchAbstractCoverage* feature, but instead other useful metadata may be available. For instance, information retrieval systems for locally indexed text corpora, such as Indri and Lucene, return continuous relevance scores for the retrieved documents, which could be used as an additional search feature.

In Section 5.3 we perform a detailed analysis of the utility of individual features for the task of ranking text nuggets by relevance. The topicality features are generally most useful for separating relevant from irrelevant text, followed by the search-related features. The surface features mostly evaluate textual quality and are less predictive on their own, but they do improve relevance estimation performance when combined with topicality and search-based features in a statistical model.

4.3.3 Relevance Models

Given a dataset of text nuggets with binary relevance judgments, the nugget scoring problem can be solved by regression or binary classification. In the latter case, the probability of the positive class or the distance of an instance from the decision boundary can be used as a continuous relevance measure. Thus, any regression method or any classification approach that yields continuous confidence estimates (e.g. logistic regression, support vector machines, or hidden Markov models) can in principle be used to score the text nuggets.

Initially we fitted a logistic regression (LR) model [Agresti, 2002] using all of the 19 topicality, search and surface features from Section 4.3.2. This model estimates the relevance of each text nugget independently, ignoring its context in the source document it was extracted from. However, when looking at a sample of the retrieved web pages, it became evident that this independence assumption does not hold in practice, but that a text nugget is more likely to be relevant if preceding or following nuggets are relevant. This is because the retrieved web pages often contain sequences of relevant nuggets, such as sentences and paragraphs in a narrative or related items in a list or table. Thus we relaxed the independence assumption by adding features of adjacent text nuggets to the LR model. More precisely, in addition to the 19 original relevance features, we added the 18 nugget-level features (i.e. all relevance features except *DocumentRank*) of the previous nugget and the next nugget, yielding a total of 55 features. The evaluation results in Chapter 5 show that this simple extension of the independent LR model improves relevance estimation performance significantly when scoring the shorter sentence-level text nuggets, and that it also yields a small improvement when scoring longer markup-based nuggets.

We also attempted adding features of more distant text nuggets to the logistic regression model but did not observe consistent performance gains. In addition, we tried capturing dependencies between text nuggets with a stacked learning approach. This method uses an LR model to make independent predictions for each text nugget, and a second LR model to combine the predictions for adjacent instances into more accurate context-sensitive relevance estimates. Interestingly, the performance of the stacked LR approach was almost identical to the model with features of adjacent instances. We chose not to use the stacked method because it requires training and applying two relevance models instead of only one when using adjacent features.

Before fitting the final model used in our experiments with question answering datasets in Chapter 6, we performed greedy backward feature elimination using Akaike's information criterion (AIC) [Akaike, 1974]. AIC is a popular optimality criterion that trades off bias and variance. Let k be the number of parameters in

a model, and $L_{max}(k)$ the maximum data likelihood attained by a model with k parameters. Then the AIC is defined as

$$AIC = 2k - 2\ln(L_{max}(k)).$$

By *minimizing* the AIC, we reward models that fit the data well (i.e. large $L_{max}(k)$), but at the same time we discourage overfitting by preferring models with fewer features (i.e. small k). In greedy backward elimination, an initial relevance model is fitted using all available features and the AIC is computed. In the next step, each feature is removed individually, a model is fitted without the feature and the AIC is calculated. The model with the smallest AIC is then used as a new baseline for the removal of additional features. This procedure is repeated until the AIC cannot be reduced further by dropping more features.

Applied to the extended set of 55 relevance features, the backward elimination algorithm selected 43 features, including all original features except *Avg3GramCount* (which is similar to *Known3GramRatio*), 12 features of the previous nugget and 13 features of the next nugget. Among the selected features of adjacent nuggets are some of the most predictive topicality features (*TopicRatioSeed*, *TFIDFSeed*, *CosineSim* and *3rdPersonPronoun*) and the feature that measures the overlap of a nugget with the search engine summary (*SearchAbstractCoverage*). This intuitively makes sense because those features allow conclusions about the relevance of nearby text. On the other hand, features such as *NuggetLength* and *NuggetOffset* were dropped since they do not add useful information for predicting the relevance of adjacent nuggets.

In preliminary experiments, we also evaluated other models that have been used successfully for text categorization, such as support vector machines (with linear, polynomial and radial basis function kernels), least squares linear regression and naïve Bayes classifiers. However, logistic regression outperformed all of these methods on our relevance estimation task when using mean average precision (MAP) as the performance metric. We also found that this method consistently yields strong results, and in comparison to SVMs it is more efficient at training time. Thus we primarily use logistic regression models in this thesis. The only exception are the active learning experiments in Section 8.1, where we found SVMs to be an interesting alternative to LR because they tend to converge rapidly.

4.4 Merging

Finally, the merging component ranks the text nuggets by their estimated relevance scores in descending order. A lexical redundancy filter traverses the ranking and drops nuggets if they repeat text that is contained in the seed or previous text nuggets. More precisely, a nugget is removed from the ranking if the percentage of its tokens that occur in the seed document or higher ranking nuggets exceeds a predefined threshold. We experimented with different thresholds and found that a value of 95% yields good results, i.e. a nugget must contain at least 5% new tokens. Stricter thresholds increase the effectiveness of the redundancy filter but may also result in useful information being removed.

In addition to removing lexically redundant text, nuggets are 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 document. This second threshold was added to collect more related content for long seed documents, since these seeds are often about particularly important or broad topics. In Section 6.2 we describe how the two thresholds were chosen in our experiments. The remaining nuggets are compiled into new pseudo-documents (one document per seed), which can be indexed and searched along with the original seed corpus, yielding a larger source with increased coverage and reformulations of existing information.

Note that the generated pseudo-documents are intended to complement and reformulate information in the seed documents, rather than repeat text that is already in the seeds. In our experiments with QA datasets, we used the expanded sources as additions to the seed corpora, not as their replacements. We also attempted merging the seeds with their expanded counterparts, expecting a positive impact on document search results since there is a higher chance of finding all query terms in the combined documents. However, search performance was hurt in some cases as relevant seed documents were merged with irrelevant content from pseudo-documents, and overall search recall remained almost unchanged compared to the setup with separate seeds and expanded documents.

As an alternative to sorting the text nuggets in decreasing order of relevance, we tried restoring the order in which the nuggets appear in the retrieved web pages. By preserving the original order, we hoped to break fewer coreferences between adjacent text nuggets and improve passage search results. In practice, however, this had very little impact on search performance because even if the nuggets are left in their original order, there are gaps where irrelevant or redundant text was dropped. In addition, the text nuggets that are included in the pseudo-documents are often several sentences long, and most of the retrieved text passages do not span more than one nugget. Thus the neighborhood of text nuggets in the expanded documents does not affect search results much.

4.5 Examples

Two examples of pseudo-documents generated for the Wikipedia seed articles about *Carnegie Mellon University* and *IBM* are given in Figures 4.3 and 4.4, respectively. In these examples, text nuggets were extracted based on HTML markup and sorted by their estimated confidence scores in descending order. Most of the high-confidence nuggets are highly relevant to the topics. Two exceptions can be seen at the beginning of the document about the topic *IBM* in Figure 4.4. We found that some topicality features and search-based features are less reliable for such short nuggets (cf. the comparison of different nugget granularities in Section 5.3), but since these nuggets take up little space in the expanded documents, they can be expected to have little impact on the effectiveness or efficiency of the SE approach. Note that the actual expanded documents used in our experiments are considerably longer than the ones shown here, and include nuggets with lower relevance estimates.

To better understand how SE improves QA performance, consider the question *What is the name of the rare neurological disease with symptoms such as: involuntary movements (tics), swearing, and incoherent vocalizations (grunts, shouts, etc.)?* (TREC 8, Question 8). The expanded version of the Wikipedia article about *Tourette syndrome* illustrated in Figure 4.1 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 (TS) 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 1.1 that source expansion helps consolidate sources by merging related content into a single document. The expanded document is retrieved by the Watson QA system, enabling it to correctly answer the question. Note that this Wikipedia article was expanded along with many other seed documents in a preprocessing step, without knowing yet which documents will be relevant for answering questions.

Now consider the question *When were the first postage stamps issued in the United States?* (TREC 11, Question 1813). 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 article includes the text nugget *Authorized by Congress the United States Government issued the first adhesive postage stamp on July 1, 1847*, which perfectly matches the question and is among Watson's search results. This nugget also contains the complete answer with the exact date, which was not covered in the seed corpus. The example illustrates how SE adds new information to a corpus, addressing source failures (type 1 in Section 1.1), and how reformulations facilitate the retrieval of relevant text, mitigating search failures (type 2).

<DOC>
<DOCNO>Expanded48093</DOCNO>
<TITLE>Carnegie Mellon University</TITLE>
<TIMESTAMP>2009-07-02T03:05:53</TIMESTAMP>
<TEXT>
<NUGGET SCORE="0.9767" SOURCE="http://www.wordiq.com/definition/Carnegie_Mellon_University">Carnegie Mellon University is a private research university located in Pittsburgh, Pennsylvania. It was formed in 1967 by the union of the Carnegie Institute of Technology (which was "Carnegie Technical Schools" until 1912), founded in 1900 by Andrew Carnegie, and the Mellon Institute of Industrial Research, founded in 1917 by Richard Beatty Mellon. The school is often referred to as CMU. CMU houses the first computer science school and the first drama school in the nation. It also houses one of the best engineering schools, and its business school is consistently ranked among the best in the nation. CMU is famous for its unique interdisciplinary environment and as an innovative leader in education. CMU is affiliated with 12 Nobel Laureates, a rare achievement considering its young age relative to its peers.</NUGGET>
<NUGGET SCORE="0.9666" SOURCE="http://www.stateuniversity.com/universities/PA/Carnegie_Mellon_University.html">In 1900 Andrew Carnegie, a Pittsburgh industrialist and philanthropist, founded Carnegie Institute of Technology and Margaret Morrison Women's College to educate the sons and daughters of local working class families. In 1967 Carnegie's institutions merged with Mellon Institute, founded by Andrew Mellon, and formed Carnegie Mellon University. In 1968 Margaret Morrison was closed and the College of Humanities and Social Sciences was founded, forming the basic model of Carnegie Mellon that is seen today. There are now six colleges within the university: Carnegie Institute of Technology (engineering) (CIT), Mellon College of Science (MCS), School of Computer Science (SCS), Tepper School of Business (Tepper), College of Humanities and Social Sciences (H&SS), and College of Fine Arts (CFA).</NUGGET>
<NUGGET SCORE="0.9397" SOURCE="http://www.topuniversities.com/schools/data/school_profile/default/carnegiemellonuniversity">The university consists of seven colleges and schools: The Carnegie Institute of Technology (engineering), the College of Fine Arts, the College of Humanities and Social Sciences, the Mellon College of Science, the David A. Tepper School of Business, the School of Computer Science and the H. John Heinz III School of Public Policy and Management. Carnegie Mellon also has campuses in California and the Arabian Gulf nation of Qatar and is expanding its international presence in Europe and Asia with master's programs and other educational partnerships.</NUGGET>
<NUGGET SCORE="0.9292" SOURCE="http://www.isri.cs.cmu.edu/">The Institute for Software Research (ISR) in the Carnegie Mellon School of Computer Science (SCS) is the focal point for research and education in Software Engineering (SE) and Computation, Organizations and Society (COS). ISR hosts Ph.D. programs in both of these areas, and about five separate professional MS programs including the Masters in Software Engineering (MSE) program now in its 17th year. ISR is home to approximately thirty faculty members, seventy visitors and staff, forty Ph.D. students, and more than one hundred MS students. ISR is also a focal point in SCS for industry and international collaboration, with substantial joint programs in Korea, India, Australia, and Portugal.</NUGGET>
<NUGGET SCORE="0.9236" SOURCE="http://www.qatar.cmu.edu/about/index.php?pg=history">Five years later, President Richard M. Cyert (1972-90) began a tenure that was characterized by unparalleled growth and development. The university's research budget soared from about $12 million annually in the early 1970s to more than $110 million in the late 1980s. The work of researchers in new fields such as robotics and software engineering helped the university build on its reputation for innovative ideas and pragmatic solutions to the problems of industry and society. Carnegie Mellon began to be recognized as a truly national research university able to attract students from across the nation and around the world.</NUGGET>
<NUGGET SCORE="0.9116" SOURCE="http://www.facebook.com/minifeed.php?id=7701216166">About Carnegie Mellon University in Qatar: With more than a century of academic excellence and innovative research, Carnegie Mellon University is a global leader in education with real-world applications. Continuously top ranked, Carnegie Mellon offers a distinct mix of programs to its 10,000 students at campuses around the globe. ...</NUGGET>
...
</TEXT>
</DOC>

Figure 4.3: Expanded document about *Carnegie Mellon University*.

<DOC>
<DOCNO>Expanded14632</DOCNO>
<TITLE>IBM</TITLE>
<TIMESTAMP>2009-07-01T21:48:33</TIMESTAMP>
<TEXT>
<NUGGET SCORE="0.9999" SOURCE="http://www.ibm.com/support">Search</NUGGET>
<NUGGET SCORE="0.9999" SOURCE="http://www.ibm.com/support">Choose a product:</NUGGET>
<NUGGET SCORE="0.9533" SOURCE="http://www.nyse.com/about/listed/ibm.html">International Business Machines Corporation (IBM) is an information technology (IT) company. The Company's major operations include Global Technology Services segment (GTS), Global Business Services segment (GBS), Software segment, Systems and Technology segment, and Global Financing segment. On January 31, 2008, the Company acquired 100% of Cognos, Inc. On April 3, 2008, IBM acquired 100% of Telelogic, AB. In July 2008, the Company acquired Platform Solutions, Inc. (PSI). In December 2008, its internal global logistics operations were acquired by SNCF Transport and logistics division of Geodis.</NUGGET>
<NUGGET SCORE="0.9069" SOURCE="http://www.wikinvest.com/wiki/IBM">International Business Machines (NYSE: IBM) is a leading global technology firm that offers a variety of products and services in the information technology industry. Their current businesses consist of 5 major divisions: Global Technology Services segment; a Global Business Services segment; a Software segment; a Systems and Technology segment; and a Global Financing segment. In 2006 IBM lost its position as the number one IT company to Hewlett-Packard in terms of annual revenue (difference of $235 million between revenues of HPQ and IBM). In 2008 that lead widened as HP generated $118.3 billion in revenue while IBM's revenue came in at $103.6 billion.</NUGGET>
<NUGGET SCORE="0.8763" SOURCE="http://investing.businessweek.com/research/stocks/snapshot/snapshot.asp?symbol=IBM">International Business Machines Corporation (IBM) develops and manufactures information technology products and services worldwide. Its Global Technology Services segment offers IT infrastructure and business process services, such as strategic outsourcing, integrated technology, business transformation outsourcing, and maintenance. The company's Global Business Services segment provides professional services and application outsourcing services, including consulting and systems integration, and application management. Its Systems and Technology segment offers computing and storage solutions, including servers, disk and tape storage systems and software, semiconductor technology and products...</NUGGET>
<NUGGET SCORE="0.8755" SOURCE="http://searchsystemschannel.techtarget.com/sDefinition/0,,sid99_gci801387,00.html">IBM (International Business Machines) is by far the world's largest information technology company in terms of revenue ($88 billion in 2000) and by most other measures, a position it has held for about the past 50 years. IBM products include hardware and software for a line of business servers, storage products, custom-designed microchips, and application software. Increasingly, IBM derives revenue from a range of consulting and outsourcing services. With the advent of the low-cost microchip, the personal computer, distributed computing, open rather than proprietary standards, and the Internet, IBM has seen its position of dominance challenged as the world of information technology no longer revolves around a single company. Yet investors and competitors continue to be impressed by IBM's long-established base of customers among middle-sized and Fortune 100 businesses and its ability to adapt its products and services to a changing marketplace.</NUGGET>
<NUGGET SCORE="0.8541" SOURCE="http://www.eweek.com/c/a/IT-Infrastructure/IBM-Acquiring-PSI/">The mainframe business proved profitable to IBM in the first quarter of 2008. IBM's System z mainframe business grew 10.4 percent year-over-year with revenues of $1.1 billion in the first quarter, according to IDC. IBM is also looking to develop its mainframe as a tool for consolidating large data centers and for virtualization.</NUGGET>
<NUGGET SCORE="0.8537" SOURCE="http://www.nortel.com/prd/si/ibm.html">IBM and Nortel have a broad-based alliance that is developing and delivering comprehensive next-generation network solutions to both service providers and enterprise customers. With our global reach, scale and expertise, Nortel and IBM can help customers reduce costs, increase productivity, and transform their IT and communications. The alliance focuses on collaborative innovation that leverages IBM services and technology capabilities with Nortels experience and leadership in communications infrastructure and solutions. ...</NUGGET>
...
</TEXT>
</DOC>

Figure 4.4: Expanded document about *IBM*.

Chapter 5

Intrinsic Evaluation

In this chapter we evaluate and compare various relevance estimation strategies that can be deployed in a source expansion system to rank text nuggets by their relevance to the topics of seed documents. In Section 5.1 we introduce a high-quality dataset that was annotated manually and that is used to fit and evaluate statistical relevance models. Section 5.2 gives an overview of different statistical models and baselines for relevance estimation and describes the experimental setup for an intrinsic evaluation of these strategies. In Section 5.3 we evaluate how each strategy performs under ideal conditions, using the high-quality annotated dataset, and in Section 5.4 we examine how different methods are affected by noise. Finally, we take a closer look at current limitations of our relevance estimation approach and areas for further improvements in Section 5.5.

5.1 Dataset

A large dataset of manually annotated web pages was created to evaluate and compare different strategies for estimating the relevance of text nuggets. For a sample of 15 Wikipedia articles about people, things and events, we fetched up to 100 related web pages each, using the retrieval component of the SE system described in Section 4.1. The web pages were shown to human annotators, who were instructed to identify substrings that are relevant with regard to the topics of the seed articles. The annotation task was supported by a graphical interface that was developed specifically for this purpose. Each of the 15 topics was assigned to one of three annotators (A, B, or C). Detailed annotation guidelines were provided in the form of a checklist to aid the annotators in making consistent decisions about the relevance of substrings in the web pages. Text nuggets were automatically labeled as relevant if any of their tokens were selected by an annotator. This approach for determining relevance was used for both markup-based paragraphs and sentences. The annotation methodology is described in more detail in Section 4.3.1.

Table 5.1 shows the topics of the seed articles along with the total number of markup-based text nuggets extracted from the retrieved web pages and the number and percentage of those nuggets that were labeled as relevant. We also specify for

each topic the annotator it was assigned to. The last column indicates how the topics were used in our evaluations: to fit language models (*LM*) or for cross-validations of relevance models (*CV*). We will refer to these labels when detailing the setup of each experiment. The annotated dataset comprises a total of 164,438 text nuggets, out of which 9,221 nuggets (5.6%) were labeled as relevant. Note that we focused on topics that are named entities since these topics are usually less ambiguous and easier to annotate, but we will show in Chapter 6 that our approach is also effective for common nouns that can be found in a dictionary.

To evaluate inter-annotator agreement and to get a better sense of the difficulty of this task and the quality of our dataset, annotator A also labeled the web pages for the topics *Mother Teresa* and *Iran-Iraq War*. The former topic was originally assigned to annotator B, while the latter was labeled by annotator C. We selected topics for which the original annotator found a substantial amount of relevant text to ensure low variance in our assessments of inter-annotator agreement. If there were only a few relevant text nuggets, the annotators could agree or disagree simply by chance. For each of the topics, the agreement was measured at the level of markup-based text nuggets, sentence-level nuggets and nuggets consisting of individual tokens. Recall that we consider a nugget relevant if an annotator labeled any substring of it, and irrelevant otherwise. The text nuggets were divided into four disjoint categories: (1) nuggets that are relevant according to both annotators, (2) nuggets that were labeled as relevant by one annotator but not by the other, (3) nuggets for which the reverse holds, and (4) nuggets that neither annotator considered relevant. The frequencies of these categories are given in Table 5.2 for the topic *Mother Teresa* and in Table 5.3 for the topic *Iran-Iraq War*. As an aggregate measure of inter-annotator agreement, we also report Cohen's κ coefficient [Carletta, 1996].

While there are no universal guidelines for the interpretation of κ statistics, scores above 0.9 are often considered to indicate excellent or almost perfect agreement [Fleiss, 1981, Landis and Koch, 1977]. These interpretations may overstate the level of agreement since there are many instances with inconsistent labels. However, the results do show that there is little ambiguity about the relevance of the vast majority of instances, be they paragraphs, sentences or tokens. We found that the annotation task is fairly straightforward most of the time as we often encountered large blocks of obviously relevant text (e.g. the main body of text in an encyclopedia article about the topic) or irrelevant text (e.g. navigation elements and advertisement in web pages, or entirely irrelevant pages). On the other hand, there are borderline cases such as marginally informative content (e.g. quotes by Mother Teresa, or in-depth reports of weapons deployed in the Iran-Iraq War), information that may be out of context (e.g. headlines and list items such as *International recognition* or *Iranian counteroffensives*) and opinions (e.g. criticism and praise of Mother Teresa, or biased accounts of the Iran-Iraq War). In such cases the annotation guidelines in Section 4.3.1 are particularly helpful, but annotators may still disagree on whether a text snippet is informative (item 2 in the guidelines), self-contained (item 3) or a substantiated opinion (item 5). In some cases we also found it difficult or time-consuming to accurately apply the annotation guidelines. For instance, short snippets of relevant text (e.g. *Mother Teresa of Calcutta*, or *Iran-Iraq War, 1980-1988*) are easily overlooked

Topic	**# Total Nuggets**	**# Relevant Nuggets**		**Annotator**	**Usage**
Jenny Toomey	10,789	145	(1.3%)	*A*	LM
Karriem Riggins	10,120	105	(1.0%)	*A*	LM
Ross Powless	13,410	97	(0.7%)	*A*	LM
Abraham Lincoln assassination	11,632	542	(4.7%)	*B*	CV
Fort Boise	8,756	73	(0.8%)	*B*	CV
Harry Blackmun	9,750	641	(6.6%)	*B*	CV
John Rolfe	10,490	283	(2.7%)	*B*	CV
José de San Martín	11,984	206	(1.7%)	*B*	CV
Mother Teresa	11,507	1,383	(12.0%)	*B*	CV
Vasco Núñez de Balboa	7,939	508	(6.4%)	*B*	CV
Amy Van Dyken	8,502	969	(11.4%)	*C*	CV
Anne Frank	11,360	830	(7.3%)	*C*	CV
Berlin Wall	10,280	1,300	(12.6%)	*C*	CV
Iran-Iraq War	15,193	1,929	(12.7%)	*C*	CV
XYZ Affair	12,726	210	(1.7%)	*C*	CV
$\sum$ LM	34,319	347	(1.0%)	*A*	
$\sum$ CV	130,119	8,874	(6.8%)	*B*, *C*	
All	**164,438**	**9,221**	**(5.6%)**	***A*, *B*, *C***	

Table 5.1: Details and usage of relevance estimation dataset. For each topic, we indicate the total number of markup-based text nuggets, the number of relevant nuggets, the annotator the topic was assigned to, and how the topic is used in our experiments.

	Markup		**Sentence**		**Token**	
	Rel. B	Irrel. B	Rel. B	Irrel. B	Rel. B	Irrel. B
Rel. A	1347	198	3715	323	89277	5612
Irrel. A	36	9926	157	13491	4085	159545
Cohen's κ	0.9085		0.9218		0.9190	

Table 5.2: Inter-annotator agreement on the topic *Mother Teresa*. The agreement between annotators A and B was evaluated at the level of markup-based text nuggets, sentence-level nuggets and nuggets consisting of individual tokens. For each granularity, the table shows the number of instances that were labeled as relevant or irrelevant by the annotators, and Cohen's κ coefficient.

	Markup		**Sentence**		**Token**	
	Rel. C	Irrel. C	Rel. C	Irrel. C	Rel. C	Irrel. C
Rel. A	1739	99	6237	158	164950	3253
Irrel. A	190	13165	428	16656	13015	179269
Cohen's κ	0.9124		0.9379		0.9097	

Table 5.3: Inter-annotator agreement on the topic *Iran-Iraq War*. The agreement between annotators A and C was evaluated at the level of markup-based text nuggets, sentence-level nuggets and nuggets consisting of individual tokens. For each granularity, the table shows the number of instances that were labeled as relevant or irrelevant by the annotators, and Cohen's κ coefficient.

if they appear in long passages of otherwise irrelevant text. Here the annotation interface helps by highlighting topic terms, but we still found that such instances are sometimes missed by annotators.

It is also interesting to note that the results are similar for both topics, even though *Mother Teresa* is much more specific and unambiguous, whereas *Iran-Iraq War* is extremely broad and open-ended. Initially we expected higher agreement on the more constrained topic, but it appears that even for the more general topic the obviously relevant or irrelevant text outweighs the instances on which annotators tend to disagree. This is the case independently of the granularity at which relevance is assessed (paragraphs, sentences or tokens).

Our annotation interface and guidelines seem to suffice for labeling data efficiently and with high agreement between annotators. Ambiguous or hard instances occur relatively infrequently, thus the annotation decisions in these borderline cases have little impact on the overall quality of the dataset. Furthermore, even if annotators disagree on these instances it may not matter much because we are not attempting to develop an accurate classifier, but instead our goal is to rank text nuggets by relevance. If marginally relevant nuggets are not labeled consistently, a model fitted to this data may rank such nuggets lower than instances that are definitely relevant. This could even result in the selection of more useful text for source expansion.

The annotation guidelines were developed to ensure that multiple annotators develop a similar notion of relevance and agree on where to draw the line. If a single annotator labels all data, it may be less important how critical this annotator is when judging relevance. Depending on how much of the training data is annotated as relevant, the confidence scores estimated by a model may overall be somewhat higher or lower. However, we can compensate for such deviations by adjusting the confidence threshold that is applied in the merging phase of the source expansion pipeline.

We found that the choice of topics for the annotation task is extremely important. While it does not seem to matter whether the topics are narrow or broad, it helps to select topics that have high coverage in the sources from which related documents are retrieved. If obscure topics are chosen, the documents mostly contain marginally relevant content that can be hard to label consistently. We further excluded "active" topics such as living people or ongoing events since the search results for these topics often cover news that are currently noteworthy but may be unimportant in the overall context of the topics. For instance, when searching for *Barack Obama*, the retrieved web pages contain news stories about speeches, meetings and other everyday business that may be insignificant from a long-term perspective. It can be difficult and time-consuming for a human annotator to distinguish between unimportant details and truly relevant information when labeling such documents. We recommend inspecting the search results carefully and avoiding topics that may involve frequent difficult annotation decisions. In most knowledge domains there is no shortage of potential topics and we found it well worth the effort to choose carefully.

The intrinsic evaluation results in this chapter and the results on QA datasets in Chapter 6 confirm that our annotated data can be used to fit effective relevance models. We also show in Section 5.4 that the intrinsic performance of these models does not degrade much even if noise is artificially added to the dataset.

5.2 Experimental Setup

Several relevance estimation strategies were evaluated through 12-fold cross-validation on the 12 topics marked with *CV* in Table 5.1. We assigned each topic to a single fold to ensure that we never train and test models on similar instances, thus avoiding biased results. Each approach was applied to both text nuggets delimited by structural HTML markup and sentence-level nuggets (see Section 4.2 for details).

The nuggets were ranked by their relevance estimates in descending order, and performance was measured in terms of mean average precision (*MAP*) and precision-recall curves. We focused on these measures of ranking performance since we were interested in finding models that can effectively rank text nuggets by relevance and that can be used in the source expansion system to select the most relevant text. Precision and recall were computed at the level of individual tokens rather than text nuggets to ensure that results are comparable across different nugget granularities. Thus, precision is the percentage of tokens up to a given cutoff-point in the ranking that were labeled as relevant by an annotator, and recall is the percentage of relevant tokens that were ranked above the cutoff point. Note that classification accuracy is

of limited use as a performance metric for the nugget scoring task because of the imbalance between relevant and irrelevant nuggets. A trivial model that classifies all nuggets as irrelevant already achieves 94% accuracy on markup-based nuggets and 85% accuracy on sentence-level nuggets.

In the following, we present evaluation results on nuggets of both granularities (*Sentence* and *Markup*) for different baseline strategies and statistical models:

- *Baseline 1: Random.* Text nuggets are ranked randomly. Note that each random ranking has a different MAP and precision-recall curve, and we report the average over all possible rankings of the nuggets. To simplify the calculations, we treat each token as an independent text nugget than can be placed anywhere in a ranking. Then at any given rank, the average of the precision values of all possible token rankings is the same as the percentage of tokens in the dataset that are relevant (28.48%).[1]

- *Baseline 2: Round Robin.* Selects the first nugget from all documents, followed by the second nugget, and so forth. We expect this strategy to outperform the random baseline since relevant text is often more concentrated at the top of documents.

- *Baseline 3: Search Rank.* Preserves the ranking of the retrieved documents by the search engine and the order of the text nuggets within the documents. This is a much stronger baseline than random rankings since documents that are ranked higher by the search engine typically contain more relevant nuggets. Note that this baseline is the same for sentence-level nuggets and markup-based nuggets since we do not alter the original order of the text nuggets.

- *Baseline 4: Cosine Sim.* The text nuggets are ranked by their cosine similarity to the seed document. This is a special case of the maximal marginal relevance (MMR) summarization algorithm [Carbonell and Goldstein, 1998, Goldstein et al., 2000], using a parameter value of $\lambda = 1$.
  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 (see Section 2.4). The parameter λ controls the tradeoff between relevance and novelty, and by setting $\lambda = 1$ we select the most relevant nuggets regardless of their novelty. Thus we do not penalize paraphrases of previously selected nuggets, which can be useful to a QA system. In the next section we also explore alternative configurations and confirm that this setup is most effective for the task of ranking text nuggets by relevance.

- *LR Independent.* Logistic regression model using only the original relevance features, described in Section 4.3.3.

[1] The percentage of relevant tokens (28%) is larger than the percentage of relevant text nuggets (7%) because relevant nuggets are on average much longer than irrelevant ones.

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

Table 5.4: MAP of baselines and linear relevance models.

- *LR Adjacent.* Logistic regression model that relaxes the independence assumptions by including features of adjacent text nuggets, also described in Section 4.3.3.

5.3 Results and Comparison

In Table 5.4 we show MAP scores for all ranking methods. Note that *Random* breaks down text nuggets into tokens and *Search Rank* preserves the original order of the nuggets, and thus the performance of these strategies does not depend on the size of the nuggets (individual sentences or markup-based paragraphs). Among the four baselines, *Cosine Sim* is clearly most effective, followed with some distance by the rankings produced by the search engine (*Search Rank*). The logistic regression models outperform the baselines, with the only exception that rankings of markup-based nuggets by cosine similarity have higher precision at very low recall levels (too low to generate expanded documents of reasonable length).

The independent nugget scoring model (*LR Independent*) performs worse than the model with features of adjacent instances (*LR Adjacent*) in terms of MAP and is therefore not further evaluated. Note, however, that the difference in performance between the two models is larger on the sentence-level nuggets, which is not surprising since it is more important to capture the context of a text nugget if the nuggets are shorter. 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 of the features, such as the likelihood ratios and the coverage of the search engine abstract, are less reliable for short text nuggets. The advantage of more fine-grained nugget selection when using sentence-level nuggets does not seem to compensate for the difficulties in estimating their relevance.

Figure 5.1 illustrates the ranking performance of the baselines and logistic regression models in terms of precision-recall curves. We omitted the independent LR models for ease of presentation. In rankings generated by the LR model with adjacent features for markup-based nuggets, relevant text nuggets are highly concentrated at the top ranks. For example, precision exceeds 80% up to a recall level of over 60%. Thus when using this model, it is feasible to expand a seed document with mostly relevant text by selecting nuggets from the top of the ranking.

We used the one-sided Wilcoxon signed-rank test [Wilcoxon, 1945, Siegel, 1956]

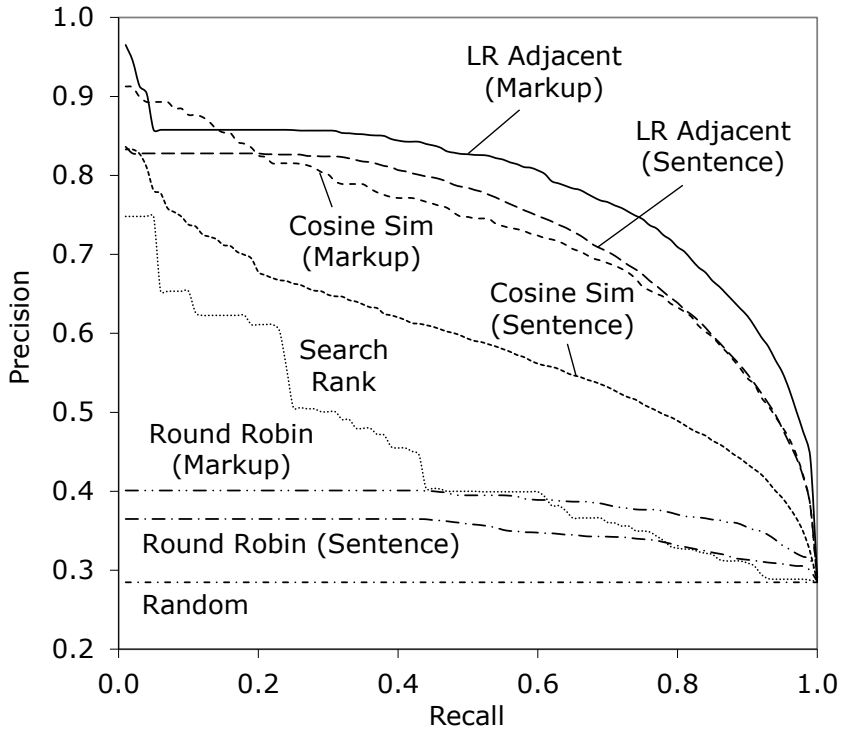


Figure 5.1: Precision-recall curves for baselines and linear relevance models.

to determine whether the performance differences between the baselines and linear relevance models are statistically significant. The Wilcoxon test is a non-parametric method that makes few assumptions about the distribution of the data. It is a better choice for this task than the paired Student's t-test because it does not require the data to be normally distributed. For each ranking strategy, we computed the average precision of the rankings generated for all of the topics used in the cross-validations. We then performed pairwise comparisons of the strategies based on the differences in average precision on each of the 12 topics.[2]

In Table 5.5 we show p-values for both sentence-level text nuggets and markup-based nuggets. In each test, we assumed that the ranking strategy at the left-hand side performs at least as well as the strategy at the top. The null hypothesis is that the strategies are equally effective at ranking nuggets, and the alternative is that the strategy to the left is more effective. It can be seen that almost all improvements are statistically significant at the 1% level. The only exception is the difference between *LR Adjacent* and *LR Independent* on markup nuggets, which is only significant at the 5% level ($p = 0.0134$). The smallest possible p-value in these significance tests is 2.44e-4, which indicates that one method outperformed the other on all 12 topics.

When interpreting these results, one should keep in mind that the outcome of the tests depends on the performance metric used to determine which rankings are better and to quantify the difference in the quality of two rankings. We used average

[2]Our setup is very similar to the significance tests performed by Gopal and Yang [2010] to compare different methods for multi-label classification.

	Random	Round Robin	Search Rank	Cosine Sim	LR Indep.
Round Robin	S: 4.88e-4 M: 2.44e-4	–	–	–	–
Search Rank	S: 2.44e-4 M: 2.44e-4	S: 1.71e-3 M: 8.06e-3	–	–	–
Cosine Sim	S: 2.44e-4 M: 2.44e-4	S: 2.44e-4 M: 2.44e-4	S: 2.44e-4 M: 2.44e-4	–	–
LR Indep.	S: 2.44e-4 M: 2.44e-4	S: 2.44e-4 M: 2.44e-4	S: 2.44e-4 M: 2.44e-4	S: 2.44e-4 M: 2.44e-4	–
LR Adjacent	S: 2.44e-4 M: 2.44e-4	S: 2.44e-4 M: 2.44e-4	S: 2.44e-4 M: 2.44e-4	S: 2.44e-4 M: 4.88e-4	S: 2.44e-4 M: 0.0134

Table 5.5: P-values for all pairs of baselines and linear relevance models, based on a one-sided Wilcoxon signed-rank test. The null hypothesis is that the ranking strategy to the left is as effective as the strategy at the top, the alternative is that it is more effective. We show p-values for both sentence-level text nuggets (S) and markup-based text nuggets (M).

precision, a metric that is common in information retrieval and that is convenient for this task because it summarizes the quality of a ranking in a single performance measure. The disadvantage of this method is its sensitivity to the top of the ranking. If some irrelevant text nuggets are ranked high, average precision can be low but the ranking may still be very effective for source expansion. This is because we compile pseudo-documents from a large number of text nuggets that are ranked above a cutoff point, and a few irrelevant nuggets have little overall impact regardless of their ranks among the nuggets that pass the threshold. As an alternative to average precision, one could compute the precision or recall at the cutoff point that is used by the source expansion system, but the ideal cutoff depends on the application the expanded sources are used for. Ultimately we are interested in selecting a ranking strategy that is effective for a given application, but task-based evaluations can be computationally intensive. Fortunately we found that our intrinsic evaluation results based on average precision closely mirror the actual impact of each method on question answering search performance (see Chapter 6), and thus we can efficiently evaluate different strategies and make an informed choice based on these results.

In Figure 5.2 we show how each of the relevance features used in the statistical models performs if used individually to rank sentence-level text nuggets, and in Figure 5.3 we give similar results for markup-based nuggets. When using only a single feature, nuggets can be ranked by their feature values in decreasing or increasing order, depending on whether instances with larger values are more likely or less likely to be relevant. We report the MAP score of whichever ranking is more effective and indicate for each feature whether it was used to rank nuggets in descending (+) or ascending (-) order. For example, text nuggets are more often relevant if they have a large cosine similarity score, and they are also more likely to be relevant if the

document rank is small. Ties between identical feature values were resolved by using the original ranking of the text nuggets in the search results (i.e. the baseline *Search Rank*). All of the features except *DocumentRank* are computed at the level of individual text nuggets and can have different values for each nugget in a document. For those features, we also show the ranking performance if the value of each instance is substituted with the value of the previous or next instance. These results allow us to draw conclusions about the effectiveness of adjacent features in the model *LR Adjacent*.

It can be seen that some relevance features are very effective when used individually. For example, when adopting the source expansion approach, one could initially use only *CosineSim* to rank text nuggets by their relevance. This feature, when used on its own, is the same as the *Cosine Sim* baseline in Table 5.4 and Figure 5.1. Its performance (74.75% MAP on markup-based text nuggets) comes relatively close to the best statistical model (80.59% MAP), and it is easy to implement and does not require training data. However, this method heavily relies on the content of the seed document since it measures the similarity between the seed and the text nuggets. In the next section we will see that the approach is only effective for the relevance estimation task if the seeds are long and of high quality.

Other features that could be used independently to rank text nuggets include *TopicRatioSeed* (69.40% MAP) and *3rdPersonPronoun* (59.53% MAP). The latter is a binary feature, but it is still effective since we use the *Search Rank* baseline to break ties. Thus nuggets that contain a third person pronoun are ranked higher than the other nuggets, but within each of the two groups the original order of the search results is preserved. This ranking is effective because third person pronouns mostly appear in well-formed English text and are often references to the topic of interest. On the other hand, most of the surface features, such as the character ratios and the average length of the tokens in a nugget, are not useful on their own. These features have MAP scores of around 30% or less, which is similar to the average performance of random rankings (28.48% MAP). However, we found that the surface features are still useful in conjunction with topicality and search features in a statistical model.

Figures 5.2 and 5.3 also show that the features of the previous and next nugget can sometimes be good relevance indicators. In particular, text nuggets are often relevant if adjacent nuggets have large topicality feature values, or if the adjacent text has many tokens in common with the search abstract (i.e. large *SearchAbstractCoverage*). Of course the features of adjacent instances can be highly correlated with the features of the text nugget that is being ranked, but they still provide additional useful information. Otherwise the logistic regression model that includes adjacent features (*LR Adjacent*) would not outperform the model that is based only on features of the current nugget (*LR Independent*).

We have seen that the *Cosine Sim* baseline is equivalent to using the maximal marginal relevance (MMR) summarization algorithm with $\lambda = 1$. The parameter λ controls the tradeoff between relevant and novel information in the generated rankings of text nuggets. If λ is close to 1 the algorithm selects relevant text nuggets even if they contain redundant information, and if λ is close to 0 a diverse sample of text is selected regardless of its relevance. In Figure 5.4 we illustrate the ranking performance

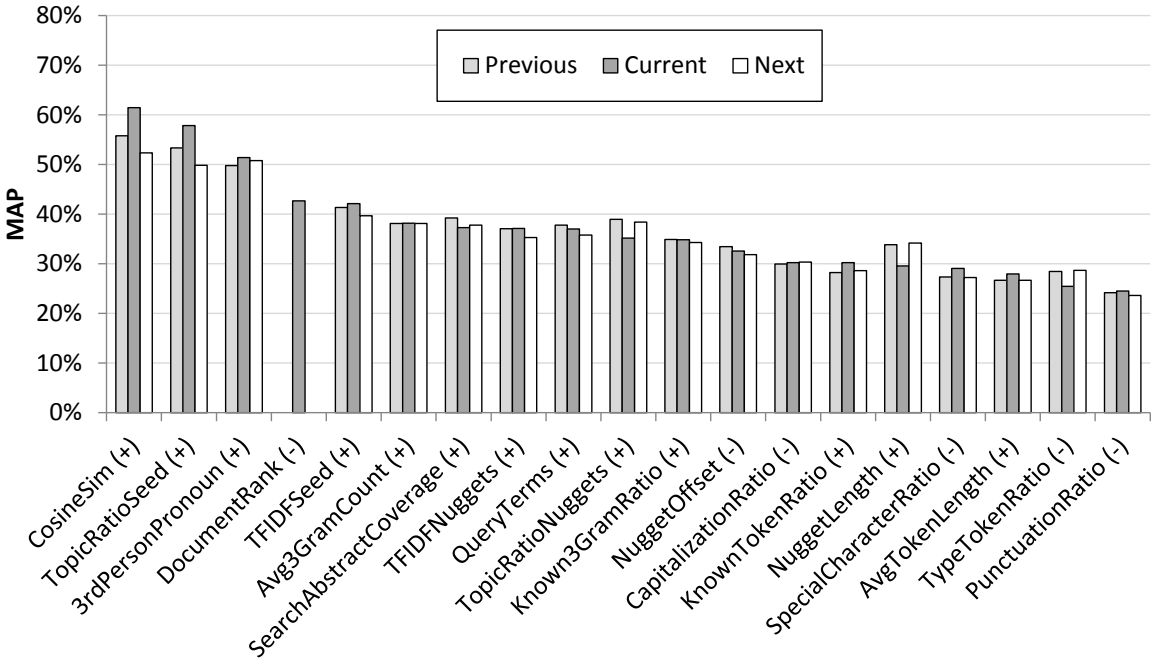


Figure 5.2: MAP of individual features on sentence-length text nuggets. We ranked nuggets by feature values in descending (+) or ascending (-) order. For nugget-level features, we also show the MAP score if the previous or next feature value is used.

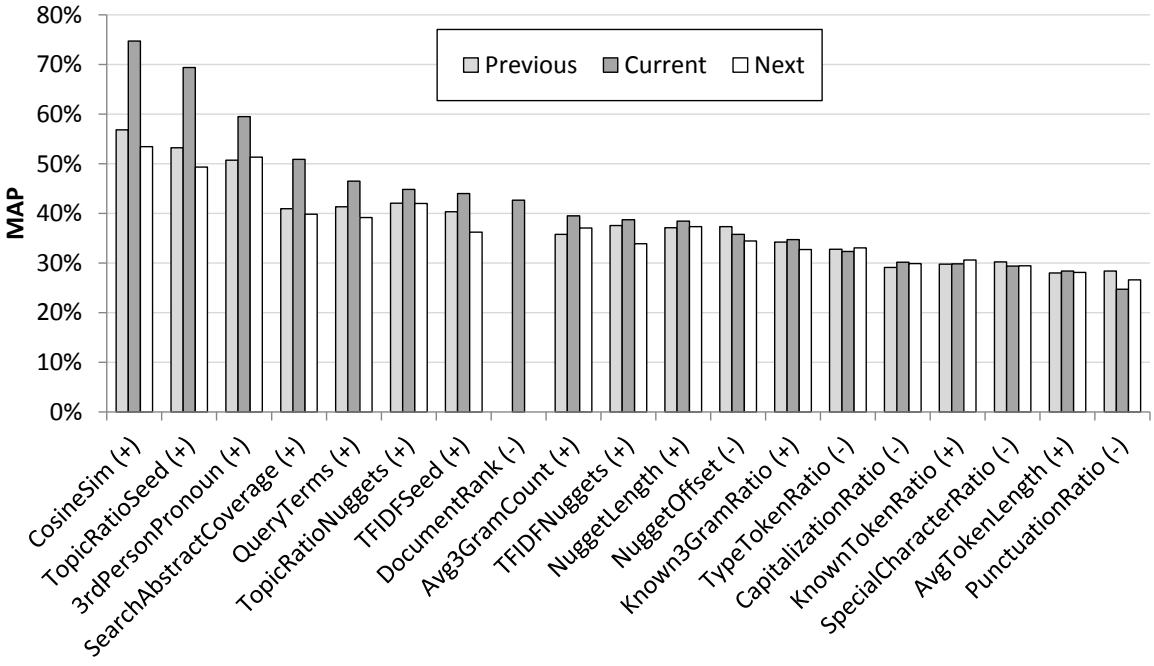


Figure 5.3: MAP of individual features on markup-based text nuggets. We ranked nuggets by feature values in descending (+) or ascending (-) order. For nugget-level features, we also show the MAP score if the previous or next feature value is used.

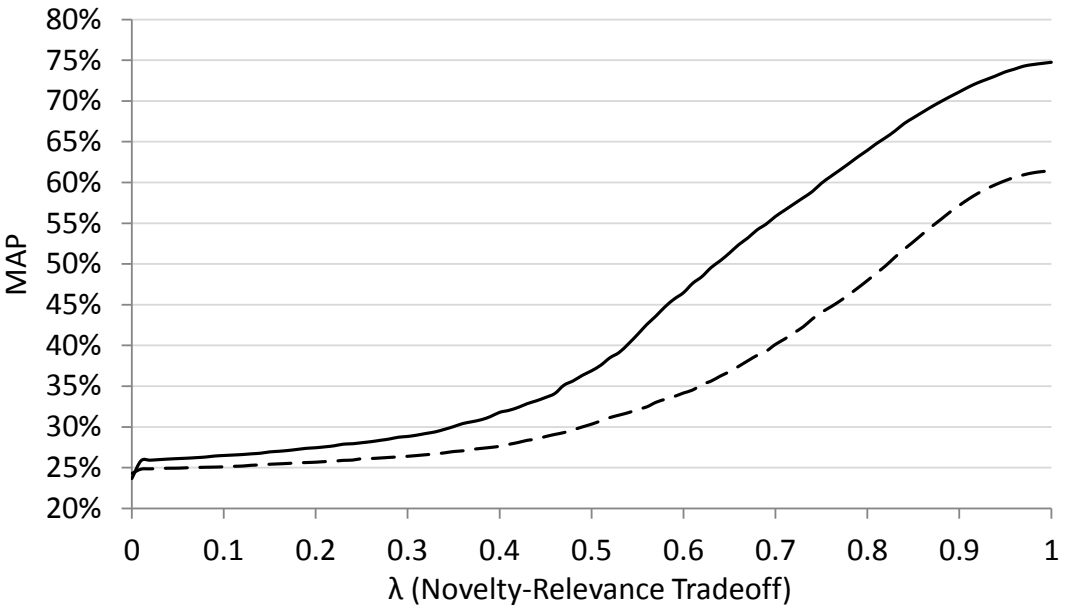


Figure 5.4: MAP of the MMR algorithm for different tradeoffs between novelty and relevance. The solid line is for markup-based nuggets, the dashed line for sentence-level nuggets.

of the MMR algorithm for sentence-length and markup-based text nuggets when using varying λ values. Since in this evaluation we do not penalize redundancy in the rankings, our initial choice of $\lambda = 1$ performs best. We also tried inverting the sign of the novelty component in the MMR algorithm, turning it into an estimate of the similarity of text nuggets to already selected nuggets. This modified algorithm selects text that is related to both the seed document and higher ranking nuggets. Again we experimented with different λ values and found that the optimum is near $\lambda = 1$, i.e. we should select the most relevant text regardless of its redundancy. However, depending on the application that leverages the expanded content, different parameter settings could be more effective. For instance, QA systems benefit from new information that is not covered in a seed corpus in addition to paraphrases of already available information, and thus a configuration of the MMR algorithm that rewards novelty may yield better results. We leave the evaluation of the impact of different relevance–novelty tradeoffs on QA performance as a possible direction for future work.

Note that we compute the cosine similarity between text nuggets and the whole seed document, which is equivalent to using the entire seed as a query in the MMR algorithm. In preliminary experiments we also attempted using shorter queries and found that this hurt relevance ranking performance considerably. For instance, if only the seed document title is used as a query, MAP drops from 61.46% to 34.85% when ranking sentence-length nuggets and from 74.75% to 37.44% when using markup-based nuggets. This again shows how important it is to fully leverage the seed content for relevance estimation.

Initially we used cosine similarities only as a baseline for ranking text nuggets, but seeing how effective this approach is for relevance estimation, we adopted it as an additional feature in the logistic regression models. The new cosine similarity feature improved MAP of the best-performing model *LR Adjacent* by 3.45 percentage points on sentence-length nuggets and by 2.47 percentage points on markup nuggets. The evaluation results shown in Table 5.4 and Figure 5.1 already reflect the improvements. This illustrates the flexibility of the statistical approach for relevance estimation. Any new method that estimates relevance differently, or that uses additional resources (e.g. external text corpora or ontologies), can easily be incorporated into a statistical model as a new feature. The extended model will, in all likelihood, be more effective than a single-strategy approach that uses the same method individually.

5.4 Robustness of Relevance Estimation

We have seen that it is possible to label training data for the relevance estimation task with high inter-annotator agreement, and that this data can be used to fit effective relevance models. But what if lower quality data is used, either because the annotation guidelines were not strictly followed, or because more ambiguous text was labeled? To answer this question, we artificially added label noise to our dataset and evaluated the robustness of logistic regression models in the presence of noise. In these experiments, the best-performing approach *LR Adjacent* was applied to markup-based nuggets. We performed a series of cross-validations in which the labels of relevant nuggets in the training folds were flipped with increasing probabilities $P_{Flip}(\text{Rel})$. To ensure that the total number of positive instances remained roughly the same, we also flipped the labels of irrelevant training nuggets with probability

$$P_{Flip}(\text{Irrel}) = P_{Flip}(\text{Rel}) \times \frac{\#\text{ relevant nuggets}}{\#\text{ irrelevant nuggets}}.$$

By flipping labels in both directions, we take into account that annotators not only overlook relevant text but also mistakenly select irrelevant text.

Figure 5.5 shows MAP scores for varying error probabilities $P_{Flip}(\text{Rel})$. It can be seen that relevance estimation performance is not very sensitive to label noise. For example, even if an annotator mislabeled half of the relevant nuggets and a similar number of irrelevant nuggets, MAP would only degrade from 80.59% to 80.11%.[3] This result may at first seem counterintuitive, but note that there would still be a strong signal in the training data because of the imbalance between positive and negative instances (only about 7% of all text nuggets are relevant). In the example, 50% of the text nuggets in the positive class would actually be relevant, whereas less than 4% of the nuggets in the negative class would be relevant. If even more label noise is added to the dataset, the variance of the evaluation results increases and ranking performance eventually drops off sharply. The dashed horizontal line illustrates the average MAP if text nuggets are ranked randomly (28.48%). At $P_{Flip}(\text{Rel}) = 0.93$

[3]Here we assume that the labels in our dataset are correct.

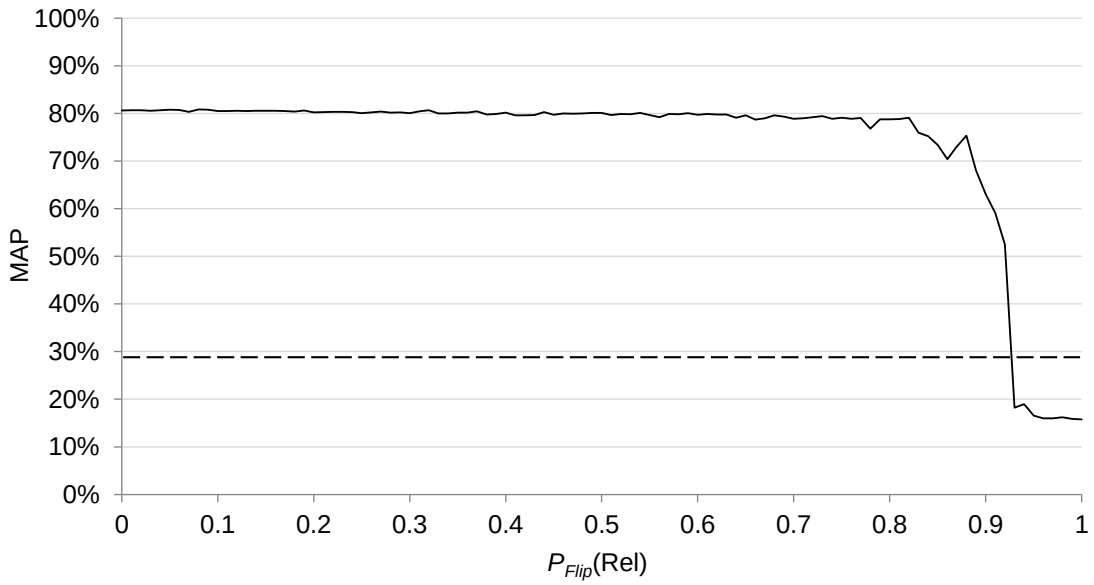


Figure 5.5: Effect of label noise on relevance ranking performance. The solid curve illustrates the MAP of logistic regression models trained on datasets with varying degrees of label noise. The dashed line is the average MAP of random rankings.

the performance of the LR model drops below the random baseline. This makes sense since in this setting about 93% of the relevant text nuggets in the positive class are replaced with irrelevant nuggets. Thus only 7% of the nuggets in the positive class are really relevant, which is about the same as the percentage of relevant nuggets in the entire dataset and the percentage in the negative class. This means that there is no signal in the training data. If additional labels are flipped, the model ranks text nuggets in reverse order of relevance, and performance drops below random.

Most of the relevant text nuggets closely match the word distribution of the seed, are ranked high by the search engine and consist of well-formed English text. These nuggets usually have similar topicality, search and surface feature values and thus form a relatively compact cluster in the feature space. On the other hand, some nuggets are relevant even though they are dissimilar from the seed, their source documents were retrieved at low ranks, or they are of low textual quality. These instances are outliers in the dataset that do not fall within the cluster of "typical" relevant text nuggets. In practice, labeling errors do not occur uniformly but annotators more often err in these unlikely cases. For instance, annotators may overlook relevant information that is not mentioned in the seed or may be less diligent when labeling low-ranking search results. These types of errors seem even less problematic than random label noise since a model that fails in such cases may still be useful for selecting most of the clearly relevant text nuggets. In other words, it may be of little consequence if outliers are misclassified as long as the model can identify text nuggets that belong to the main cluster of relevant instances.

Both the logistic regression models and the cosine similarity baseline leverage the

content of seed documents to estimate the relevance of text nuggets to the topics of the seeds. Thus the effectiveness of these methods depends on the quality and length of the seed documents. To evaluate the extent to which both approaches are affected by the amount of useful content and noise, we artificially degraded the Wikipedia seeds in our annotated dataset in two different manners:

1. We randomly dropped tokens from the seed documents.

2. We replaced tokens in the seeds at random with tokens drawn uniformly from a collection of 7,500 randomly selected Wikipedia articles.

Tokens were dropped or replaced with probabilities ranging from 0 to 1 (in 0.1 increments). In each step, we recomputed the features that depend on the seeds (*TopicRatioSeed*, *TFIDFSeed* and *CosineSim*) using the degraded seed content, and we evaluated the ranking performance of the logistic regression model with features of adjacent instances (*LR Adjacent*) and the *Cosine Sim* baseline. LR models were evaluated through 12-fold cross-validation on the topics in Table 5.1 marked as *CV*. The baseline does not require training, and thus we simply ranked the text nuggets for each of the 12 topics by their cosine similarity scores in descending order.

Figure 5.6 illustrates how the length of the seeds impacts the two ranking approaches, and Figure 5.7 shows how their performance is affected by noise in the seed documents. It can be seen that both methods are more effective for long, high-quality seeds. If few tokens are dropped or replaced, relevance ranking performance remains almost constant, which indicates that both approaches are relatively robust. It can also be seen that replacing seed content with random noise hurts performance somewhat more than removing the content.

The logistic regression model consistently outperforms the cosine similarity baseline, and the performance gap widens if seeds are very short or noisy. This makes sense because the LR model does not solely rely on the seed content but leverages a host of other features that are based on the query used to retrieve related content, the search results and the surface forms of the text nuggets. Even if the seed content is replaced or dropped entirely, the LR model still performs at 67% MAP, whereas the performance of the *Cosine Sim* baseline degrades to about 43% MAP. This is similar to the *Search Rank* baseline because we used this strategy as a fallback to rank nuggets if the cosine similarity scores were not unique or could not be computed because the whole seed content was dropped.

We simulated noise by sampling tokens at random from a background document collection. In practice, however, the noise may not be random but there can be a bias towards specific topics. For instance, articles in an encyclopedia often contain lists of references and links, and the entries in a dictionary may all contain subheadings such as “Pronunciation” or “Translations”. This could bias the selection of relevant content through topicality features that use the word distribution in the seeds, and it may hurt performance more than random noise. However, the worst case impact on the LR model is limited: even if no usable seed content is available at all, the model still performs at 67% MAP.

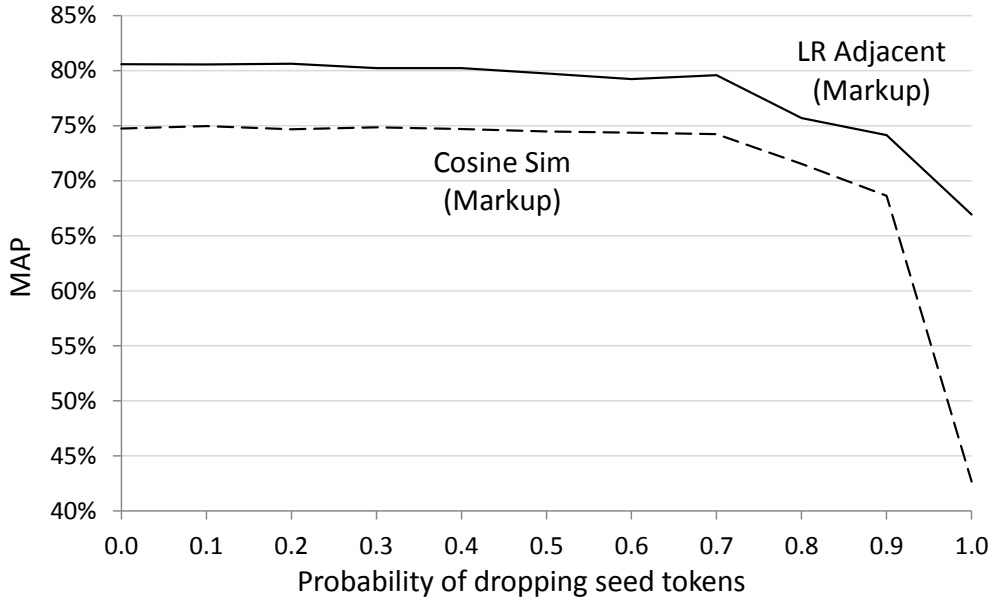


Figure 5.6: Effect of the seed document length on the ranking performance of logistic regression models and the cosine similarity baseline. The plot illustrates how MAP degrades as tokens in the seeds are dropped with increasing probabilities.

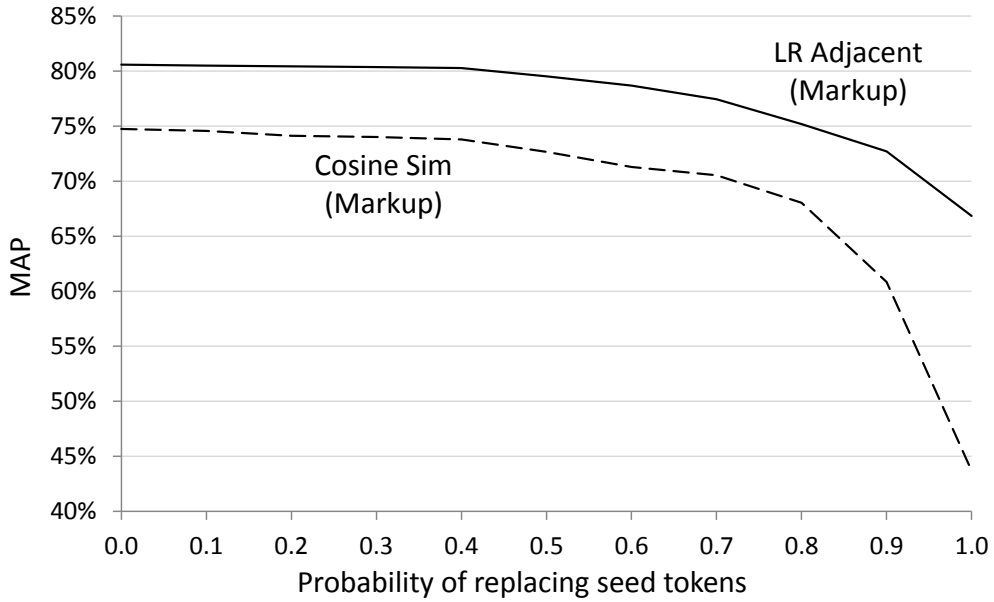


Figure 5.7: Effect of noise in seed documents on the ranking performance of logistic regression models and the cosine similarity baseline. The plot illustrates how MAP degrades as tokens in the seeds are replaced by noise with increasing probabilities.

Based on these experiments, we recommend using a statistical relevance model if enough training data is available or can be annotated. Otherwise the *Cosine Sim* baseline may be a viable alternative, but this method should be used with caution if the seed documents are short or of low quality. We will confirm these conclusions in Section 6.3 by showing that the statistical model also yields higher QA search recall, and that the gain over the baseline is larger when expanding short Wiktionary entries than when using long Wikipedia articles as seeds.

5.5 Error Analysis

To better understand under which conditions the statistical relevance estimation approach fails and to identify opportunities for further improvements, we analyzed the rankings generated by the best-performing model *LR Adjacent* for markup-based text nuggets in the hand-labeled dataset. In particular, we took a closer look at nuggets that were ranked high by the model but were labeled as irrelevant by a human annotator (false positives), and nuggets that were ranked low but were annotated as relevant (false negatives). Many of these text nuggets are borderline cases that could be labeled differently by other annotators, or instances where the annotator clearly made a mistake. Thus we somewhat underestimated the ranking performance of the statistical models and the baselines in Sections 5.3 and 5.4. However, we also found examples of false positives and false negatives that reveal limitations of the current relevance model and feature set.

Some text nuggets do not provide much useful information but are ranked high by the model because they contain many topical terms that also appear in the seed document and because they consist of well-formed English text. For example, these nuggets have high confidence scores but are largely irrelevant:

- **Topic:** Berlin Wall
  Score: 0.8661
  Freedom wins! View the photos of this historic occasion as witnessed by four American adventurers who joined East Germans in their introduction to freedom and who helped tear down the Wall. Read the adventures of The Berlin Wall Freedom Expedition. Post your stories from this historic period. Buy a piece of the Berlin Wall as a symbol of the end of the Cold War and the triumph of freedom.

- **Topic:** Mother Teresa
  Score: 0.7386
  The more than 180 fine art quality tri-tone photographs, along with spiritual counsel from Mother Teresa, will provide a lifetime of rich material for prayer and meditation. Also included and published for the first time ever, with Mother Teresa's special permission, is an appendix containing the contents of the Missionaries of Charity daily prayer book as well as a most personal and profound letter on the interior life written by Mother Teresa during Holy Week of 1993 and addressed to her entire order. Though meant originally as an instruction and appeal to those in her order, this "I Thirst" letter is certain to become a source of spiritual light and encouragement, drawing innumerable hearts and souls closer to God.

Both text nuggets are of high textual quality and contain multiple direct references to their topics *Berlin Wall* and *Mother Teresa*. The first nugget also was extracted from a document that ranked relatively high (15^{th}) in the Yahoo! hit list when using the topic as a query. In addition, both nuggets contain terms that are clearly associated with their topics, and thus topicality features such as cosine similarities, *tf-idf* scores and topic likelihood ratios have high values. For instance, the first nugget mentions the terms *Freedom*, *historic*, *American*, *East Germans*, *tear down*, *symbol* and *Cold War*, which all appear in the seed document. The second nugget also has several terms in common with its seed, including *spiritual*, *lifetime*, *prayer*, *Missionaries of Charity*, *letter*, *Holy*, *order* and *God*. Because the current set of topicality features mostly relies on term frequencies in the seed document and retrieved text, the relevance model cannot distinguish between these nuggets and text that contains useful information. A deeper semantic analysis may reveal that neither of the examples provides much relevant content beyond mentioning the topic and related terms. However, compared to the statistical features used in the current model, deeper natural language processing may be brittle and computationally intensive.

Another limitation of the current relevance model is that it cannot distinguish between factual information and subjective accounts or opinions. For example, these text nuggets were ranked high even though they clearly state opinions rather than generally accepted facts:

- **Topic:** Iran-Iraq War
  Score: 0.7166
  It is understandable that Western governments should feel uneasy and prefer to water down their involvement with the Iraqi regime during the Iran-Iraq war. After all, who can argue that Iranian pride doesn't stand to gain the most from the failure of the Iraqi (US and UK) invasion of Iran? The entire strategy has always been that Iranian pride should never be allowed to flourish...

- **Topic:** Anne Frank
  Score: 0.7713
  The Western World has for some years been made aware of a Jewish girl through the medium of what purports to be her personally written story, "Anne Frank's Diary." Any informed literary inspection of this book would have shown it to have been impossible as the work of a teenager.

To penalize subjectivity, the current feature set could be augmented with a relevance feature that indicates whether a text nugget contains opinions. Such a feature could be derived from a rule-based subjectivity analysis approach that uses a sentiment lexicon [Wiebe and Riloff, 2005], or a statistical classifier that is trained on a corpus with opinion annotations [Pang and Lee, 2004].

Often, the statistical relevance model assigns high scores to text nuggets containing personal accounts or quotations that are about the topic of interest but that are too detailed to warrant their inclusion in the expanded corpus. For instance, a nugget may describe a historic event or an encounter with a famous person from the perspective of an eye witness. Here are two examples:

- **Topic:** Berlin Wall
  Score: 0.8554
  We drove first to Brandenburgerplatz, where the statute of Winged Victory stands atop a 50 meter column, which celebrates a military victory in the 1890s over Denmark. Cars were abandoned everywhere, wherever there was space. Over 5,000 people were there. I began talking to people. We left the car and began to walk through a village of television trucks, giant satellite dishes, emergency generators, and coils of cables, and tents. Cameramen slept under satellite dishes. At the wall, West German police and military was lined up to prevent chaos. West German military trucks were lined up against the wall, to protect it from the West Germans. Hundreds of West German police stood in rows with their tall shields. On top of the wall, lined up at parade rest, stood East German soldiers with their rifles. Groups of West Germans stood around fires that they had built. No one knew what was going on.

- **Topic:** Mother Teresa
  Score: 0.3985
  We were attending a private Mass for the families and Sisters who would take their Final Vows, June 1995, in Washington, when Mother Teresa spotted this little baby in the crowd and took her into her arms, blessing her and singing a lullaby.

The first text nugget is a personal account of the fall of the *Berlin Wall*. It consists of well-formed English text and contains many key terms that are associated with the topic, but most of the information is too specific to be useful for source expansion. The second nugget describes an encounter with *Mother Teresa* that was important to the narrator, but that is arguably not a key event in Mother Teresa's life. Unfortunately it is difficult to automatically distinguish such nuggets from more informative text. If some of the facts in a nugget are not mentioned in the seed document, it may be that they are too specific or they could in fact be useful additions to the seed. Human annotators can decide on their importance using common sense and general knowledge, but the SE system relies on redundancy to determine which text nuggets are most relevant. If a nugget shares a sufficient number of key terms with the seed document or other retrieved text, it is likely to be included in the expanded sources even if it contains unimportant details about the topic. However, these particular examples of false positives might be avoided by adding a feature that indicates whether a text nugget contains first person pronouns. This feature could penalize quotes and narratives that are often overly detailed.

Finally, in spite of all the surface features that are designed to evaluate the textual quality of text nuggets, sometimes a malformed nugget is ranked high just because it closely mirrors the word distribution in the seed. For example, the following nugget has a relatively high confidence score even though it is nonsensical because it contains many terms that are associated with the topic:

- **Topic:** Mother Teresa
  Score: 0.5562
  apologetic s biography calcutta catechism catholic catholic church catholic reading catholic teaching catholic today catholicis m charity charles curran christian inspiratio nal christian living christiani ty dark night of the soul deus encouragem ent faith hope

> india inspiratio n inspiratio nal jesus leadership love morals mysticism nonfiction peace philosophy poor pope pope benedict pope benedict xvi pope john paul ii poverty prayer religion religion and spirituali ty roman catholic roman catholicis m saint saints scott hahn self-help spiritual spiritual growth spirituali ty women

There are few such instances in the labeled dataset, and this type of failure may simply be avoided by including more low-quality, noisy text in the training data. This should yield a model that gives more weight to some of the surface features, such as the ratios of known words and trigrams, and that imposes a higher penalty on malformed text.

On the other hand, the logistic regression model sometimes assigns a low relevance score to a text nugget that covers an important aspect of the topic. This happens if the aspect is not mentioned in the seed, or if it is only mentioned in passing or using different terminology. In either case the nugget shares few key terms with the seed and thus has low topicality feature values. For example, the following nuggets are all ranked very low despite being relevant:

- **Topic:** Berlin Wall
  Score: 0.1135
  Margaret Thatcher held an impromptu press conference outside of her official residence, No. 10 Downing Street, on the morning following the initial opening of the Berlin Wall. In her remarks, it is clear that she is hesitant to reply directly to the idea of a unified German state. Instead, she expressed a desire to move slowly and to facilitate the internal growth of democracy from within East Germany...

- **Topic:** Iran-Iraq War
  Score: 0.0216
  The Iraqi Navy numbered about 4,000 men and consisted of submarine chasers, patrol boats, missile boats, torpedo boats and minesweepers of Soviet, British and Yugoslavian origin. The two main naval bases were at Basra and Um Qasr, neither of which is in a secure position militarily.

- **Topic:** Mother Teresa
  Score: 0.0390
  She received an honorary PhD in Theology (Doctor Honoris Causa in Theology), University of Cambridge in England.

The first text nugget is about the reaction of the British government to the fall of the *Berlin Wall*, which our annotator considered to be a relevant aspect of the topic but which is completely absent in the seed document. The second example describes Iraq's navy during the *Iran-Iraq War*, which again is arguably an important aspect of the topic but it is only covered superficially in the seed. The final nugget mentions that *Mother Teresa* received an honorary Ph.D. from Cambridge, which also seems relevant, but the Wikipedia seed article only mentions in passing that she was awarded honorary degrees and it uses different terminology ("Universities in both the West and in India granted her honorary degrees."). In addition, the last nugget contains a coreference to the topic instead of mentioning it directly, which further hurts its topicality estimates.

The tendency to focus on aspects of the topic that are already well covered in the seed document is a weakness of the current approach for selecting relevant content. The source expansion system often expands on information that is in the seed corpus, but it is less likely to include information about aspects of a topic that are not mentioned at all. However, we believe that the SE approach can be extended to select more new information and to further improve its effectiveness for low-quality or outdated seeds. For example, one could augment the document representations of the seeds with content from similar documents in the seed corpus and high-scoring related text nuggets. The extended document representations could be used to compute topicality features such as likelihood ratios and maximal marginal relevance. Query expansion techniques and latent semantic analysis could alleviate vocabulary mismatches between text nuggets and seed content. We further discuss these extensions as possible directions for future research at the end of Chapter 9.

We also observed that the statistical model is biased against short text nuggets, particularly ones that do not contain an explicit reference to the topic. The following nuggets were both ranked low even though they provide useful information:

- **Topic:** Anne Frank
  Score: 0.0508
  Emigrates from Frankfurt/Main to Amsterdam

- **Topic:** Abraham Lincoln assassination
  Score: 0.0285
  Michael O'Laughlin, a Lincoln consipirator, manacled

The second text nugget also contains a typo ("consipirator") which affects the values of topicality features that compare its word distribution to the seed document. It is hard to avoid a general bias against short nuggets since in our dataset they are indeed rarely relevant. On the other hand, it does not seem to matter much whether short nuggets are selected because they often state common facts about the topic that are also covered in the seed and other related text.

In addition, we found that our approach for extracting text nuggets based on structural markup can cause problems if it splits relevant text passages into multiple nuggets. The individual text units may be ranked low if they are too short or if they share few key terms with the seed, and often they are no longer self-contained. Here are two examples:

- **Topic:** Iran-Iraq War
  Score: 0.0403
  Unknown, est. 1,000,000–2,000,000;

- **Topic:** Mother Teresa
  Score: 0.0073
  Agnes Gonxha Bojaxhi

In the first example the annotator labeled the text string "Casualties Unknown, est. 1,000,000–2,000,000" as relevant, but because of a line break in the source document it was split into two nuggets that are no longer self-contained. In the other example

a different annotator labeled "Name at birth: Agnes Gonxha Bojaxhi", which was divided in two parts for the same reason. The nugget formation rules could be refined to mitigate some of these issues, or adjacent text passages could be merged if the relevance estimate for the combined text is much higher.

Chapter 6

Application to Question Answering

The source expansion approach was evaluated on datasets from two question answering tasks: the Jeopardy! quiz show and the TREC QA track. In Section 6.1 we give an overview of the datasets, and in Section 6.2 we describe how text corpora used by Watson and the OpenEphyra QA system were expanded. The effect of source expansion on QA search results is analyzed in Section 6.3, and the impact on end-to-end accuracy is evaluated in Section 6.4. Finally, in Section 6.5 we take a closer look at how source expansion helps, and show that it improves QA performance both by adding semantic redundancy to the seed corpus and by increasing the coverage of the information sources.

6.1 Datasets

We used datasets comprising over 4,000 Jeopardy! questions and over 3,000 TREC factoid questions to evaluate the impact of source expansion on question answering results. In Section 3.2 we gave an overview of the two QA applications and analyzed differences between these tasks that affect the performance of our method. We now describe the question sets and the evaluation methodology used in our experiments in more detail.

6.1.1 Jeopardy!

Our source expansion algorithm was evaluated on questions from past Jeopardy! episodes that were collected by fans of the quiz show on J! Archive[1]. Most of these questions ask for factoid answers that can be extracted from unstructured document collections, but there are also puzzles, word plays and puns that require additional processing and special inference. A Jeopardy! episode consists of two rounds with up to 30 regular questions each (sometimes not all questions are revealed), followed by a Final Jeopardy! round with a single question. The Final Jeopardy! question is generally harder than regular questions, both for human players and the Watson QA system, which will be reflected in our evaluation results. In addition, the contestants

[1] `http://www.j-archive.com/`

Dataset	# Episodes	# Questions
Regular Jeopardy!	66	**3,508**
Final Jeopardy!	788	**788**

Table 6.1: Questions in Jeopardy! datasets.

are given 30 seconds to come up with the answer to the Final Jeopardy! question, instead of only 5 seconds for regular questions. Thus, the Final Jeopardy! questions leave more headroom for source expansion and we can use large expanded corpora even if this increases QA runtime noticeably. More than 6,000 Jeopardy! shows have aired over a period of almost 30 years, and the total number of questions available on J! Archive exceeds 200,000. In our source expansion experiments, we used datasets consisting of regular Jeopardy! questions and Final Jeopardy! questions from randomly selected past game shows. Questions with audio or visual clues are currently not supported by Watson and were excluded. In Table 6.1 we give an overview of the Jeopardy! datasets.

For each question that was revealed during a Jeopardy! game show, J! Archive provides a correct answer that was given by one of the contestants or the host. These answer keys were used to automatically evaluate search results and final answers returned by Watson. Jeopardy! questions are usually designed to have only a single correct answer, but this answer may change over time or there may be variations of an answer string that would still be considered acceptable. In the DeepQA project at IBM, human assessors who were not working on the QA system periodically updated the original answers from J! Archive with additional correct answer strings found in internal evaluations to more accurately estimate true system performance. We used these extended answer keys in all our experiments.

6.1.2 TREC

The source expansion approach was also evaluated on factoid questions from the TREC 8–15 evaluations, which are available on the NIST website[2]. The questions were partitioned into a dataset consisting of independent factoid questions from TREC 8–12 [Voorhees, 2003], and a dataset comprising series of questions about common topics from TREC 13–15 [Dang et al., 2006]. Examples of both types of questions were given in Section 3.2.1. QA systems usually perform worse on the second dataset because these questions are overall more difficult and can require more complex reasoning. In addition, the questions in TREC 13–15 often contain coreferences to their topics, previous questions or answers. These coreferences can be difficult to resolve accurately since not much context is provided. When evaluating the impact of our SE method on Watson's performance, we only used independent factoid questions from TREC 8–12 because the system currently does not support TREC question series. In experiments with OpenEphyra on TREC 13–15, we included the topic strings in

[2] `http://trec.nist.gov/data/qamain.html`

Dataset	# Factoid − # *NIL* Questions = # Used		
TREC 8–12	2,306	169	**2,137**
TREC 13–15	995	55	**940**
All	3,301	224	**3,077**

Table 6.2: Questions in TREC datasets.

the queries used to retrieve relevant content from the source corpora. This approach is relatively effective for resolving coreferences to the topics of question series, which occur frequently in TREC. A summary of the datasets is given in Table 6.2.

To judge QA search results and final answers automatically, we used answer keys that were compiled from correct answers found during the TREC evaluations and that are also available on the NIST homepage. *NIL* questions without known answers were removed from the datasets. TREC questions often have more than one acceptable answer, and the answer patterns only cover answers that were found in the reference corpora used in the evaluations. They are often incomplete, outdated or inaccurate, and evaluation results that are based on these patterns can significantly understate true system performance. When we evaluated end-to-end QA results (Section 6.4), assessors who were not involved in the development of the SE approach manually judged the top answers returned by the QA system and extended the answer keys with additional correct answers. The reported accuracies are based on these extended answer patterns, and are therefore unbiased estimates of true system performance. When measuring search performance (Section 6.3), we did not further extend the answer keys because it was impractical to manually judge tens or hundreds of search results for each question. However, since we used the same imperfect patterns in all search experiments, our comparative evaluation results are still valid.

6.2 Sources

We expanded local copies of two sources that are both useful for the Jeopardy! and TREC QA tasks: The online encyclopedia Wikipedia[3] and the online dictionary Wiktionary[4]. Wikipedia, in particular, has proven to be a valuable resource for a variety of natural language processing tasks [Bunescu et al., 2008, Medelyan et al., 2009] and has been used successfully in QA [Ahn et al., 2004, Kaisser, 2008]. The sources differ in two important 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 the topicality features that leverage the seed document body, *TopicRatioSeed*, *TFIDFSeed* and *CosineSim*, less effective. In Section 5.4 we confirmed that relevance estimation performance drops if less seed content is available. Web queries for common dictionary terms yield more noise, which we alleviated by adding the keyword "define" to

[3] `http://www.wikipedia.org/`

[4] `http://www.wiktionary.org/`

all queries and dropping search results that did not contain the topic in their title or URL. The web pages retrieved with this modified search approach are mostly online dictionary entries that provide additional definitions of the seed topic.

Our snapshots of Wikipedia and Wiktionary contain about 3.5 million and 600,000 documents, respectively. These collections only include English documents, and entries in Wiktionary were merged if they represented alternative spellings of the same term and cross-referenced one another. To reduce computational costs and to avoid adding noise, we focused on expanding seeds that we deemed most relevant for Jeopardy! and TREC. The 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. Popular topics also typically have high coverage on the Web, and thus we can retrieve large amounts of relevant content through web searches when expanding these topics. The measure of popularity was defined differently for each source based on our intuition and practical considerations. For Wikipedia, in which rich hyperlink information is available, we defined the popularity of an article as the number of references from other articles. Since Wiktionary contains only sparse hyperlink information, we estimated the popularity of a Wiktionary entry as 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 two QA tasks. A seed was considered relevant if its title was the answer to a TREC or Jeopardy! question. This is an approximation of relevance based on the potential impact of a seed if only document titles are retrieved. For the Jeopardy! challenge, we used a set of 32,476 questions selected randomly from episodes that are not part of our test sets to approximate relevance and plot relevance curves. After finalizing the rankings of the Wikipedia articles and Wiktionary entries, we performed the same analysis on the 3,077 questions with known answers in the TREC 8–15 datasets to judge the relevance of the seeds for the TREC task. The relevance curves for both Wikipedia and Wiktionary, shown in Figure 6.1 for Jeopardy! and in Figure 6.2 for TREC, have decreasing slopes. This indicates that popularity-based rankings indeed outperform random rankings of the seeds, which are illustrated by straight lines. Of course this analysis does not take into account whether a QA system could already answer a question without SE, and by focusing on popular seeds we may bias the expansion towards easy topics that were already well covered in the seed documents. However, we will show in Section 6.3 that the expansion of popular Wikipedia articles has the largest impact on QA search recall, and thus this approach for selecting relevant seeds is clearly effective in practice.

Based on these relevance curves, we chose to expanded the top 300,000 Wikipedia articles and the top 100,000 Wiktionary entries using content from web search results. In the merging step (cf. Section 4.4), text nuggets were dropped if more than 95% of their tokens were covered by higher scoring nuggets or the seed document. In addition, the size of the expanded pseudo-documents was restricted to at most 5 times the length of the seeds, and nuggets with relevance scores below 0.1 were removed. We analyzed a sample of pseudo-documents generated by the SE system

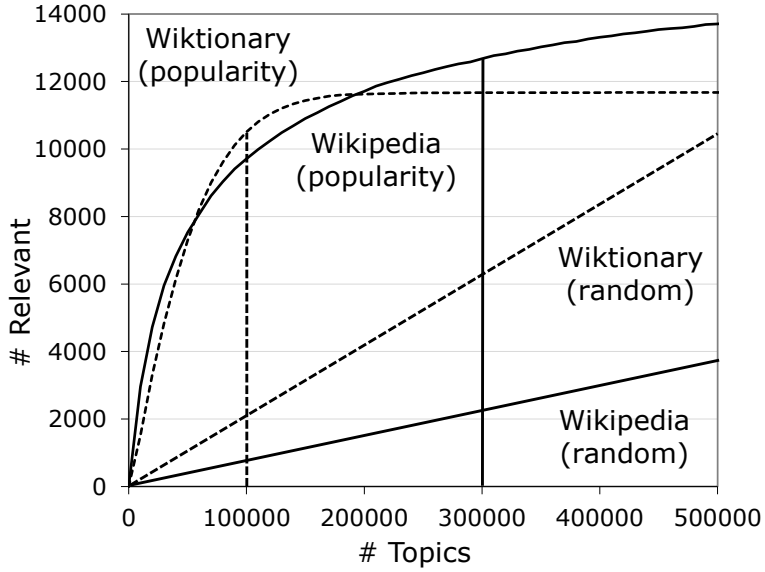


Figure 6.1: Relevance of Wikipedia and Wiktionary seed documents for the Jeopardy! task when ranked by popularity or randomly.

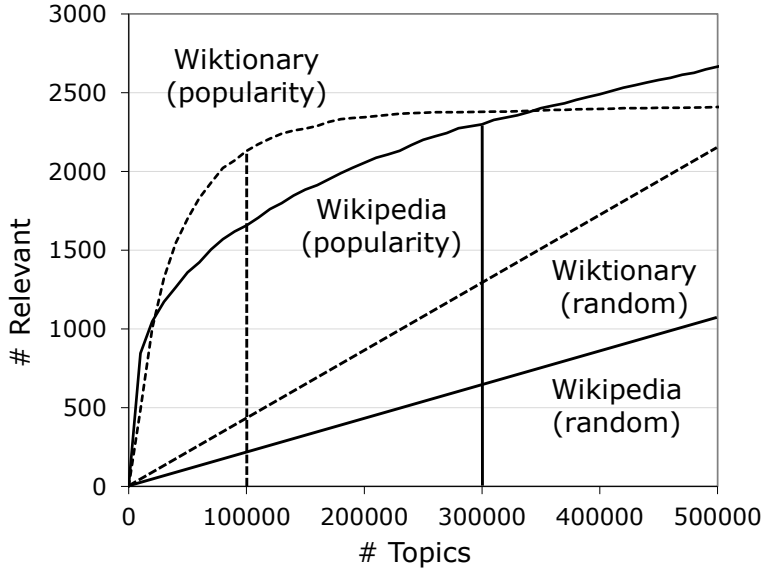


Figure 6.2: Relevance of Wikipedia and Wiktionary seed documents for the TREC QA task when ranked by popularity or randomly.

Source	# Documents	Size
Wikipedia	3,482,953	12.6 GB
Expanded Wikipedia	100,000	5.8 GB
	200,000	9.5 GB
	300,000	12.5 GB
Wiktionary	565,335	433 MB
Expanded Wiktionary	100,000	382 MB

Table 6.3: Sizes of Wikipedia, Wiktionary and expansions of these sources generated from web search results.

Source	# Documents	Size
World Book	18,913	54 MB
Expanded World Book	18,913	243 MB
Microsoft Encarta	32,714	77 MB
Expanded Microsoft Encarta	32,714	354 MB

Table 6.4: Sizes of additional encyclopedias and their expansions generated from web search results.

and chose parameter values that resulted in reasonable cutoff points. That is, with these parameters the final pseudo-documents contained a large number of relevant text 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 response times.

We extracted paragraph-length text nuggets from web pages based on HTML markup and scored them using a logistic regression model with features of adjacent instances, trained on all labeled data. Overall, the SE system processed about 40 million web pages, totaling about 2 TB of web data (including markup) and 4 billion text nuggets. The expansion of each seed document took around 15–45 seconds and most of this time was spent on the retrieval of web pages from different hosts. These numbers illustrate that a highly efficient and robust implementation is required. The sizes of the seed corpora and expanded sources are given in Table 6.3. 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.

Aside from Wikipedia and Wiktionary, we expanded two smaller document collections that are helpful for answering Jeopardy! and TREC questions: World Book and Microsoft Encarta. Table 6.4 shows statistics about the sizes of these sources and their expansions. Both sources are encyclopedias that are more similar to Wikipedia than to Wiktionary in terms of the distribution of their topics, and thus we used the same configuration of the SE system as for Wikipedia. Since these sources consist of relatively few articles and mostly cover popular topics, we expanded all documents

without performing popularity-based seed selection. In addition, we expanded two more sources, Encyclopedia Britannica and Columbia Encyclopedia, but QA search performance did not improve further. This is not surprising since these sources are comparatively small (123 MB and 55 MB respectively) and their topics are mostly covered in Wikipedia and the other encyclopedias. Thus we did not use their expansions in our experiments.

6.3 Search Experiments

We evaluated the impact of source expansion on search results using Jeopardy! and TREC datasets. In the following, we present experimental results for both Watson (Sections 6.3.1 and 6.3.2) and OpenEphyra (Sections 6.3.3 and 6.3.4). We also take a closer look at how our method affects the search rankings generated by the two QA systems in Section 6.3.5.

6.3.1 Experimental Setup using Watson

In the experiments with Watson, we used the following text corpora as baselines for source expansion: (1) Wikipedia, (2) Wiktionary, and (3) a large collection of existing sources that were manually identified to be relevant for Jeopardy! and TREC in an iterative error analysis performed by the Watson development team. The collection (subsequently referred to as *All Sources*) comprises 25.6 GB of text, including Wikipedia and the other encyclopedias in Section 6.2, dictionaries such as Wiktionary, thesauri, newswire sources such as a New York Times archive, literature and other sources of trivia knowledge [Chu-Carroll et al., 2012b]. It also includes the AQUAINT newswire corpus, which was the reference source in TREC 11–15 and contains the answers to all questions in these datasets (except *NIL* questions, which were not used in our experiments). We compare the Wikipedia and Wiktionary baselines to configurations that include the corresponding expanded sources with 300,000 and 100,000 pseudo-documents, respectively. The collection of all sources is compared to a corpus to which only expanded versions of Wikipedia, Wiktionary and the two small encyclopedias in Table 6.4 were added.

The sources were indexed and searched with both Indri[5] and Lucene[6], and the search results were pooled. Watson's retrieval component [Chu-Carroll et al., 2012a] was used to generate queries from Jeopardy! clues and TREC questions. The queries mostly consist of keywords and phrases combined with proximity operators and term weights. Two complementary search strategies were evaluated:

1. We retrieved *passages* of 20 tokens and subsequently adjusted them to align with sentence boundaries. The adjusted passages have different lengths, but the average value was similar in all experiments and thus the search performance of different configurations is directly comparable. Multiple searches were performed for each question and a total of 20 passages were retrieved:

[5] `http://www.lemurproject.org/indri/`
[6] `http://lucene.apache.org/`

- 5 passages were retrieved with Indri from all documents.
- 10 passages were retrieved with Indri only from documents whose titles appear in the Jeopardy! or TREC question. These documents are more likely to contain relevant text.
- 5 passages were retrieved with Lucene from all documents.

2. We fetched documents and used their *titles* as search 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*[7] (Answer: *vesper*). Source expansion helps for these searches by generating more complete and redundant descriptions of the entities that are more likely to match the query, not by adding new entities. For each question, we retrieved up to 55 document titles using the following search strategies:

   - 50 titles were retrieved with Indri from an index of long documents, which includes the encyclopedias and all expanded sources.
   - 5 titles were retrieved with Indri from an index of short documents, which includes mostly dictionary entries. Short documents were searched independently because the search engine has a bias towards longer documents, and thus few short documents would be retrieved if the documents were combined in one index. However, when we used only Wikipedia or Wiktionary as seed corpora, it was not necessary to perform two independent searches, and we instead retrieved 50 titles from a single index.

We used this combination of search strategies and retrieval systems because it was effective in experiments with Jeopardy! development data. Note that all documents and most passages are retrieved with Indri, but the Lucene passage searches still yield a small improvement in search recall. Lucene usually returns different passages because different query operators and retrieval models are used. For instance, Lucene can perform fuzzy matches based on the Levenshtein distance and thus can find relevant text passages that do not exactly match the question terms. While Indri uses a retrieval model that combines language modeling with inference networks, Lucene relies on a combination of vector space retrieval and a Boolean retrieval model.

In Table 6.5 we show queries generated by Watson to retrieve passages and documents from Indri and Lucene indices. The first example is a very short Jeopardy! clue from the category COMMON BONDS. The Indri queries consist only of individual keywords extracted from the clue, whereas the Lucene query also uses proximity operators to give more weight to passages that contain the terms next to each other (in any order). In Indri, the operator "#combine[passage20:6](...)" retrieves passages of 20 tokens, using increments of 6 tokens when traversing the indexed documents and evaluating candidate passages. The caret operator "^" boosts query terms in Lucene (or discounts them, if the factor is less than 1), and the tilde symbol "~" performs approximate matching, using the Levenshtein distance to measure the similarity between terms in the query and the index. The keywords in the Lucene query

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

Question (Jeopardy!)	COMMON BONDS: A picture, curtains, blood (*Answer:* drawn)
Query 1 (Indri Passage)	#combine[passage20:6](picture curtains blood)
Query 2 (Lucene Passage)	contents:pictur~ contents:curtain~ contents:blood~ (spanNear([contents:pictur~, contents:curtain~], 2, false) spanNear([contents:curtain~, contents:blood~], 2, false))∧0.4
Query 3 (Indri Document)	#combine(picture curtains blood)
Question (TREC)	What university did Thomas Jefferson found? (*Answer:* University of Virginia)
Query 1 (Indri Passage)	#weight[passage20:6](1.0 university 2.0 #weight(0.8 #combine(Thomas Jefferson) 0.1 #combine(#1(Thomas Jefferson)) 0.1 #combine(#od8(Thomas Jefferson))) 1.0 found)
Query 2 (Lucene Passage)	contents:univers~ contents:thoma~ contents:jefferson~ contents:found~ (spanNear([contents:thoma~, contents:jefferson~], 2, false) spanNear([contents:jefferson~, contents:found~], 2, false))∧0.4
Query 3 (Indri Document)	#combine(university #weight(0.8 #combine(Thomas Jefferson) 0.1 #combine(#1(Thomas Jefferson)) 0.1 #combine(#od8(Thomas Jefferson))) found)

Table 6.5: Examples of queries generated by Watson.

are stemmed, but Indri also performs stemming internally. The TREC question in the second example illustrates how named entities are represented in Indri queries. Search results are given higher scores if they contain the terms *Thomas* and *Jefferson* in this order with at most 7 terms in between them, and even more weight is given to passages and documents in which *Thomas Jefferson* appears as a continuous string. The Indri passage query also assigns a higher weight to the named entity than to the other query terms. Note that these questions were chosen as examples because the queries are relatively simple, and that they are not intended to fully illustrate Watson's query generation capabilities. For additional details, we refer the reader to Chu-Carroll and Fan [2011] and Chu-Carroll et al. [2012a].

Watson was developed for the Jeopardy! challenge, but has also been adapted to the TREC task [Ferrucci et al., 2010]. Thus we were able to use this QA system to evaluate search performance on both Jeopardy! and TREC questions. However, we only used TREC 8–12 in our experiments because Watson currently does not support the question series in more recent TREC datasets. Each text passage and document title retrieved from the sources was automatically matched against the answer keys,

and was judged as relevant if any subsequence of its tokens was a correct answer. Performance was measured in *search recall*, the percentage of questions with relevant results. Search recall is an important metric in QA because it constitutes an upper bound on end-to-end accuracy. Note that mean average precision (MAP) is not a suitable performance metric for this task since there is no predefined set of positive instances. Thus a configuration that retrieves a single relevant search result at a high rank would score higher than a setup that retrieves additional relevant results at low ranks. Likewise, the mean reciprocal rank (MRR) should be used with caution when evaluating search rankings because it is most sensitive to the top few search results and is hardly affected by relevant results at low ranks.

6.3.2 Watson Results and Analysis

Table 6.6 shows the impact of source expansion on Watson's search recall when retrieving only passages or titles and when combining these search results. The recall numbers are based on 20 text passages and up to 55 document titles per question (50 titles in experiments with only Wikipedia or Wiktionary and their expansions), retrieved with the queries described in the previous section. We also indicate for each combination of sources and test set the number of questions gained and lost through SE and the percentage gain. The performance on the regular Jeopardy! dataset is higher than on Final Jeopardy! data since the regular questions are generally easier to process and ask about less obscure facts. The performance numbers for TREC are lower than the regular Jeopardy! results because Watson's sources and its retrieval component are somewhat less effective for TREC questions. Most of the information sources were selected based on Jeopardy! data, and TREC questions usually contain less useful information about the answer that can be leveraged by the search component. In addition, the answer keys for TREC questions have lower coverage because these questions were not intended to be unambiguous and often have multiple correct answers. For instance, the dataset includes questions such as *Name a civil war battlefield.* (TREC 9, Question 543) or *How far is it from Denver to Aspen?* (TREC 10, Question 894), which have many acceptable answers that are not all covered by the regular expressions used in our automated evaluations.

Statistical source expansion consistently improves search recall on all datasets, independently of the baseline or search strategy used. We performed one-sided sign tests and found that all gains are statistically significant with $p < .001$. Even if a seed corpus with reasonable coverage for a QA task exists, such as Wikipedia for Jeopardy! and TREC, the expanded corpora improve performance substantially. If a source with lower coverage is expanded, such as Wiktionary, very large gains are possible. Compared to the strongest baseline comprising a collection of sources that were specifically selected for the two QA tasks, our method 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 quite large even though of all sources used in the baseline only Wikipedia, Wiktionary and two small encyclopedias (about half of the collection) were expanded. It can further be seen that relatively few questions are hurt by our method. For example, 171 regular Jeopardy! questions are gained

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***

Table 6.6: Search recall of Watson on Jeopardy! and TREC questions when using sources that were expanded with web search results. 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.

	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 6.7: Search recall of Watson on Jeopardy! and TREC questions when expanding increasing numbers of Wikipedia seed articles using web search results.

but only 44 out of 3,508 questions in the dataset are lost when adding expanded content to the manually acquired corpora. This distinguishes the source expansion approach from most query expansion techniques, which typically hurt performance more frequently [Collins-Thompson and Callan, 2007]. Our approach adds noise to the seed corpus in addition to relevant data, but this is less likely to cause a search to fail than irrelevant terms added to the query.

The recall curves in Figure 6.3 illustrate that source expansion improves search performance independently of the hit list length for both Jeopardy! and TREC questions. In these experiments we used only Indri and performed only one passage search and one title search per question to obtain a single hit list for each strategy. Passages were retrieved from all available sources, and titles were fetched from an index of long documents that includes all encyclopedias. Both collections were augmented with expansions of Wikipedia, Wiktionary and the other encyclopedias. The gains in passage recall are larger for Jeopardy! than for TREC questions because there is more headroom in Jeopardy! search performance. The index used for passage searches includes the reference corpus used since TREC 11, and passage recall on the TREC question set 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 recall curves also show that when adding the expanded sources, fewer passages or titles 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 by downstream components in the QA pipeline.

Table 6.7 shows the impact of SE when applied to increasing 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. As seeds with lower popularity estimates are expanded, more noise and less useful information is added to the sources, and the improvements in recall diminish. On TREC, performance even degrades (though not significantly) if more than 200,000 seeds are expanded. Of course one should also avoid expanding irrelevant seeds because the response time of a QA system increases with the size of the corpus used. The searches for relevant text in the knowledge sources are among the most expensive processing steps in Watson's pipeline, and the computational costs of these searches are linear in the index size.

We also evaluated how different strategies for estimating the relevance of text nuggets in the scoring phase of the source expansion system (Section 4.3) affect search

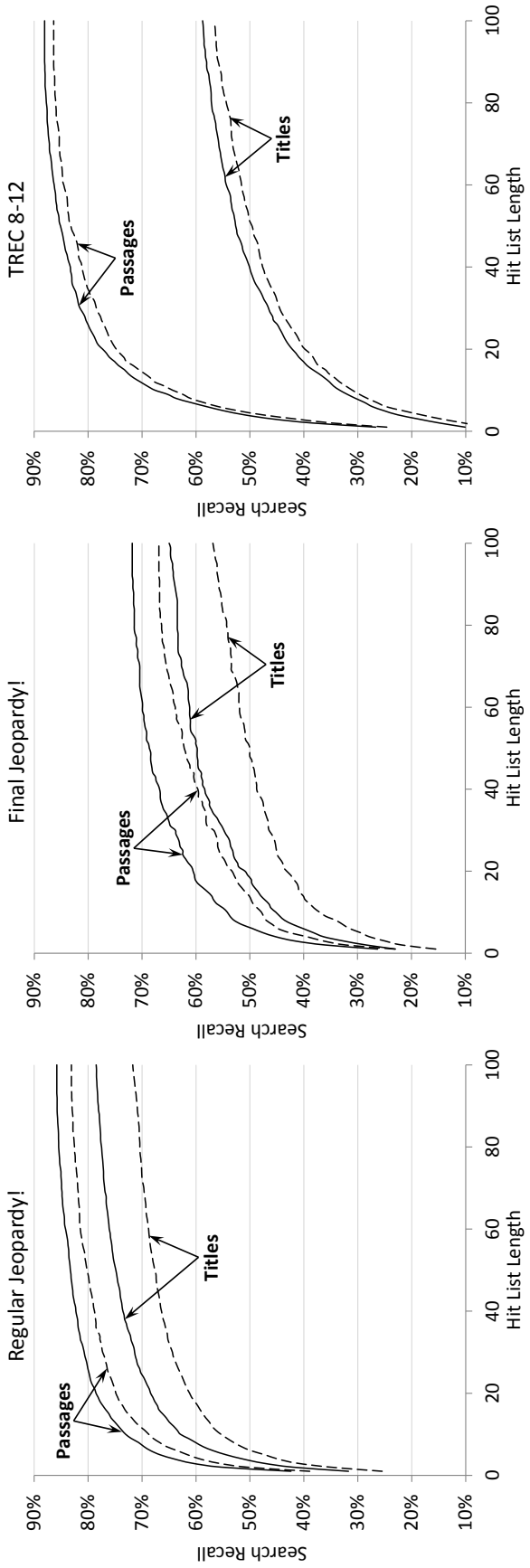


Figure 6.3: Search recall of Watson on Jeopardy! and TREC questions as a function of the hit list length. Dashed lines: all sources without expansion; solid lines: all sources with expansion of Wikipedia, Wiktionary and other small encyclopedias using web search results.

	Regular J!	**Final J!**	**TREC 8–12**
No Expansion	81.33%	63.32%	76.74%
Random	83.98%	66.37%	78.90%
Round Robin	84.52%	69.04%	80.35%
Search Rank	85.58%	68.91%	79.64%
Cosine Sim	86.15%	70.30%	81.09%
LR Adjacent	**86.23%**	**72.21%**	**82.17%**

Table 6.8: Search recall of Watson on Jeopardy! and TREC questions when expanding Wikipedia with web search results using different relevance estimation strategies.

	Regular J!	**Final J!**	**TREC 8–12**
No Expansion	30.39%	13.32%	29.15%
Random	45.27%	18.65%	44.78%
Round Robin	39.91%	18.53%	39.82%
Search Rank	46.12%	24.24%	47.40%
Cosine Sim	46.15%	23.60%	47.17%
LR Adjacent	**51.20%**	**27.79%**	**52.46%**

Table 6.9: Search recall of Watson on Jeopardy! and TREC questions when expanding Wiktionary with web search results using different relevance estimation strategies.

results. Wikipedia and Wiktionary were used as seed corpora, and performance was evaluated on both Jeopardy! and TREC datasets. We compared the search recall of Watson when using the seed corpora without SE to expanded sources generated with different relevance estimation baselines and the logistic regression model with features of adjacent instances (*LR Adjacent*). As baselines, we used random rankings of text nuggets (*Random*), an approach that selects nuggets from the top of the retrieved web pages (*Round Robin*), Yahoo! search rankings (*Search Rank*), and rankings of nuggets by their cosine similarities to the seed documents (*Cosine Sim*). These methods were described in more detail in Section 5.2. Note that we adjusted the relevance threshold and the maximum length of the expanded documents in the merging step of the SE pipeline (Section 4.4) to generate expanded sources of similar sizes, and thus we can directly compare the impact of different strategies. The results are summarized in Table 6.8 for Wikipedia and in Table 6.9 for Wiktionary.

In general, relevance estimation methods that were more effective for ranking text nuggets in Chapter 5 also have higher search recall. Thus we can select an effective method based on intrinsic evaluation results and, with few exceptions, can expect it to perform well when applied to the question answering task. It can also be seen that search recall improves even if the seed documents are expanded with content selected randomly from the retrieved web pages (*Random* baseline). This is most apparent if a small seed corpus is used that does not contain the answers to many of the Jeopardy! and TREC questions, such as Wiktionary. However, one should keep in mind that these are not arbitrary web pages but that they were selected by the Yahoo! search

engine among billions of candidates, and they are related to the topics of the seeds that are most popular and useful for our QA tasks. Furthermore, the gains in search recall are clearly larger if relevant text is selected based on a more informed relevance estimation approach.

The performance of the *Round Robin* baseline is relatively low for Wiktionary because the retrieved web pages are mostly entries in online dictionaries that often start with irrelevant text such as headings, menu bars and advertisements. Because Wiktionary seeds are usually very short, the threshold that restricts the size of the pseudo-documents relative to the seed length can be exceeded before the relevant sections of the web pages are reached. On the other hand, the baseline that uses search engine rankings (*Search Rank*) is relatively ineffective when expanding Wikipedia because the web pages retrieved by Yahoo! for Wikipedia seeds can be very long, and sometimes the search results at the highest ranks do not contain useful content. In that case, the allotted length for the expanded document is mostly taken up by noise that is not relevant for QA. The *Cosine Sim* baseline is less effective for Wiktionary than for Wikipedia because it exclusively relies on the seed documents for estimating the relevance of text nuggets. Wiktionary entries are not only much shorter on average than Wikipedia articles, but they also contain more text that is not directly relevant, such as examples of how a dictionary term is used in a sentence, its pronunciation and translations to other languages. On the Final Jeopardy! and TREC datasets, the approach performs worse than search engine rankings, which do not depend on the content of the seeds since we only used the seed document titles as queries.

The relevance model *LR Adjacent* consistently outperforms all baselines, including *Cosine Sim*, independently of the seed corpora and test set. The gains over *Cosine Sim* are quite small when expanding Wikipedia, but very noticeable for Wiktionary. The reason for the larger performance difference in the experiments with Wiktionary is that the statistical model does not only rely on the seeds when scoring related text but it also uses a variety of other features. Some of these features leverage the search ranking and metadata from the search engine, others are based on surface characteristics of the retrieved text. Thus, even if the seed documents are short or have low quality, the model still works relatively well. The downside of the statistical approach is that its implementation is more complex and labeled training data is needed to fit the model. Based on these findings, we recommend using a statistical model if enough data can be annotated. Otherwise, the baseline *Cosine Sim* can be a reasonable alternative, but one should keep in mind that this method does require high-quality seeds to be effective. These conclusions are consistent with the results of experiments using artificially degraded seed documents in Section 5.4.

6.3.3 Experimental Setup using OpenEphyra

We also evaluated the impact of source expansion on the search performance of the open-source QA system OpenEphyra[8]. The following text corpora were used in these search experiments:

[8] `http://sourceforge.net/projects/openephyra/`

1. Seed corpora without source expansion (*Wikipedia*, *Wiktionary*, and the combination *Wikipedia + Wiktionary*).

2. Seed corpora expanded using search rankings generated by the Yahoo! search engine (*Search Rank SE*). This is the most effective relevance ranking strategy in Chapter 5 that does not rely on the seed documents to model topicality.

3. Seed corpora expanded with a statistical relevance model that leverages the content of the seeds to estimate the relevance of related text nuggets (*Statistical SE*). We used the model *LR Adjacent*, which includes features of adjacent nuggets to capture dependencies between nearby text. This is the best-performing relevance model in Chapter 5.

When comparing different source expansion approaches, we built corpora of equal size by adjusting the relevance threshold in the merging phase of the SE system (Section 4.4) to ensure comparable results.

All seed corpora and expanded sources were indexed and searched with Indri. We did not use Lucene because OpenEphyra currently does not provide an interface for this retrieval system, and the contribution of the Lucene searches to overall search recall in experiments with Watson is relatively small. Queries consisting of keywords and phrases were generated with OpenEphyra's question analysis component. Again we used two complementary search strategies:

1. We searched for *passages* comprising 50 tokens and aligned them with sentence boundaries in two steps: first we cut off sentences that did not contain any question terms, then we extended the remaining text to the nearest boundaries. The passage length was optimized to achieve high search recall for TREC questions when using only the seed corpora without expansion. Interestingly, we found it more effective to use longer passages than Watson (with 50 instead of 20 tokens), even if the number of search results is reduced accordingly and the same total amount of text is retrieved. This may be because we tuned the passage length using smaller sources, which are less likely to contain the question terms and answers in close proximity to one another. Note that the retrieval of fixed length token windows is not ideal for the expanded pseudo-documents because the windows are not necessarily aligned with nugget boundaries but can span several independent text nuggets. This was unavoidable to enable comparisons between expanded sources and baseline corpora without nugget annotations, and in experiments with Watson it did not affect search results much because the retrieved passages rarely covered more than one nugget. However, the negative effect of passages crossing nugget boundaries was more noticeable when retrieving longer passages using OpenEphyra. We tried reducing the passage length to 20 tokens and found that the impact of SE increased, but search recall with and without SE was lower even if longer hit lists were used.

2. We retrieved documents and extracted the *titles* as search results. Again, these searches are most useful for questions that provide a definition of an entity

Question (TREC)	What is the average body temperature? (*Answer:* 98.6 °F)
Query 1 (Indri Passage)	#combine[passage50:25](average body temperature #1(body temperature))
Query 2 (Indri Document)	#combine(average body temperature #1(body temperature))
Question (TREC)	What is the nickname for the national New Zealand basketball team? (*Answer:* Tall Blacks)
Query 1 (Indri Passage)	#combine[passage50:25](nickname national New Zealand #1(New Zealand) basketball team #1(basketball team))
Query 2 (Indri Document)	#combine(nickname national New Zealand #1(New Zealand) basketball team #1(basketball team))

Table 6.10: Examples of queries generated by OpenEphyra.

and ask for its name, such as *What is the study of ants called?* (TREC 12, Question 2086, Answer: *myrmecology*).

Table 6.10 gives examples of Indri passage queries and document queries generated for TREC questions by OpenEphyra. The first question contains the compound noun *body temperature*, which is recognized through lookups in WordNet [Fellbaum, 1998] and search results that contain it as an exact phrase are given higher weights. In the second example, the country name *New Zealand* is detected by a named entity recognizer for locations (though it could also be found in WordNet) and is added as a phrase to the Indri queries. OpenEphyra also supports query expansion using synonyms and other relations in WordNet, but the algorithm was disabled in our experiments because it does not consistently improve search results. In contrast to Watson, OpenEphyra does not use any proximity operators other than exact phrases and it does not assign weights to the query terms, though named entities and compound nouns are given more weight by including them both as a phrase and as individual keywords in the query. The passage queries are identical to those used for document searches, except that the operator "#combine[passage50:25](...)" is used to retrieve fixed-size windows of 50 tokens, evaluated at increments of 25 tokens.

While Watson supports both Jeopardy! and TREC questions, the question analysis component of OpenEphyra was developed specifically for TREC-style factoid questions. It has not been adapted to Jeopardy! clues, which are on average much longer, often contain unnecessary or even misleading information, and are given in the form of statements rather than in interrogative form. Thus we used OpenEphyra only for experiments with TREC questions, but we evaluated its performance both on independent factoid questions in TREC 8–12 and question series about common topics in TREC 13–15. Again, the retrieved passages and document titles were judged automatically by determining whether any token sequence matches the answer patterns, and performance was evaluated in terms of *search recall*. We also report the *average number of relevant results*, which is a measure of the redundancy in the search

results. Multiple relevant results for a single question can facilitate answer extraction and scoring.

6.3.4 OpenEphyra Results and Analysis

Table 6.11 shows OpenEphyra's search recall and the average number of relevant search results for the seed corpora, SE using search engine rankings and statistical SE when using hit lists of 20 passages and 10 titles. Both expansion methods consistently improve search performance, independently of the sources, search strategy and dataset. Similarly to the experiments with Watson, the largest improvements are realized if Wiktionary is expanded, and the gains are smaller if larger seed corpora with higher coverage are used. When expanding both Wikipedia and Wiktionary using the statistical method, passage recall increases by 4.0% and title recall improves by 13.7% on the full dataset that includes factoid questions from TREC 8–15. The recall gains on this combined dataset are statistically significant ($p < .0001$) based on a one-sided sign test. Search recall also improves on every individual TREC dataset, which is not shown in these aggregate results. The expansion method that uses search engine rankings has lower performance than statistical SE, which confirms that the relevance model that leverages the seed content to estimate the topicality of retrieved text is effective. The difference in search performance is most apparent when expanding Wikipedia because the statistical model was trained on text nuggets retrieved for Wikipedia seeds, and because the topicality features used in the model are more effective for longer seeds.

The performance on TREC 8–12 is higher than on TREC 13–15, which was to be expected because the factoid questions in more recent datasets are generally harder and sometimes require more sophisticated reasoning than earlier questions. In addition, the question series in recent evaluations contain coreferences, which were not always resolved accurately (see Section 6.1.2). Search performance is particularly low on TREC 13–15 if passages or titles are retrieved from Wiktionary. This is also not surprising since the question series frequently ask about named entities such as people, organizations and events, which are often not dictionary entries. On the other hand, older TREC questions such as *What is the fear of lightning called?* (TREC 11, Question 1448, Answer: *astraphobia*) or *What name is given to the science of map-making?* (TREC 11, Question 1580, Answer: *cartography*) are covered in Wiktionary.

Compared to the results for Watson on TREC 8–12 in Section 6.3.2, the search recall in experiments with OpenEphyra with and without source expansion is much lower. Watson generates more effective Indri queries that include term weights, various types of proximity operators and synonyms, whereas OpenEphyra's queries consist only of unweighted keywords and phrases. Also, OpenEphyra sometimes includes irrelevant question terms in the queries, and the algorithm for extracting phrases based on named entity recognizers and dictionary lookups has comparatively low precision and recall. Moreover, Watson uses a combination of Indri and Lucene searches, and it performs multiple searches against different subsets of its sources. This mix of retrieval systems and search strategies yields higher recall than the single-strategy approach used in the experiments with OpenEphyra. In addition, there are a number

Sources	**TREC 8–12**				**TREC 13–15**				**All**			
	Passages@20		Titles@10		Passages@20		Titles@10		Passages@20		Titles@10	
	Recall	Rel	Recall	Rel	Recall	Rel	Recall	Rel	Recall	Rel	Recall	Rel
Wikipedia	59.57%	2.69	28.64%	0.42	52.66%	2.06	18.51%	0.28	57.46%	2.50	25.54%	0.38
Search Rank SE	60.79%	2.75	30.18%	0.47	54.36%	2.14	19.26%	0.28	58.82%	2.57	26.84%	0.41
Statistical SE	62.19%	2.88	33.27%	0.56	54.47%	2.20	21.06%	0.31	59.83%	2.67	29.54%	0.49
	($p < .0001$)		($p < .0001$)		($p = .0020$)		($p = .0040$)		($p < .0001$)		($p < .0001$)	
Wiktionary	26.91%	0.64	9.78%	0.12	15.53%	0.48	1.60%	0.03	23.43%	0.59	7.28%	0.09
Search Rank SE	48.62%	1.59	16.14%	0.20	31.70%	0.91	4.89%	0.07	43.45%	1.38	12.71%	0.16
Statistical SE	49.04%	1.80	17.22%	0.21	32.23%	0.94	5.32%	0.08	43.91%	1.54	13.58%	0.17
	($p < .0001$)		($p < .0001$)		($p < .0001$)		($p < .0001$)		($p < .0001$)		($p < .0001$)	
Wikipedia + Wiktionary	60.32%	2.74	28.97%	0.43	52.77%	2.08	18.72%	0.29	58.01%	2.54	25.84%	0.38
Search Rank SE	61.82%	2.80	30.65%	0.48	55.11%	2.20	19.47%	0.28	59.77%	2.62	27.23%	0.42
Statistical SE	62.94%	2.93	33.18%	0.57	54.47%	2.22	20.74%	0.31	60.35%	2.71	29.38%	0.49
	($p < .0001$)		($p < .0001$)		($p = .0147$)		($p = .0197$)		($p < .0001$)		($p < .0001$)	

Table 6.11: Search recall of OpenEphyra on TREC questions when using sources that were expanded with web search results. We show the impact of source expansion based on search engine rankings (*Search Rank SE*) and a statistical relevance model (*Statistical SE*). For each setup, we also indicate the average number of relevant search results. The p-values are based on a one-sided sign test and compare statistical SE to the seed corpora without expansion.

Sources	**Passages@20**		**Titles@10**	
	Recall	**Rel**	**Recall**	**Rel**
Wikipedia	57.46%	2.50	25.54%	0.38
Top 100,000	59.08%	2.59	29.02%	0.45
Top 200,000	59.70%	2.64	29.31%	0.48
Top 300,000	**59.83%**	**2.67**	**29.54%**	**0.49**

Table 6.12: Search recall of OpenEphyra on TREC questions when expanding increasing numbers of Wikipedia seed articles using web search results.

of more subtle differences between the search components of the two QA systems. For example, the sentence detection algorithm used by Watson to align the retrieved token windows with sentence boundaries appears to be more reliable, and Watson deploys a modified version of Indri whose passage scoring model has been customized to improve QA search results. Also note that Watson retrieved at least 50 document titles per question (55 titles when searching indices of long and short documents separately), whereas in the experiments with OpenEphyra on TREC datasets we reduced the hit list length to 10 since the title search approach is less effective for TREC questions. This is because there are few questions that provide a definition or properties of an entity and ask for its name.

The recall curves in Figure 6.4 illustrate that statistical SE also consistently improves OpenEphyra's search recall on TREC 8–15 independently of the hit list length. We show results for passage and title searches using Wikipedia and Wiktionary as seed corpora. When retrieving titles, SE does not add new candidates that could not be found in the seed corpus, but it helps return relevant titles at higher ranks. This explains why the increase in title recall for Wikipedia is largest if relatively few titles are retrieved. If longer hit lists are used, the order of the titles in the ranking has less impact on search recall. The improved ranking performance is reflected by the mean reciprocal rank of the top 100 titles (MRR@100), which increases from 0.119 to 0.151 when expanding Wikipedia, and from 0.035 to 0.071 when applying our method to Wiktionary. More effective rankings can improve answer selection performance, and thus end-to-end QA accuracy, if the search rank is used as a feature. In addition, if relevant search results are ranked higher, shorter hit lists can be used with little loss in recall to reduce the number of incorrect candidate answers and the computational costs for scoring the candidates.

Table 6.12 shows OpenEphyra's search performance on TREC 8–15 if varying numbers of Wikipedia seed articles are expanded. Similarly to the results for Watson, the 100,000 most popular seed documents yield the largest gains and the final 100,000 seeds have little impact on search results. This applies to search recall as well as the number of relevant results, and to both passage and title searches. Thus our method for popularity-based seed selection appears to be equally effective for OpenEphyra.

For about 43% of the factoid questions in TREC 8–15 OpenEphyra generates queries that only consist of individual keywords, whereas 57% of the questions contain compound nouns and named entities that are used as phrases in the Indri queries.

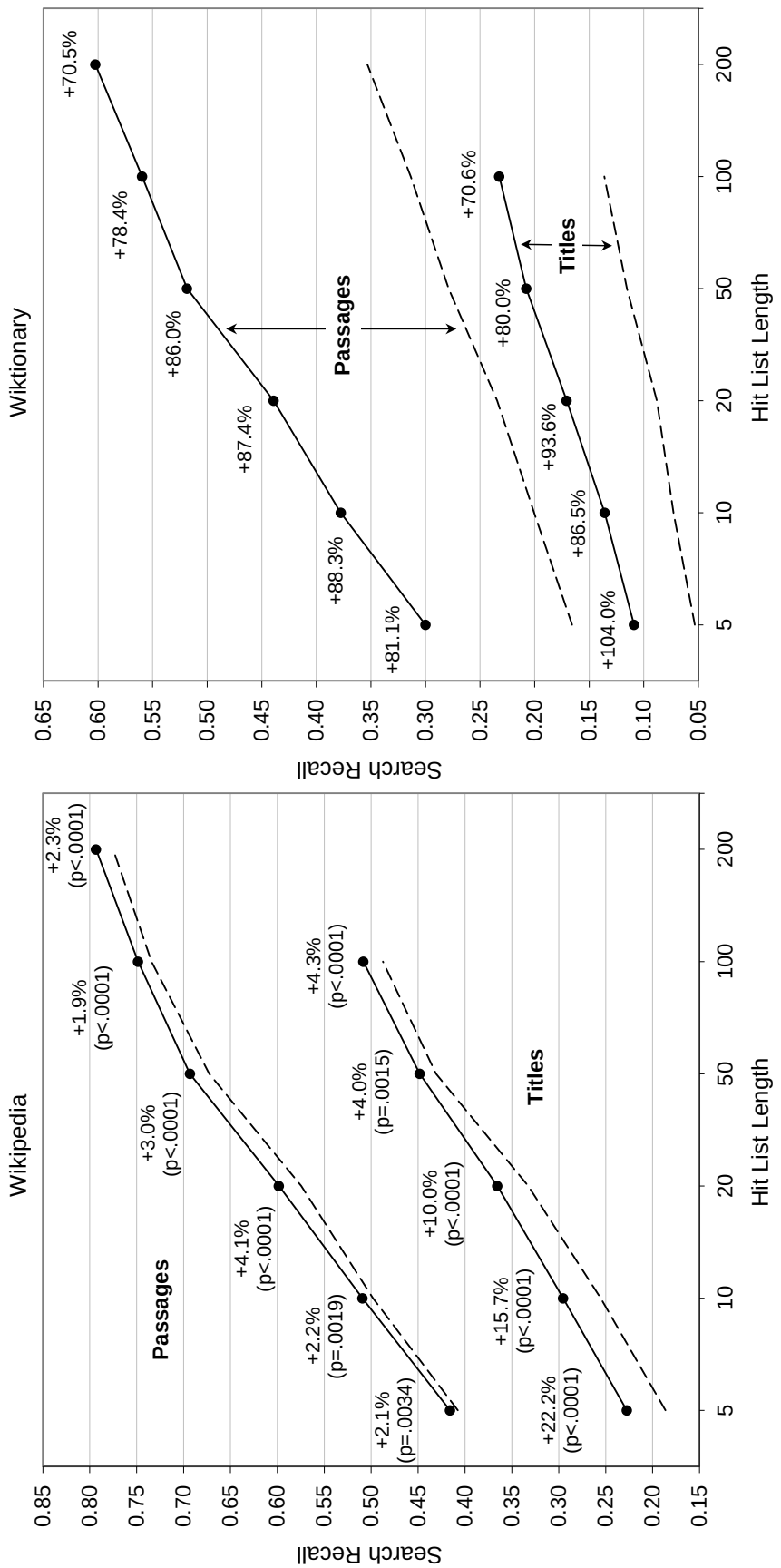


Figure 6.4: Search recall of OpenEphyra on TREC questions as a function of the hit list length when Wikipedia *(left)* or Wiktionary *(right)* are used as sources. Dashed lines: seed corpora without expansion; solid lines: statistical source expansion using web search results. All p-values for Wiktionary are below .0001.

Sources	Passages@20		Titles@10	
	Keywords	**Phrases**	**Keywords**	**Phrases**
Wikipedia	53.89%	60.11%	23.17%	27.31%
Statistical SE	55.41%	63.12%	26.45%	31.84%
	(+2.8%)	*(+5.0%)*	*(+14.2%)*	*(+16.6%)*

Table 6.13: Search recall of OpenEphyra on two subsets of TREC 8–15: (1) questions from which queries consisting of individual keywords were generated, and (2) questions for which queries with at least one phrase were constructed.

In Table 6.13 we show the search recall of OpenEphyra and the impact of statistical source expansion on these two subsets of TREC 8–15 for both passage searches and title searches. Queries containing at least one phrase have higher recall than keyword queries, and they also benefit more from expanded sources. This comes to no surprise since phrase queries are often more specific and are less affected by noise added through source expansion in addition to useful information. Thus, to maximize the impact of our method, one should consider using more constrained queries that are resistant to noise.

6.3.5 Robustness of Search

So far we have seen that statistical source expansion yields consistent gains in search recall while hurting relatively few questions. This sets our method apart from query expansion techniques, which typically have much higher variance [Collins-Thompson and Callan, 2007]. Now we take a closer look at how search rankings generated by Watson and OpenEphyra are affected by source expansion, following the evaluation methodology proposed by Collins-Thompson [2008]. For each question in the Jeopardy! and TREC datasets, we retrieved 100 passages and 100 document titles from baseline corpora and expanded sources generated through statistical SE. In experiments with Watson, we used the collection of manually acquired sources as a baseline and expanded Wikipedia, Wiktionary and the two small encyclopedias. The results for OpenEphyra are based on Wikipedia with and without source expansion. We then examined how SE affects the rank of the first relevant passage or title. This analysis gives us insights into how much our method helped or hurt for each question, and into the robustness of the search rankings to source expansion. The positions of the first relevant passage and title are important because the search ranks can be used as features for answer scoring. If relevant results are generally retrieved at high ranks, answer scoring accuracy can be improved by giving more weight to candidates that were extracted from the top results.

We generated histograms that illustrate the distribution of the change in retrieval rank of the first relevant search result if expanded sources are added. The histograms are given in Figure 6.5 for Watson on both Jeopardy! and TREC datasets, and in Figure 6.6 for OpenEphyra on TREC data. On the horizontal axis we show different intervals for the change in retrieval rank, and the two bars over each range indicate

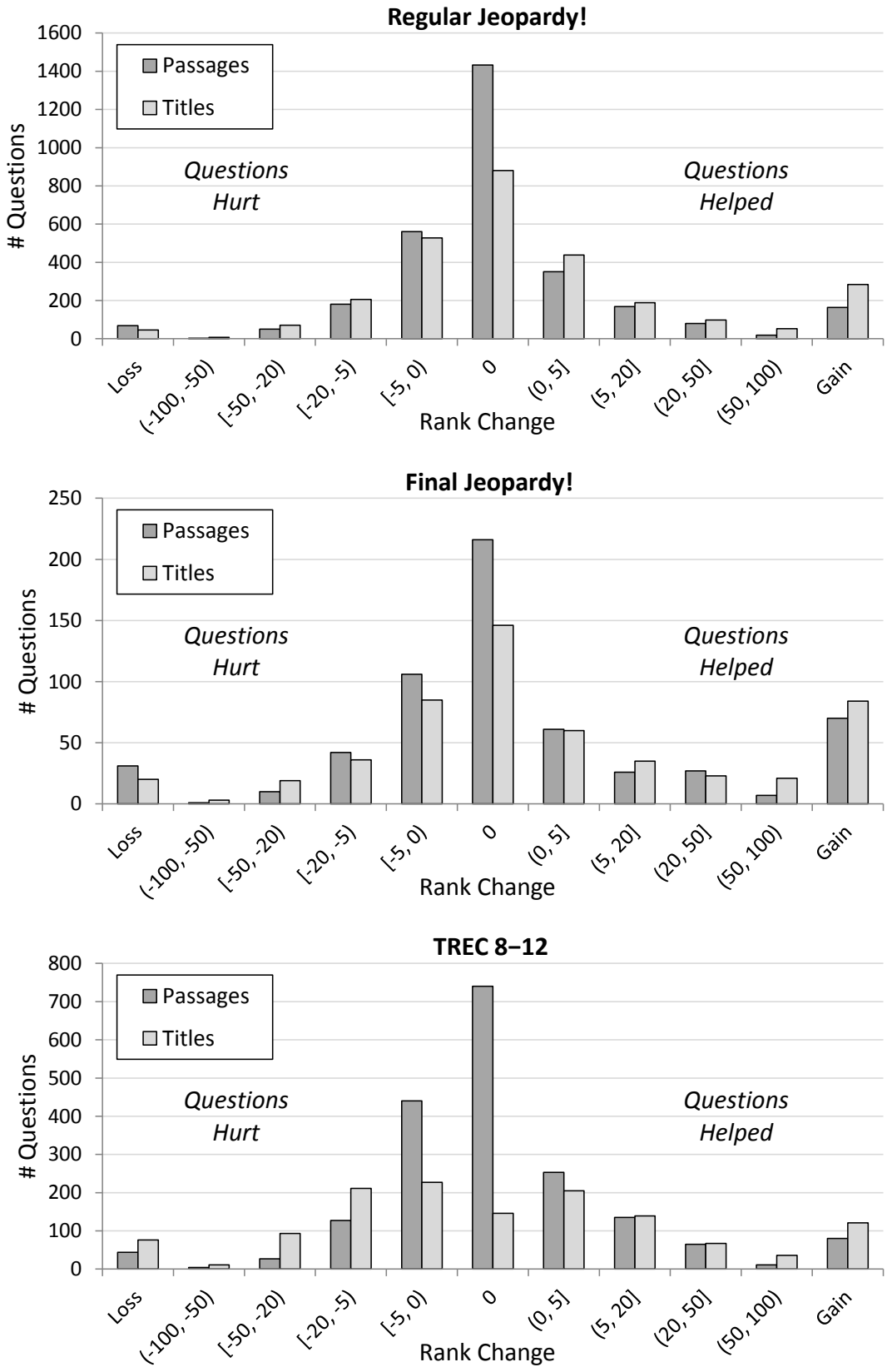


Figure 6.5: Robustness of Watson's search results to source expansion.

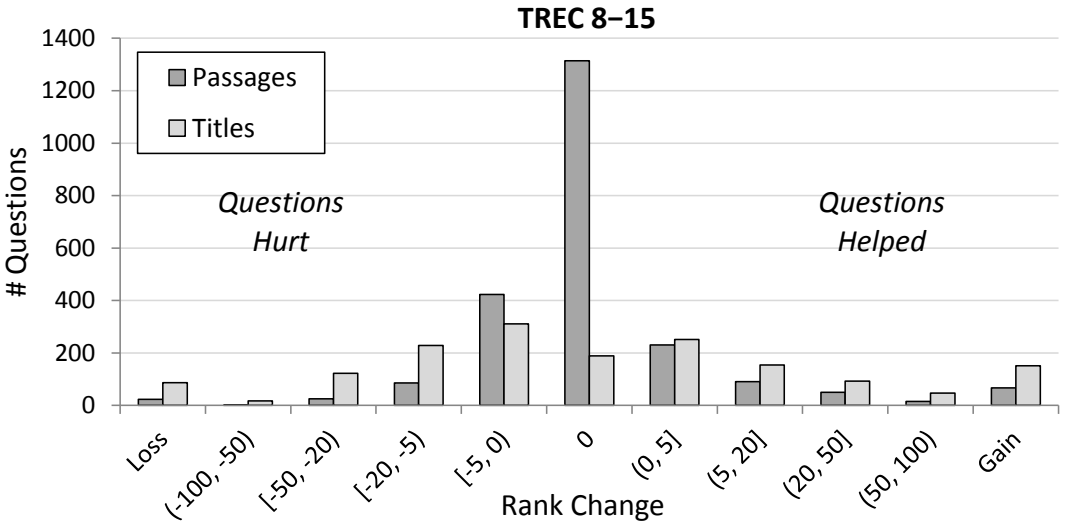


Figure 6.6: Robustness of OpenEphyra's search results to source expansion.

the number of questions for which the rank change of the first relevant passage or title falls within that bracket. For instance, in experiments with Watson on regular Jeopardy! questions, the rank of the top relevant text passage was not affected for 1,433 questions, and the top relevant title improved by up to 5 ranks for 438 questions. The number of questions for which relevant passages or titles were found only with SE is illustrated by the bars on the right-hand side, and the number of questions lost in the search phase due to SE is given by the bars on the left-hand side. Note that these numbers are different from the results presented earlier because we used much longer hit lists in this analysis. Questions for which no relevant results could be retrieved in both the baseline experiment and the configuration with SE were omitted.

It can be seen that the rank of the first relevant passage often remained unchanged. For example, out of the 2,846 regular Jeopardy! questions for which relevant passages were retrieved with and without source expansion, the first relevant result had the same rank for 1,433 questions. The title search results are less robust because Indri more often retrieves expanded documents, which are much longer on average than the documents in the original corpus. Here the rank of the first relevant result did not change for 880 of the 2,470 questions for which relevant titles were found in both experiments. The results for Final Jeopardy! are similar, but source expansion overall had a larger impact on search performance because there is more headroom for improvements, and the variance of the rank changes is higher. In experiments with TREC data, the title rankings are much less stable because the document searches are relatively ineffective independently of the sources used. This strategy is mainly intended for questions seeking the name of an entity that matches a given description, which are less common in TREC datasets. For both types of searches, small differences in the retrieval ranks are more frequent than large changes. Note that we used bins of increasing sizes in the histograms since otherwise most of the bins for large gains or losses would be empty. Collins-Thompson [2008] used different datasets when evaluating the robustness of query expansion techniques and the histograms are based

on the change in average precision, but nonetheless it seems that source expansion has lower variance.

When source expansion hurts QA search performance it often only results in a slightly worse ranking, but when it helps large changes are relatively more frequent. For instance, the rank of the first relevant passage retrieved by Watson for regular Jeopardy! questions was improved on average by 10.4 positions if SE helped, but it only worsened by 5.8 ranks if the expansion hurt. For title searches, the average gain was 12.6 ranks and the average loss was 7.3 ranks. When considering all questions for which the rank of the first relevant search result changed in either direction, source expansion on average yielded an improvement of 1.3 ranks for passages and a gain of 2.5 for titles. The results are similar for other question sets and for OpenEphyra. Thus source expansion not only increases search recall but it also slightly improves the rankings for questions whose answers were already found in the seed corpus. This is not an obvious result because the expanded documents contain noise in addition to useful content, and sometimes low-quality text is retrieved at high ranks because it contains many query terms. This analysis shows that the impact of relevant content in the expanded sources outweighs the effect of noise even for those questions that have good coverage in the original sources.

6.4 End-to-End Experiments

The improvements in search performance are promising, but what ultimately matters is final QA accuracy after extracting and ranking the candidate answers. We now show that statistical source expansion also significantly improves end-to-end QA results. In fact, the gains can even be larger than the improvements in search recall if the expanded sources are used to retrieve additional supporting evidence and candidate answers are scored effectively.

6.4.1 Experimental Setup

We again used Watson to evaluate the impact of SE on end-to-end QA performance. OpenEphyra was not a suitable framework for these experiments because its answer scoring component was designed to leverage redundancy in web search results. When using smaller local sources, the search results may only mention a correct answer once, and OpenEphyra often cannot distinguish it from incorrect answers of the same type. Thus the final QA accuracy tends to be low and has high variance. Watson, on the other hand, uses a comprehensive statistical framework for answer scoring that combines hundreds of features that leverage various sources of evidence ranging from unstructured text to ontologies and triple stores. These features take into account whether a candidate is of the correct type implied by the question, the correlation of the candidate with key terms in the question, its popularity, the reliability of its source, the lexical and semantic similarity of the question and passages that contain the answer, and much more. This statistical answer selection approach is more effective and depends much less on redundant search results.

In end-to-end experiments, Watson leverages the expanded sources both to generate candidate answers and to score the candidates and select the final answer:

1. As described previously, Watson retrieves text passages from the sources during a primary search phase, and extracts candidate answers from these passages. These passage searches are effective if the answer occurs in close proximity to question key terms in the knowledge sources. SE improves search and candidate recall, and yields relevant search results at higher ranks.

2. Similarly, Watson retrieves documents during the primary search phase and extracts candidates from their titles. These document searches target questions that provide one or more definitions or attributes of an entity and ask for its name, such as the Jeopardy! question *A set of structured activities or a planned sequence of events* (Answer: *program*). Again, the SE approach yields relevant results for additional questions and improves the search rankings.

3. In the answer scoring phase, the search ranks of candidate answers are used as features, and thus better rankings result in more effective candidate scoring.

4. During answer scoring, Watson performs additional searches for supporting passages for each of the top-ranking candidate answers, using a query comprising key terms from the question and the candidate itself. These 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 Watson's scoring framework [Ferrucci et al., 2010]. Thus Watson not only uses the expanded corpora in the initial searches for candidate answers but also leverages them as a source of additional evidence.

5. Finally, other features use corpus statistics such as *idf* scores, which may be more reliable when including the expanded corpora in the sources.

Supporting passage retrieval can be computationally expensive since this step requires a separate search for each individual candidate answer, and thus it may only be feasible on parallel hardware or in batch evaluations where runtime is not important. However, almost every QA system can leverage expanded information sources by retrieving text passages and extracting candidate answers. Furthermore, many open-domain QA systems use a statistical component for answer scoring that can benefit from source expansion by including search-related features such as passage or document ranks and scores.

Similarly to the search experiments in the previous section, we compare Watson's end-to-end performance using Wikipedia and Wiktionary with and without statistical source expansion. In addition, we again use the collection of manually acquired sources comprising encyclopedias, dictionaries, thesauri, newswire corpora, literature, other sources of trivia knowledge, and the TREC 11 reference corpus. This baseline is compared to a setup that includes expanded versions of Wikipedia, Wiktionary and the other two encyclopedias in Section 6.2. Performance is evaluated in terms of

candidate recall and *QA accuracy*. Candidate recall is the percentage of questions for which a correct candidate answer is generated, which is typically lower than search recall because Watson sometimes fails to extract the correct answer from relevant search results. It is an upper bound on accuracy, the percentage of questions answered correctly in first place. For regular Jeopardy! questions, we also report *precision@70*, the percentage of questions Watson answers correctly if it only attempts 70% of the questions for which the top candidate answers have the highest confidence estimates. This is a key measure of performance for regular Jeopardy! questions because contestants do not have to answer all of these questions but can decide when to "buzz in" based on their confidence.

Watson 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 an independent set of 11,550 questions and used the same test sets as in previous search experiments. However, for the TREC task we did not have enough training data to fit answer scoring models with the complete feature set. At first we attempted training models on Jeopardy! data and applying them to TREC questions without adaptation, but this approach proved to be ineffective because the question characteristics and the feature set used by Watson differ between the two tasks. Thus we used a transfer learning approach, fitting a base model to Jeopardy! data and adapting it to the TREC task. In the adaptation step, the factoid questions from TREC 8, 9, 10 and 12 were used to fit a new model that combines scores predicted by the base model with a small set of additional features. Some of these features were already used in the base model, others are specific to TREC. Transfer learning became increasingly important throughout the development of Watson because the number of features in the answer scoring models increased, requiring more and more training data to avoid overfitting. In experiments with the most effective setup of Watson using all available sources including SE, the transfer learning step improved QA accuracy on TREC questions by 7.7 percentage points. However, because most of the TREC questions were used for training, we were left with only TREC 11 (444 questions with known answers) as an independent test set.

6.4.2 Results and Analysis

In Table 6.14 we report Watson's candidate recall with and without source expansion on Jeopardy! and TREC questions when using Wikipedia, Wiktionary or the collection of all manually acquired sources as a baseline. It can be seen that our approach improves candidate recall significantly (with $p < .01$) independently of the dataset and seed corpus. When using all sources, our method yields a 5.2% increase in candidate recall on regular Jeopardy! questions, a 11.0% increase on Final Jeopardy! and a 3.7% increase on TREC 11. The gain on the Final Jeopardy! dataset is largest because there is more headroom for improvements, with only 68.02% recall as a baseline. The improvement on TREC 11 is a conservative estimate because the TREC answer keys are based on correct answers found in the reference corpus, which is included in the baseline, but they do not cover some of the additional correct answers found in the expanded sources.

	Regular J!	**Final J!**	**TREC 11**
Wikipedia	75.43%	61.17%	77.70%
Expansion	80.84%	70.43%	81.53%
% Gain	*+7.2%*	*+15.1%*	*+4.9%*
# Gain/Loss	*+248/-58*	*+92/-19*	*+27/-10*
Wiktionary	25.54%	15.23%	23.65%
Expansion	45.32%	25.76%	45.50%
% Gain	*+77.4%*	*+69.1%*	*+92.4%*
# Gain/Loss	*+808/-114*	*+102/-19*	*+110/-13*
All Sources	82.10%	68.02%	84.68%
Expansion	86.37%	75.51%	87.84%
% Gain	*+5.2%*	*+11.0%*	*+3.7%*
# Gain/Loss	*+206/-56*	*+78/-19*	*+24/-10*

Table 6.14: Candidate recall of Watson on Jeopardy! and TREC questions when using sources that were expanded with web search results. For each setup, 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**
Wikipedia	95.8	97.3	84.8
Expansion	96.2	97.2	84.9
Wiktionary	95.2	96.1	78.3
Expansion	96.9	96.7	78.9
All Sources	95.9	98.7	84.5
Expansion	95.9	98.0	84.3

Table 6.15: Average number of candidates returned by Watson for Jeopardy! and TREC questions when using sources that were expanded with web search results.

We did not use a fixed cutoff point when measuring candidate recall but instead judged all candidate answers returned by Watson. The number of candidates varies between questions and source corpora, but we did not systematically increase it through SE. This can be seen in Table 6.15, which shows the average number of candidate answers returned per question for all source corpora and datasets. These average values are very similar in the baseline experiments and the experiments with source expansion. This is important because candidate recall can easily be increased by considering more candidates, but then the answer selection task becomes more challenging and QA accuracy may be hurt. Note that Watson often generates hundreds of candidate answers per question, but once the candidates have been scored and ranked, the list is cut off at 100 candidates and similar answers are merged. Thus the candidate numbers reported in Table 6.15 are all below 100.

Table 6.16 shows Watson's QA accuracy for the same sources and datasets. Again statistical source expansion yields consistent gains across all baselines and question

	Regular J!	**Final J!**	**TREC 11**
Wikipedia	58.49%	38.58%	50.23%
Expansion	64.94%	47.46%	59.68%
% Gain	*+11.0%*	*+23.0%*	*+18.8%*
# Gain/Loss	*+322/-96*	*+99/-29*	*+61/-19*
Wiktionary	14.71%	5.96%	9.23%
Expansion	28.53%	11.93%	22.75%
% Gain	*+93.9%*	*+100.2%*	*+146.5%*
# Gain/Loss	*+586/-101*	*+58/-11*	*+71/-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*

Table 6.16: QA accuracy of Watson on Jeopardy! and TREC questions when using sources that were expanded with web search results. For each setup, 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.

types. Compared to the strongest baseline comprising all manually selected document collections, SE improves accuracy by 7.6% on regular Jeopardy! questions, by 12.9% on Final Jeopardy! and by 7.6% on TREC 11. These results are based on answer keys that were extended with correct answers returned by Watson in first place and therefore closely reflect true system performance. The improvements from source expansion are statistically significant with $p < .01$, even though only about half of the sources in the strongest baseline were expanded. It is also worth noting that the collection of manually acquired sources includes the AQUAINT newswire corpus, which was used as the reference source in TREC 11 and contains the answers to all questions. Nevertheless, the setup that uses only Wikipedia and its expansion is equally effective on the TREC test set (59.68% vs. 59.46% accuracy) even though these sources are not guaranteed to support the answers. The gains in accuracy exceed the gains in candidate recall on all datasets, which supports our claim at the beginning of this thesis that SE also facilitates answer selection (failure type 3 in Section 1.1).

The expanded sources were first intended for the Jeopardy! challenge, but these results indicate that they are equally effective for TREC questions. This is not surprising because apart from the popularity-based seed selection procedure we did not make any design choices based on Jeopardy! questions, and we have confirmed that the seed selection approach is also effective for TREC. Thus our source expansion method can be applied to build text corpora that are general enough to be leveraged for different QA tasks without adaptation. On the other hand, the strongest baseline in Table 6.16 consists of information sources that were chosen specifically because they improved performance on Jeopardy! and TREC data. This manual source acquisition approach is more likely to overfit since useful content was selected based

	Wikipedia	**Wiktionary**	**All Sources**
No Expansion	77.12%	19.87%	84.12%
(Threshold)	(0.5020)	(0.1720)	(0.6024)
Expansion	83.96%	38.11%	87.74%
(Threshold)	(0.5553)	(0.2426)	(0.6552)

Table 6.17: Precision if Watson answers 70% of all regular Jeopardy! questions using sources that were expanded with web search results. We also show for each setup the confidence threshold that determined whether Watson would attempt to answer.

on the distribution of the questions and topics in these QA tasks, and the selected sources even have perfect coverage for the TREC test set.

In Section 2.2 we described a related approach for leveraging web data to improve QA performance. Clarke et al. [2002] used raw web crawls as sources for question answering and evaluated the impact on the performance of their QA system on the TREC 10 dataset. The web crawler started with web sites of educational institutions and retrieved linked web pages in breadth-first order. The retrieved documents were of relatively high quality, and duplicates were removed from the collection. In these experiments, about 75 GB of web data were needed to match the performance of the 3 GB reference corpus used in the TREC evaluation (a collection of about 1 million newspaper articles). In comparison, we use a much larger document collection as a baseline, which contains 25.6 GB of high-quality, manually selected sources including the TREC 11 reference corpus, and achieve comparable performance by expanding only Wikipedia to about the same total size (25.1 GB). This shows how important it is to target the selection of web data by using appropriate seed topics and effective relevance estimation strategies.

In Table 6.17 we show Watson's precision@70 for different seed corpora and expanded sources on regular Jeopardy! questions. This is the precision if Watson only answers questions for which its confidence in the top candidate exceeds a threshold. 70% of the questions in the regular Jeopardy! dataset have answers with confidence scores above this threshold, and the remaining 30% have lower estimates. It can be seen that Watson's precision is much higher if it only attempts questions when it is relatively certain about its answer. If all manually acquired sources are used, Watson has a precision@70 of 84% without expansion, and 88% with the expansion enabled. This can make the difference between winning or losing a Jeopardy! match against very strong opponents. We also indicate for each source corpus the probability threshold that was used to decide whether to answer. Watson's confidence estimates reflect the amount of supporting evidence found for the answers in the knowledge sources. If source expansion is used, Watson can often retrieve additional evidence for correct answers, resulting in higher probability estimates. Thus it can be more conservative while still answering the same number of questions, i.e. the confidence bar is higher. On the other hand, if there is little supporting evidence in the expanded sources, Watson can be more certain that a candidate answer is incorrect and is less likely to give a wrong answer.

	Regular J!	**Final J!**	**TREC 11**
All Sources	80.49%	66.79%	70.22%
Expansion	82.34%	67.90%	72.81%
	(+2.3%)	*(+1.7%)*	*(+3.7%)*

Table 6.18: Ratio of accuracy over candidate recall for Jeopardy! and TREC questions when using sources that were expanded with web search results.

6.5 Redundancy vs. Coverage

We have seen in Section 4.5 that the statistical source expansion approach increases the semantic redundancy of a seed corpus, and that it also augments the corpus with new information that was not already covered in the seeds. By semantic redundancy we mean paraphrases that state the same information differently, making it more accessible to a QA system. For instance, the same facts may be mentioned repeatedly using different terminology, which increases the probability that at least one relevant text passage closely matches the question and can be retrieved by a QA system. In some cases the information is also made more explicit, e.g. by directly referring to an entity instead of using a nominal or pronominal coreference. In addition, information about related entities that was previously distributed over multiple text passages or documents may be given in the same context within a much shorter text span. Thus the expanded sources can provide additional links between entities, making it easier for a QA system to find the answers to questions involving those entities.

This raises the question whether our method improves QA results by adding semantically redundant information or by increasing the coverage of the seed corpus. There is evidence that SE helps in both ways, but redundancy seems to play a more important role in improving end-to-end QA performance. First of all, we found that the relative gains in QA accuracy are larger than the gains in candidate recall, and thus source expansion must improve answer scoring. In Table 6.18 we show the ratio of accuracy over candidate recall before and after SE for the most effective configuration of Watson that uses all manually acquired sources. This ratio is the percentage of questions that are answered correctly in first place given that a correct candidate answer was found. It can be seen that the ratio is larger with SE on all datasets, i.e. answer scoring is more effective if evidence is retrieved from the expanded sources. Since this analysis is based only on questions whose answers are covered in the corpus, the improvement in answer scoring accuracy can only be explained by increased redundancy. Furthermore, the collection of manually selected sources used as a baseline includes the AQUAINT newswire corpus, which was the reference corpus in the TREC 11 evaluation and has perfect coverage for the 444 questions with known answers in TREC 11. Thus the 7.6% gain in accuracy on this dataset shown in Table 6.16 must be due to an increase in redundancy.

We have also seen that source expansion significantly improves search recall, but it is not clear whether this is because of an increase in coverage or semantic redundancy. If no relevant search results were retrieved without SE, it may be that the answer was not supported by the original sources, or the answer was not found because

of vocabulary mismatches between the query and the corpus or because additional inference would have been required. We looked at all questions in our three datasets for which a relevant search result was retrieved only from the expanded sources and determined whether the correct answer also appears somewhere in the original corpus. Interestingly, this was the case for every one of these questions. However, even if a corpus contains the answer string, it may not support it as the correct answer to a given question. That is, the answer may appear in a different context and it can be impossible, even for a human reader, to deduce that this is the entity the question is asking for. It is often difficult to determine whether an answer is supported by a large document collection, but we did find instances where we are reasonably certain that crucial information is missing in the original sources. Here are examples of Jeopardy! questions that could only be answered after improving the coverage of the baseline corpus through source expansion:

- *Category:* AMERICANA
  Clue: In 2003 chef Daniel Boulud started shaving truffles onto this fast food favorite & charging $50 for it
  Answer: hamburger

  Wikipedia contains an article about *Daniel Boulud* but it does not mention the answer. In addition, we searched the original sources using queries consisting of keywords from the clue and the answer, but were unable to find any documents that mention (ham)burgers with truffles.

- *Category:* HUMANITARIAN LADIES
  Clue: Throughout the 1930s, she taught geography at St. Mary's High School in Eastern India
  Answer: Mother Teresa

  None of the original documents that mention *Mother Teresa* and *St. Mary's High School* contain the information that she was a teacher at this school. One could make an educated guess based on other facts that are covered in the sources, e.g. that Mother Teresa was a female humanitarian who lived in India and that she was a Christian, but this information alone is not enough to be certain about the answer.

- *Category:* BOBBING FOR POETS
  Clue: In 1785 he wrote "Gie me ae spark o' nature's fire, that's a' the learning I desire"
  Answer: Bobby Burns

  The original sources do not contain this quotation – not even substrings such as "nature's fire" or "learning I desire".

- *Category:* LIBRARIES
  Clue: The millionth visitor to this library received a signed copy of "My Turn", some jelly beans & a weekend getaway
  Answer: Ronald Reagan Presidential Library

  We looked at encyclopedia articles about *Ronald Reagan Presidential Library* and *My Turn* (Nancy Reagan's autobiography) as well as other documents that contain both entities, but none of them mentions a price given to the millionth visitor of the library. Of course, the clue contains two hints that point towards Ronald Reagan, the memoirs of his wife and his favorite candy, but again this information would only suffice for an educated guess.

Overall, semantic redundancy seems to account for most of the gain in QA accuracy, which is not surprising since the current implementation of our source expansion method is biased towards adding redundant information. This is because some of the most predictive relevance features estimate the topicality of a text nugget by comparing it to the seed document. If the nugget has high overlap with the seed, it is more likely to be included in the expanded document. However, the paragraph-length text nuggets used in our experiments are often several sentences long, and they frequently contain a mix of novel and redundant information. Thus source expansion still selects new information despite this bias towards content that is similar to the seed.

Future research could focus on mechanisms for controlling the tradeoff between redundancy and coverage, and on determining the ideal combination of the two types of content for source expansion. For instance, longer text nuggets could be extracted from the retrieved documents to select more novel information along with text that closely resembles seed content, or shorter nuggets could be used to focus even more on redundant information. Similarly, more constrained queries could be formulated that include additional key terms extracted from the seed to retrieve documents that are more closely related to existing content. We discuss other techniques for biasing the expansion when proposing areas for future work in Section 9.3. The ideal tradeoff may depend on the seed corpora and the application that leverages the expanded sources. For example, when developing a QA system for the medical or legal domain, one may need to focus on content that is similar to the information in an authoritative seed corpus to ensure that the selected data is reliable. Redundancy can also be more important if the QA domain is relatively narrow and the seeds already have high coverage for that domain.

Chapter 7

Unstructured Sources

In previous source expansion experiments, we relied on a web search engine to retrieve information that is related to the topics of seed documents. This raises the questions of how much our approach depends on high-quality search results, and whether a seed corpus can be expanded even without using a retrieval system. In Section 7.1 we demonstrate that it is possible to extract relevant content directly from a locally stored, unstructured text corpus, and that this extraction-based method is equally effective as search-based source expansion.

We further outline in Section 7.2 how a similar approach may be used to expand unstructured sources in which there exists no one-to-one correspondence between documents and topics for which related content can be gathered. The principal idea is to transform unstructured sources into topic-oriented seed corpora for the source expansion pipeline that was applied previously to encyclopedias and a dictionary. While we address some of the challenges involved in this transformation process, we leave its implementation as a promising direction for future work.

7.1 Extraction-Based Source Expansion

While web searches can be an effective means of obtaining relevant content for source expansion, we now show that a search engine is not required for our method to work. Instead of retrieving web pages about the topic of a seed document, it is possible to automatically browse through a local text corpus and extract content that is related to the topic. This is an important extension of the source expansion approach because it allows us to access information that is not available on the Web. For instance, we can expand a seed corpus with confidential data from the intranet of a company or other proprietary information sources. Similarly, source expansion can be applied to domains for which existing search engines have insufficient coverage (e.g. financial literature) or for which the information provided on web pages can be too unreliable (e.g. the medical and legal domains). In addition, we can reduce network traffic and circumvent restrictions of web search APIs regarding the number of queries that can be issued per host in a given time period, and the maximum number of search results accessible for each query.

Examples of text corpora that can be stored locally and used for the extraction-based source expansion approach include:

- General purpose web crawls that cover a wide range of different topics, or crawls of websites about a particular knowledge domain (e.g. movies databases or political blogs and discussions).
- Pre-existing information sources that target a specialized domain (e.g. collections of scientific publications or product descriptions and manuals), or that fill in gaps in the coverage of an open-domain QA system (e.g. world literature or religious texts).
- News articles that were published during a specific time period (e.g. the 2007–2010 financial crisis), or news about recent events that are not yet covered in the knowledge sources used for QA.

Furthermore, multiple document collections can be combined and used as a large, unstructured corpus of related content for the source expansion methodology described in the following.

7.1.1 Approach

When expanding a seed corpus with information from a locally stored, unstructured source, we first select seed documents about topics that have high coverage in that source. Then we extract text nuggets that mention these topics and compile an initial, noisy pseudo-document for each of the selected seeds. Finally, we remove irrelevant and lexically redundant content by applying a modified configuration of the source expansion pipeline from Chapter 4. Instead of retrieving web pages and splitting them into text nuggets, we use the previously extracted nuggets as input for the statistical relevance model and the merging phase. In the following, we describe each of these steps in more detail.

In these experiments, Wikipedia was again chosen as a seed corpus because it is a useful resource for answering both Jeopardy! and TREC questions. The Wikipedia seed articles are expanded with related content extracted from ClueWeb09[1], a large open-domain web crawl generated in 2009. We used the English portion of the crawl, which comprises about 12 TB of web content, including HTML markup. Web pages that originated from Wikipedia or one of its mirrors were excluded to avoid selecting content that is already in the seed corpus.

Seed Selection

Text corpora that can be stored and processed locally without using excessive hardware resources are necessarily much smaller than the indices of major web search engines such as Google, Yahoo! and Bing, and they can be expected to have lower coverage for the topics in an open-domain seed corpus. Thus if seed documents are

[1] `http://boston.lti.cs.cmu.edu/clueweb09/`

selected based on a measure of popularity, as we did for Wikipedia and Wiktionary in Section 6.2, we risk choosing topics for which not much relevant content can be found in a local corpus. Therefore the expansion should be focused on topics that have high coverage in the source from which related information is extracted. If a text corpus is chosen that is relevant for a given QA task, then many questions will revolve around these topics, and the extracted content is likely to help for answering these questions. The coverage of candidate topics in an unstructured source can be approximated by corpus statistics such as the number of occurrences of a topic in the source or the number of documents that mention it. Often different surface strings are used to refer to the same topic (e.g. *Barack Obama*, *President Obama*, *the president*, *he* etc.) and for more accurate estimates these variants should be taken into account. Note that the candidate set of topics can be very large, and an efficient mechanism is required for quickly looking up mentions in a text corpus.

In the experiments with Wikipedia and ClueWeb09, we ranked all articles in the encyclopedia by the total number of occurrences of their topics in the web crawl. Wikipedia uses manually compiled variants of topics to automatically redirect users who search for alternative forms of an article title. For instance, there are automatic redirects for the queries *Obama* and *President Obama* to the article about the U.S. president *Barack Obama*. We used these redirects as different surface forms of topics in addition to the article titles when counting frequencies in ClueWeb09. In our copy of Wikipedia there are about 3.5 million seed documents and 5.9 million redirects, and we searched 12 TB of web data for occurrences of the topics. To support constant time lookups, we split the article titles and variants into word tokens and stored them in a prefix tree[2]. Then we tokenized the content of each document in the web crawl, traversed the document and looked up token sequences of varying lengths in the tree. If a sequence matched one of the Wikipedia titles or redirects, we incremented the frequency count for the corresponding topic.

To confirm that the ranking of Wikipedia seeds by our estimates of their coverage in ClueWeb09 favors topics that are useful for question answering, we plotted relevance curves for this ranking based on Jeopardy! and TREC datasets. The same methodology was used as for popularity rankings in Section 6.2, except that a seed document was judged as relevant if its title *or any variant* was the answer to a question in the datasets. In Figures 7.1 and 7.2 we show the relevance curves for the two QA tasks and compare them to random rankings, illustrated by straight lines. It can be seen that the rankings by coverage are much more effective, but that there still are relevant topics at low ranks, i.e. the curves continue to have positive slopes. Since we no longer rely on web searches, we can expand large numbers of seeds without worrying about search API restrictions or the data transfer volume. However, it is still important to avoid irrelevant topics since the response time of a QA system increases with the amount of data, and noise in the knowledge sources can hurt search performance. Based on these considerations and the relevance curves, we decided to expand up to 500,000 Wikipedia articles, and we also evaluated search recall using fewer expanded documents.

[2]Also called a *trie* data structure.

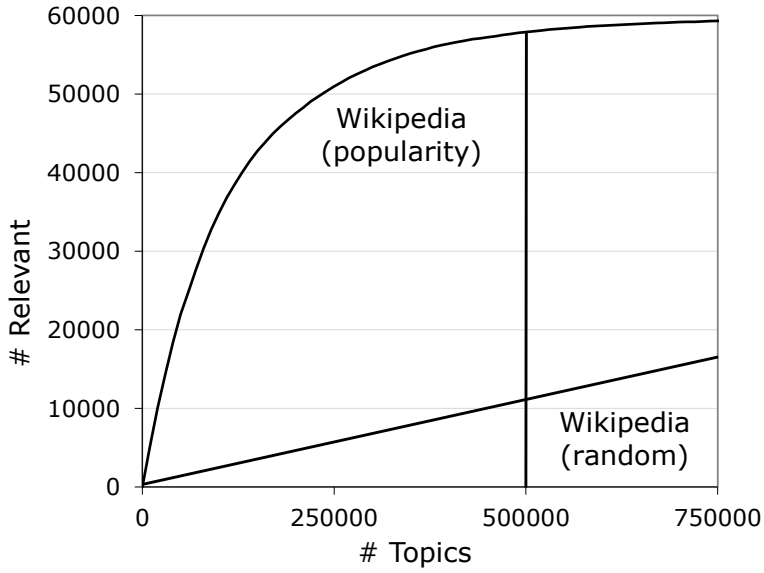


Figure 7.1: Relevance of Wikipedia seed documents for the Jeopardy! task when ranked by their coverage in ClueWeb09 or randomly.

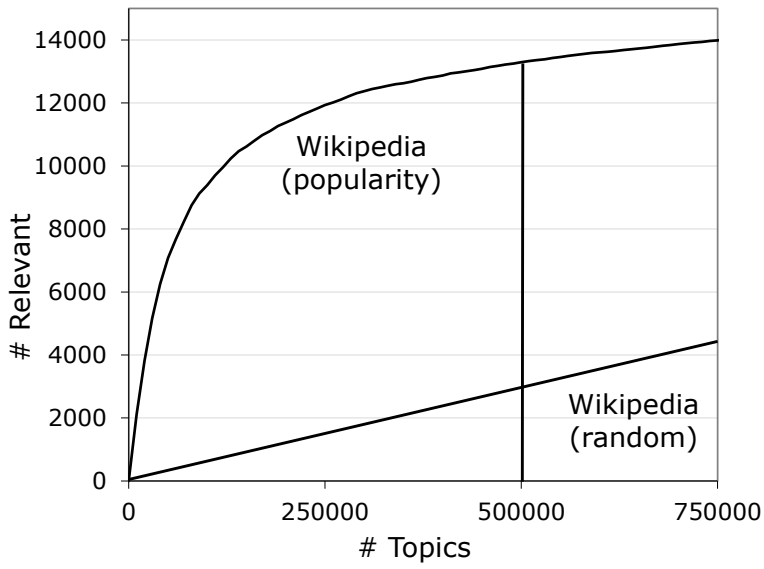


Figure 7.2: Relevance of Wikipedia seed documents for the TREC QA task when ranked by their coverage in ClueWeb09 or randomly.

Many of the seeds with high coverage in the web crawl are about topics that are relevant for Jeopardy! and TREC, but we also selected topics for which there may already be an abundance of useful content in the information sources used for QA. Other metrics for ranking seed documents may be more suitable for choosing topics that are both relevant and helpful for questions that currently cannot be answered. For example, one could compare the frequencies in an unstructured corpus to the counts in the sources that are currently used by a QA system to identify topics that are central to the given corpus but at the same time are not yet sufficiently covered in the QA knowledge sources. We leave this as a possible direction for future research.

Content Extraction

For each selected seed document, text nuggets that mention its topic are extracted from the unstructured corpus and are compiled into a large initial pseudo-document. Similarly to the experiments with web search results, we extracted paragraph-length nuggets delimited by structural HTML markup from ClueWeb09. Again, Wikipedia redirects were taken into account as alternative surface forms of the same topic. The highest-ranked topics were mentioned in millions of text nuggets throughout the web crawl, and it was not feasible to store all these nuggets and estimate their relevance using the statistical model due to space and runtime constraints. Therefore we restricted the size of the pseudo-documents to 2 million characters for each topic. To avoid taking up this space with irrelevant text, the web pages in the crawl were ranked using spam probabilities estimated by Cormack et al. [2011] and the pages that are least likely to be spam were processed first. This approach is specific to web crawls, but one could also prioritize the extraction of text nuggets using a simple and efficient relevance model that is based only on surface features, or if a smaller source is used it may not be necessary at all to preselect content.

Often the extracted text nuggets mention more than one of the selected topics, and thus they are added to the pseudo-documents for multiple seeds. Of course duplicates can also occur when using web search results as a source of related content, but we found that it is much less common that a web page is retrieved with different queries. One could ensure that each text nugget occurs only once in the final expanded corpus to reduce its size and to avoid duplicates in the passage search results of a QA system. However, nuggets that are only relevant with regard to a single topic are likely to be automatically removed from the pseudo-documents for other topics in the merging phase of the SE pipeline. On the other hand, if a nugget contains useful information about multiple topics, document search results may be improved if that nugget is mentioned more than once in different documents. Additional experiments would be required to determine whether it is overall beneficial to remove duplicates from the expanded corpus.

The initial pseudo-documents are too large and too noisy to be useful as sources in a QA system. Topics that are common nouns are frequently ambiguous and their surface forms may not have the intended word sense when mentioned in a text nugget. Most proper names have similar meanings throughout a text corpus, but there are some exceptions, particularly when short forms are used such as *Washington* for

Washington, D.C. or *Bush* for *George W. Bush*. In addition, even if a topic is unambiguous, it may still be mentioned in text nuggets that consist of low quality text or provide irrelevant details. For instance, many of the nuggets for the topic *United States* include such details as meetings that were held in the U.S. or people who live there. Thus the extracted text nuggets are often irrelevant despite containing a surface form of the topic, and an additional relevance estimation step is required to remove noise and reduce the pseudo-documents to a manageable size.

Relevance Estimation

We adapted the source expansion pipeline in Chapter 4 to apply the statistical relevance model to content extracted from a local source. Instead of retrieving documents about a given topic from an external source (step 1) and splitting the documents into text nuggets (step 2), the previously generated pseudo-document for that topic is fetched. The nuggets in the pseudo-document are then scored (step 3) and the most relevant nuggets are merged into a new document (step 4), similarly to the search-based approach. Lexically redundant text is again filtered out, using the token overlap with the seed and previously selected nuggets as a measure of redundancy. The statistical model for estimating the relevance of text nuggets extracted from local sources can be very similar to the model used to score nuggets from search results, with the exception that search-related features are not available. For instance, our relevance model for source expansion based on Yahoo! searches uses the retrieval rank of the source document of a nugget and the overlap of the nugget with the abstract generated by the search engine for that document as features. In the experiments with the local web crawl, we fitted a new model without these two features to the same annotated dataset.

However, the relevance model should ideally be adapted to the characteristics of the local source, and additional features that were previously not necessary or unavailable may be included in the statistical model. For example, the ClueWeb09 corpus contains much more irrelevant information and low-quality text than the top results returned by a web search engine. To better recognize and avoid this noise, precomputed spam probabilities, page ranks or readability estimates for the documents in the crawl could be leveraged as features. If different surface forms of a topic are used when extracting text nuggets that mention it from a corpus, we can also give less weight to nuggets that contain a more generic and possibly ambiguous variant. This could be accomplished by including a feature that is based on the *idf* score of the surface form that appears in a nugget. In Section 8.1 we demonstrate that active learning techniques can be applied to fit a new relevance model without requiring much annotated training data, and in Section 9.3 we propose a method for adapting an existing model to a different source corpus or a new domain by combining active learning with a transfer learning approach.

In Table 7.1 we show size statistics for the Wikipedia seed corpus and expansions with increasing numbers of pseudo-documents generated from ClueWeb09 content. The original 12 TB of redundant and very noisy web data were reduced to 500,000 documents about topics with high coverage in the crawl and a total size of 14.3 GB.

Source	# Documents	Size
Wikipedia	3,482,953	12.6 GB
Expanded Wikipedia	100,000	3.2 GB
	200,000	6.3 GB
	300,000	9.1 GB
	400,000	11.8 GB
	500,000	14.3 GB

Table 7.1: Sizes of Wikipedia and expansions generated from a local web crawl.

In comparison, we selected 300,000 seed documents using popularity estimates and generated a 12.5 GB corpus when expanding Wikipedia with web search results. Thus the expanded corpus that is based on the local web crawl is somewhat larger, but we have seen that more data is not always helpful and can even hurt QA performance. In the experiments with related content from Yahoo! results, Watson's search recall already started to decline on the TREC dataset when more than 200,000 seeds were expanded (see Table 6.7 in Section 6.3.2).

It can be seen that the size of the expanded corpus increases almost linearly with the number of seed documents. Since the length of the expanded pseudo-documents is controlled by a threshold that is relative to the seed length, this means that seed selection based on the estimated coverage in the web crawl only has a small bias towards longer Wikipedia articles. In contrast, when selecting seeds using popularity estimates, the first 100,000 pseudo-documents are much larger (5.8 GB) and the size decreases quickly when seeds with lower popularity are expanded. Thus longer Wikipedia articles are favored, which may improve relevance estimation performance since some of the topicality features used in the statistical model benefit from more seed content. However, additional related information about the most popular topics may be less useful because the documents in the seed corpus already have good coverage for those topics. We further found that the threshold that is used to remove text nuggets with low relevance estimates in the merging phase has little effect on the length of the generated documents when using the crawl, which indicates that there are large amounts of relevant text even for seeds that are ranked relatively low. On the other hand, the top 100 web search results usually contain less relevant information, and pseudo-documents generated from this data for the same topics are on average shorter if the same relevance threshold is applied.

7.1.2 Experiments and Analysis

We evaluated the impact of source expansion with content extracted from the ClueWeb09 crawl on QA performance, and compared this method to the approach that is based on web search results. The experimental setup was described in Section 6.3.1 for search experiments and in Section 6.4.1 for end-to-end evaluations. Watson was used as a framework for all experiments, and performance was measured on both Jeopardy! and TREC datasets. Again, passages and document titles were retrieved

from the original sources and the expanded corpora using Indri and Lucene. The final answers returned by Watson were judged manually and the answer keys were extended with additional correct answers. Search results were judged automatically using these extended answer patterns because a manual evaluation of all results was not feasible. We report search recall, candidate recall and QA accuracy on all datasets, and we also use precision@70 on regular Jeopardy! data as a measure of Watson's performance if it only answers when it has high confidence.

Table 7.2 compares the search recall of Watson on Jeopardy! and TREC questions when only using Wikipedia as an information source to expansions generated with related content from web search results, the local web crawl, or both. It can be seen that the two SE approaches have a similar impact on QA search performance, and that the combination of these methods is most effective. This is not an obvious result because QA performance does not necessarily improve if additional source content is used. For instance, the effectiveness of a QA system can decline if a large, raw web crawl is added to its sources (Section 2.2), and we have shown that there is little additional benefit in expanding more than 200,000 Wikipedia seeds using content from web search results (Sections 6.3.2 and 6.3.4). In the last three rows, we report Watson's search recall when using the collection of all manually acquired sources without expansion, when adding expansions of Wikipedia, Wiktionary, World Book and Microsoft Encarta generated from web search results, and when also including an expansion of Wikipedia generated from the local web crawl. The extraction-based approach yields small but consistent recall gains on top of all other sources even though it was only applied to Wikipedia seeds.

In Table 7.3 we show the improvements in search recall when increasing numbers of Wikipedia seed articles are expanded with content extracted from the web crawl. Recall increases almost monotonically, and the configuration with 500,000 expanded documents is most effective in all experiments except when combining passages and document titles for regular Jeopardy! questions. The largest improvements are realized on Final Jeopardy! data, and there may be headroom for further gains on this dataset by expanding additional Wikipedia seeds using content from ClueWeb09. This is because Final Jeopardy! questions ask about less common topics or obscure facts about popular topics that may not yet be covered sufficiently in the seed corpus or previously expanded documents.

Table 7.4 shows the candidate recall of Watson when using the original document collections and expanded corpora generated with one or both of the SE techniques. Again our approach consistently increases recall independently of the source of related content. It can also be seen that both expansion methods have low variance, i.e. they improve candidate recall far more often than they hurt recall. For instance, when expanding all manually acquired sources using web search results and locally extracted content, correct candidate answers can be found by Watson for an additional 240 regular Jeopardy! questions, but only 63 out of 3,508 questions are hurt. We confirmed that the average number of candidates in the experiments that include content from ClueWeb09 is roughly the same as the numbers given in Table 6.15 for search-based SE and the original sources without expansion. The reported recall numbers are therefore directly comparable.

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 Search	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*
Expansion Crawl	81.13%	72.75%	86.20%	60.03%	58.25%	73.35%	79.13%	54.42%	81.94%
% Gain	*+8.8%*	*+11.6%*	*+6.0%*	*+14.3%*	*+29.7%*	*+15.8%*	*+9.4%*	*+10.1%*	*+6.8%*
# Gain/Loss	*+318/-87*	*+342/-77*	*+230/-59*	*+87/-28*	*+124/-19*	*+100/-21*	*+217/-71*	*+213/-106*	*+171/-60*
Expansion Both	82.44%	74.12%	87.60%	61.55%	60.15%	74.62%	80.86%	53.72%	82.92%
% Gain	*+10.6%*	*+13.7%*	*+7.7%*	*+17.1%*	*+33.9%*	*+17.8%*	*+11.8%*	*+8.7%*	*+8.1%*
# Gain/Loss	*+370/-93*	*+397/-84*	*+274/-54*	*+105/-34*	*+140/-20*	*+112/-23*	*+255/-72*	*+207/-115*	*+200/-68*
All Sources	78.48%	71.07%	**85.55%**	57.11%	52.54%	**69.54%**	75.76%	51.90%	**79.64%**
Expansion Search	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***
Expansion Both	84.29%	78.39%	**90.11%**	63.96%	63.45%	**77.66%**	81.28%	54.89%	**83.53%**
% Gain	*+7.4%*	*+10.3%*	***+5.3%***	*+12.0%*	*+20.8%*	***+11.7%***	*+7.3%*	*+5.8%*	***+4.9%***
# Gain/Loss	*+315/-111*	*+323/-66*	***+209/-49***	*+92/-38*	*+108/-22*	***+90/-26***	*+195/-77*	*+175/-111*	***+149/-66***

Table 7.2: Search recall of Watson on Jeopardy! and TREC questions when using sources that were expanded with web search results and a locally stored web crawl. For each setup, we show the percentage gain and the number of questions gained/lost when comparing the results to the sources without expansion. All improvements are significant with $p < .001$ based on a one-sided sign test.

	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%
Top 100,000	77.34%	68.47%	83.69%	55.08%	49.11%	65.99%	77.30%	51.33%	80.30%
Top 200,000	79.39%	70.44%	85.29%	57.87%	52.92%	69.80%	78.43%	53.63%	81.56%
Top 300,000	80.05%	72.06%	85.80%	58.38%	56.09%	71.83%	78.61%	53.49%	81.75%
Top 400,000	80.90%	72.66%	**86.23%**	59.14%	57.61%	72.72%	78.76%	54.19%	81.61%
Top 500,000	**81.13%**	**72.75%**	86.20%	**60.03%**	**58.25%**	**73.35%**	**79.13%**	**54.42%**	**81.94%**

Table 7.3: Search recall of Watson on Jeopardy! and TREC questions when expanding increasing numbers of Wikipedia seed articles using a local web crawl.

	Regular J!	**Final J!**	**TREC 11**
Wikipedia	75.43%	61.17%	77.70%
Expansion Search	80.84%	70.43%	81.53%
% Gain	*+7.2%*	*+15.1%*	*+4.9%*
# Gain/Loss	*+248/-58*	*+92/-19*	*+27/-10*
Expansion Crawl	80.62%	69.04%	82.43%
% Gain	*+6.9%*	*+12.9%*	*+6.1%*
# Gain/Loss	*+276/-94*	*+91/-29*	*+37/-16*
Expansion Both	82.10%	71.32%	83.11%
% Gain	*+8.8%*	*+16.6%*	*+7.0%*
# Gain/Loss	*+319/-85*	*+107/-27*	*+45/-21*
All Sources	82.10%	68.02%	84.68%
Expansion Search	86.37%	75.51%	87.84%
% Gain	*+5.2%*	*+11.0%*	*+3.7%*
# Gain/Loss	*+206/-56*	*+78/-19*	*+24/-10*
Expansion Both	87.14%	76.02%	89.41%
% Gain	*+6.1%*	*+11.8%*	*+5.6%*
# Gain/Loss	*+240/-63*	*+89/-26*	*+47/-26*

Table 7.4: Candidate recall of Watson on Jeopardy! and TREC questions when using sources that were expanded with web search results and a locally stored web crawl. For each setup, we show the percentage gain and the number of questions gained/lost when comparing the results to the sources without expansion. All improvements are significant with $p < .01$ based on a one-sided sign test.

In Table 7.5 we report Watson's QA accuracy for the same combinations of information sources and datasets. Again an additional performance gain is realized when Wikipedia is expanded with text nuggets extracted from ClueWeb09 on top of using all other manually acquired and automatically compiled sources. The improvement is small on the regular Jeopardy! dataset because the performance bar is already very high, but much larger for Final Jeopardy! and TREC questions. The evaluation results are unbiased since the answer patterns were manually augmented with additional correct answers returned by Watson in first place in any of these experiments. In Section 6.4.2 we have seen that search-based source expansion improves precision@70 on regular Jeopardy! questions from 84.12% to 87.74% when using the large collection of manually acquired sources as the baseline. The extraction-based approach, when applied only to Wikipedia, further improves this metric to 88.68%. This means that if Watson chooses to answer a regular Jeopardy! question, it will be right almost 9 out of 10 times.

The experimental results confirm that the statistical relevance model deployed in the SE system can select relevant information from a very large and noisy text corpus without pre-filtering the content using a retrieval system. Our method works equally well when applied directly to a locally stored corpus, and thus its effectiveness is not explained by the use of a web search engine. In practice, the choice between

	Regular J!	**Final J!**	**TREC 11**
Wikipedia	58.49%	38.58%	50.23%
Expansion Search	64.94%	47.46%	59.68%
% Gain	*+11.0%*	*+23.0%*	*+18.8%*
# Gain/Loss	*+322/-96*	*+99/-29*	*+61/-19*
Expansion Crawl	64.65%	47.46%	61.94%
% Gain	*+10.5%*	*+23.0%*	*+23.3%*
# Gain/Loss	*+358/-142*	*+100/-30*	*+76/-24*
Expansion Both	66.59%	50.76%	62.39%
% Gain	*+13.8%*	*+31.6%*	*+24.2%*
# Gain/Loss	*+426/-142*	*+127/-31*	*+78/-24*
All Sources	66.08%	45.43%	59.46%
Expansion Search	71.12%	51.27%	63.96%
% Gain	*+7.6%*	*+12.9%*	*+7.6%*
# Gain/Loss	*+286/-109*	*+82/-36*	*+44/-24*
Expansion Both	72.32%	54.44%	66.89%
% Gain	*+9.4%*	*+19.8%*	*+12.5%*
# Gain/Loss	*+341/-122*	*+103/-32*	*+70/-37*

Table 7.5: QA accuracy of Watson on Jeopardy! and TREC questions when using sources that were expanded with web search results and a locally stored web crawl. For each setup, we show the percentage gain and the number of questions gained/lost when comparing the results to the sources without expansion. All improvements are significant with $p < .01$ based on a one-sided sign test.

search-based and extraction-based source expansion should be made based on the knowledge domain and the available information sources. For instance, web searches may be unavoidable if up-to-date information is required (such as recent news) or if there simply exists no local corpus for a given domain. Conversely, local sources can be the only choice if the QA domain involves confidential data or specialized knowledge that is not available on the Web. We have also seen that the two SE approaches can complement one another, and that additional improvements in search performance and end-to-end QA results can be attained if these methods are combined.

7.2 Expansion of Unstructured Sources

In this thesis, we applied the statistical source expansion approach to seed corpora in which each document is about a distinct topic that is specified and disambiguated by its title. Topic-oriented document collections such as Wikipedia are widely used for question answering and other natural language processing applications. Furthermore, there exist topic-oriented sources for many specialized knowledge domains, such as dictionaries and encyclopedias of medical, legal or financial knowledge. Nevertheless, real-world applications and evaluation tasks often benefit from or require sources that are not topic-oriented. For example, the Jeopardy! QA challenge can be addressed

more effectively by leveraging additional sources such as world literature and news archives, and participants in the TREC QA evaluations were required to use newswire corpora and blogs as reference sources [Dang et al., 2007].

These document collections cannot directly be expanded with the topic-oriented source expansion pipeline since a document may cover multiple topics, or the same topic may be discussed in more than one document. However, we can transform an unstructured text corpus in which there exists no injective mapping between documents and topics into a topic-oriented source, which can subsequently be expanded with the approach introduced in Chapter 4. For this purpose, the techniques used to expand a seed corpus with content extracted from a locally stored source can be adapted to (1) discover topics that have high coverage in an unstructured source, and (2) generate pseudo-documents from relevant text nuggets about those topics. In the following, we describe these transformation steps in more detail. Their implementation and an evaluation using unstructured seed corpora is left as a promising direction for future research (Section 9.3).

It may be necessary to automatically discover topics in an unstructured source without the aid of a topic-oriented document collection such as Wikipedia. For example, when expanding a crawl of recent news or a collection of publications in a specific research area, a topic-oriented resource that covers the most important topics may not be available. In that case, topics could be identified in the unstructured corpus using named entity recognizers, markup and other metadata, pattern learning and matching techniques, or lookups in dictionaries and ontologies. When transforming a newswire corpus into a topic-oriented source, named entities such as person names, organizations and locations that are mentioned in the news articles may be used as candidate topics, or the entries in a domain-specific ontology could serve as an initial set of topics when using a collection of scientific papers as a seed corpus. In some newswire sources, the articles have also been manually annotated by editors with entities that are central to the news stories and could be used as candidates. Finally, if the expanded sources are intended for a QA system, one could also focus on entity types that are frequently answers to questions in a development set.

We can select candidate topics with high coverage in the unstructured corpus and extract text nuggets that mention those topics by applying the same methodology as in the previous section. Again, one could match different surface forms of a topic and generate a single pseudo-document that contains the extracted nuggets for all different variants. This may improve relevance estimation effectiveness because more content would be available to compute topicality features, and the document retrieval performance of a QA system may increase if related facts about the topic of a question are stored in one place. Instead of relying on Wikipedia redirects to identify variants of a topic, coreference resolution algorithms could be applied to match different surface strings that refer to the same topic within a document, or a dictionary with synonyms could be used to match variants across documents. However, even if alternative forms are treated as separate topics, the approach for transforming unstructured sources can still be effective. Usually there exists a preferred variant that is used most of the time to refer to a topic, and this variant would be chosen if the coverage in the corpus is used as a ranking criterion. If text nuggets that mention this most common form

are extracted, a sufficient amount of relevant content may be found to create a useful pseudo-document.

Given the initial pseudo-documents for the selected topics, we can again use a statistical relevance model to filter out noise and low-quality text. If a topic can be mapped to an existing topic-oriented document such as an article in an encyclopedia or a dictionary entry, that document may be used as a seed to compute topicality features. However, we have shown in Section 5.4 that the relevance models for topic-oriented source expansion are reasonably effective even if no seed content is available. In addition, we have seen that related text that is retrieved or extracted for a given topic can be used in the place of a seed document to estimate topicality since this text on average contains more relevant information than randomly selected content. In the experiments with ClueWeb09 we found that much more topical information can be extracted from the local corpus than from the top 100 web pages retrieved for a topic. Thus, if a large unstructured source is transformed, the effectiveness of topicality features that leverage the extracted text nuggets may increase.

The transformed text corpus may itself already be a useful source for question answering since it contains less noise and the content has been organized by topics. Through effective topic discovery and statistical relevance estimation, we should be able to select information about only the most important topics and to focus on the most relevant facts about these topics. When using a collection of web pages, this method is also likely to remove noise such as markup and embedded code, advertisements and everything else that is not part of the actual content of a web page. In addition, the topic-oriented source can be leveraged to answer factoid questions seeking entities that match a given description, or to retrieve information about a given topic for definitional QA. However, unless the unstructured source already had sufficient coverage and a high degree of semantic redundancy, the transformed corpus should be expanded with additional relevant content. The pseudo-documents can be used as seeds for the topic-oriented source expansion system, and related content can be retrieved or extracted from the Web or large locally stored sources.

Chapter 8

Extensions for Relevance Estimation

Suppose we need to adapt the statistical source expansion approach to a new QA domain, or to a different information retrieval or extraction application. In previous experiments, we fitted relevance models using a large dataset comprising about 1,500 manually annotated web pages. The total annotation time that was required to create this dataset amounted to approximately 5–6 person weeks. In Section 8.1 we demonstrate that active learning can substantially reduce the manual annotation effort with only a slight loss in relevance estimation performance, thus greatly facilitating the adaptation of SE to new domains and applications.

In Section 8.2 we further propose a sequential model that estimates the relevance of text nuggets given their surrounding text in a document. In contrast to our previous approach of using features of adjacent instances to capture the context of a text nugget, we now model dependencies in a probabilistic framework and leverage more distant relationships. The sequential model augments existing relevance features with transition features that are motivated by text segmentation algorithms and predict boundaries between relevant and irrelevant text.

8.1 Active Learning

While there is no shortage of unlabeled data for relevance estimation, labeled text nuggets for supervised learning are not readily available. When expanding Wikipedia and Wiktionary with web search results, we automatically retrieved about 40 million web pages and extracted about 4 billion unlabeled text nuggets (on average about 10,000 per seed or 100 per web page). However, it took about 5–6 weeks to create the annotated dataset described in Section 5.1, which comprises about 160,000 labeled text nuggets. In addition, when we manually labeled this data, we found that less than 6% of all paragraph-length nuggets contain relevant information. The availablility of vast amounts of unlabeled data and the imbalance between relevant and irrelevant instances make the relevance estimation task a natural application of active learning. It has been shown that active learning can benefit from a large pool of unlabeled data

[Tong and Koller, 2001], and that it is particularly effective for sparse classes with few positive examples [McCallum and Nigam, 1998].

The principal idea of active learning is to repeat the following three steps until performance on a labeled test set converges or, if no labeled data is available, until additional iterations have little effect on model parameters:

1. Select an instance from a (possibly very large) pool of unlabeled training data as a query.

2. Ask a human annotator to label the selected instance and add it to a pool of labeled training data.

3. Fit a model to the labeled data, and evaluate it on an independent test set if one is available.

This basic algorithm can be instantiated with different learning techniques and different query selection strategies. In Section 8.1.1 we describe the configurations used in our experiments, and in Section 8.1.2 we present evaluation results.

8.1.1 Experimental Setup

We used logistic regression (LR) and support vector machines (SVMs) with linear kernels for active learning experiments. LR was very effective in the supervised relevance estimation experiments in Chapter 5, and outperformed other popular regression and classification algorithms such as least-squares linear regression and naïve Bayes in a preliminary analysis. Linear SVMs performed worse than LR in a supervised setting, but are nevertheless an interesting alternative for active learning because they often converge rapidly. We also evaluated SVMs with polynomial kernels and radial basis function (RBF) kernels, but they performed similarly to SVMs with linear kernels on our relevance estimation dataset. This is because the relevance features were designed for linear models, i.e. large values indicate that text nuggets are either more likely or less likely to be relevant. The expected sign of the coefficient for each feature in a linear model is shown in Figure 5.3 of Section 5.3.

Initially we select one positive instance and one negative instance from the labeled data described in Section 5.1 to ensure that we can fit and evaluate models after only two iterations. The initial queries are selected randomly and thus each active learning run has a different non-deterministic outcome. This allows us to average over multiple random restarts to reduce the variance of the evaluation results and obtain smoother learning curves. In practice, of course, it is not possible to automatically select relevant or irrelevant text nuggets because the labels of the instances in the pool of training data are not known a priori. Thus a human annotator would need to initially select an instance of each class. Alternatively, nuggets that have a high probability of being relevant can be selected automatically using a single predictive feature, such as cosine similarities or topic likelihood ratios, until at least one relevant and one irrelevant nugget have been found. After this initial phase, we apply one of the following query selection strategies.

Random. We select a random query in each iteration to obtain a representative sample of the full dataset. This strategy preserves the distribution of the data since on average more instances are selected from dense areas in the feature space, and fewer instances from sparse areas. Random sampling does not take into account previously selected queries or the current decision boundary, and is used as a baseline for more informed query selection strategies in our experiments. Note that a random selection of queries on average yields very few positive instances because of the imbalance between relevant and irrelevant text nuggets.

Maximum Score. The unlabeled instance with the largest relevance score estimated by the latest model is chosen as a query in each step. This sampling method selects more relevant text nuggets than random sampling, which should initially result in a steeper learning curve. However, once a reasonably effective model has been found, this strategy can be expected to perform poorly because it will select almost no negative instances.

Diversity. We select an instance in the unlabeled dataset that is farthest from already labeled instances as a query Q:

$$Q = \arg\max_{x \in U} \; Diversity(x) = \arg\max_{x \in U} \; \min_{x' \in L} \; ||x - x'||$$

Here U denotes the set of unlabeled training instances, and L is the set of queries that have been labeled in previous iterations. The features are normalized to have a sample standard deviation of 1, and $||x - x'||$ is the Euclidean distance between the normalized feature vectors. This strategy does not depend on the current model, and thus it is possible to select all queries at once and present them to a human annotator in one batch, which eliminates the waiting time in between queries. The goal of this method is to explore the entire feature space and to fit a reasonably accurate model after only few iterations.

Uncertainty. In each step, we select a query Q the current model is most uncertain about:

$$Q = \arg\max_{x \in U} \; Uncertainty(x)$$

The objective is to choose hard instances that will help the most for improving the relevance model. This sampling strategy gradually refines the decision boundary and often yields very accurate final models, but it can be ineffective in early iterations if queries are selected based on an inaccurate model. In addition, this method has high variance and can converge to a local optimum if the initial decision boundary is too far from the truth.

For logistic regression, we use the misclassification probability to quantify the uncertainty:

$$P_{err}(x) = \begin{cases} P(y=1|x) & \text{if } \hat{y} = 0 \\ P(y=0|x) & \text{if } \hat{y} = 1 \end{cases} = \begin{cases} P(y=1|x) & \text{if } \hat{P}(y=1|x) < 0.5 \\ P(y=0|x) & \text{if } \hat{P}(y=1|x) \geq 0.5 \end{cases}$$

Here y is the true label of the instance x, and $\hat{y}$ is the label estimated by the current logistic regression model. The second equality holds if we predict whichever class label is more likely based on the probability estimate $\hat{P}(y = 1|x)$ from the LR model. We do not know the true posterior probability $P(y = 1|x)$ but can use the LR estimate instead to obtain the following uncertainty term:

$$\begin{aligned} Uncertainty(x) = \hat{P}_{err}(x) &= \begin{cases} \hat{P}(y = 1|x) & \text{if } \hat{P}(y = 1|x) < 0.5 \\ 1 - \hat{P}(y = 1|x) & \text{if } \hat{P}(y = 1|x) \geq 0.5 \end{cases} \\ &= 0.5 - |\hat{P}(y = 1|x) - 0.5| \end{aligned}$$

Thus we can select the query with the largest estimated error probability by maximizing the final expression. This is equivalent to selecting a query x whose probability estimate $\hat{P}(y = 1|x)$ is closest to 0.5.

When using SVMs for active learning, we do not have probability estimates and instead select the query with the smallest Euclidean distance to the current decision boundary. This is equivalent to using the following uncertainty term:

$$Uncertainty(x) = 0.5 - Distance(x)$$

Similarly to the uncertainty function for logistic regression, the maximum uncertainty is 0.5 at the decision boundary, and the value decreases monotonically as the distance of a query to the boundary increases in either direction.

Diversity $\times$ Uncertainty. We select a query Q that maximizes the product of diversity and uncertainty:

$$Q = \arg\max_{x \in U} \; Diversity(x) \times Uncertainty(x)$$

The diversity and uncertainty expressions are identical to those used in the previous sampling strategies. With this method we follow two objectives: to choose instances that are useful for improving the current model (i.e. high uncertainty), and to avoid getting caught in a local optimum by selecting a diverse sample of the training data. We found it effective to take the product of the two selection criteria, but of course one could also combine the terms in a different way, e.g. using a weighted average.

Diversity $\rightarrow$ Uncertainty. This strategy also combines diversity and uncertainty-based query selection. However, instead of applying both criteria at once, we first select a diverse sample of queries and then switch entirely to uncertainty sampling. Based on the first set of queries we can reliably fit a relevance model that is already quite effective. In the second phase, the decision boundary is fine-tuned by focusing on hard queries the model is still uncertain about. This hybrid method can be expected to have lower variance than uncertainty sampling because it selects a more diverse set of instances, and to yield a more accurate model than a diversity-based approach that does not leverage the current decision boundary to select the most useful queries.

In our experiments, we selected the first 200 instances by maximizing diversity, and each of the following queries based on the uncertainty criterion. The cross-over

point was chosen manually based on the shape of the learning curves for diversity and uncertainty sampling. Of course the parameter should ideally be tuned on an independent development set, but this was impractical because we used all available labeled data to evaluate the active learning models and obtain results that are comparable to the supervised learning results in Chapter 5. However, we found that the ranking performance of the final model is not overly sensitive to the precise cross-over point, so we chose a single parameter value and used it in experiments with different learning methods and feature sets.

This multi-strategy approach is similar in spirit to the DUAL sampling strategy introduced by Donmez et al. [2007]. The main differences are that DUAL uses a density criterion instead of maximizing diversity, and that it gradually shifts the focus from density-based sampling towards queries with a higher degree of uncertainty. In addition, Donmez et al. do not use a fixed cross-over point but switch to a different sampling method when the expected error reduction on unlabeled data falls below a threshold. We do not use density-based sampling strategies because these methods depend on parameters that must be tuned on held-out data. For instance, a separate dataset should be used to optimize the number of components in a Gaussian mixture model, or the bandwidth when using kernel density estimation. Similarly, the threshold for a cross-over based on estimated error reduction cannot easily be chosen based on intuition or a visual inspection of learning curves but must be optimized on independent development data.

Each combination of learning method and query selection strategy was evaluated on the manually annotated dataset that was used previously to fit and evaluate supervised models. We performed 12-fold cross-validations on the topics marked as *CV* in Table 5.1, using the instances in the training folds as a pool for query selection. Since the entire dataset has been labeled by annotators, we did not need to query a human at runtime but simulated active learning by only revealing the label of an instance once it had been selected as a query. We focused on paragraph-length text nuggets delimited by structural markup because relevance estimation performance is higher for this type of nuggets (cf. Section 5.3). For each fold, we performed 1,000 active learning steps, evaluated performance on the test data after each step, and averaged the results over 10 random restarts to reduce their variance. We trained models that are based on only the original 19 relevance features described in Section 4.3.2, as well as models that include features of adjacent instances to capture the context of a text nugget, using a total of 55 features. Since we are primarily interested in the ranking performance of the models, we evaluate mean average precision (MAP) after each iteration.

8.1.2 Results and Analysis

In Figures 8.1 and 8.2 we show learning curves for logistic regression models that use only the original relevance features, and LR models that also leverage features of adjacent instances. Comparable learning curves for linear SVMs are given in Figures 8.3

and 8.4. In each plot, the performance of supervised learning using all instances in the training folds is indicated by a solid horizontal line.

The *Random* query selection strategy performs reasonably well if the feature set is small and even catches up with uncertainty sampling after 1,000 steps when using logistic regression models. This is because not much training data is needed to fit a model with only 19 relevance features. If a larger feature set is used, more training data is required to avoid overfitting, and random sampling converges more slowly. *Maximum Score* is usually more effective in early iterations because it selects a more even number of positive and negative instances than random sampling. However, after this initial phase it performs poorly since it almost only selects positive examples, and it is outperformed by the random strategy after 1,000 iterations in most settings. Query selection based on the *Diversity* criterion is initially highly effective because it quickly explores the feature space and yields a relatively accurate model after few iterations. This method also consistently performed well across all test folds in the cross-validation and all random restarts. However, the slope of the learning curves decreases quickly because there is not much merit in selecting additional queries based on this criterion once a diverse sample of the training data has been labeled.

Uncertainty sampling is clearly more effective for SVMs than for LR models. We observed that this method only performs well with logistic regression if a relatively accurate model is found at an early stage. If, on the other hand, queries are selected based on ineffective models, this approach continues to make poor choices and the learning rate can be very low. When using the larger feature set, uncertainty sampling with LR models performs worse than random query selection for over 250 iterations. Sometimes this method cannot recover if the models are too far from the optimum and yields an extremely poor final model. These occasional failures have a noticeable impact on the learning curves obtained by averaging over multiple folds and random restarts. However, when performing live active learning one could start over if the algorithm is caught in a local optimum. This is not difficult to detect even without labeled test data because most of the selected queries would be negative examples. When using SVMs, there is a strong theoretical motivation for uncertainty sampling that explains why this method is very effective. It can be shown that if the data are linearly separable, the worst-case expected rate of convergence to the true decision boundary is maximized by a query selection strategy that halves the size of the current version space in each iteration [Tong and Koller, 2001]. The version space [Mitchell, 1982] is the set of all decision boundaries that are consistent with the already labeled data, i.e. that classify all instances correctly. By always choosing a query with maximum uncertainty, i.e. an unlabeled instance that is closest to the current decision boundary, we approximately achieve the goal of cutting the version space in half.

The multi-strategy approach *Diversity* $\times$ *Uncertainty* is most effective after 1,000 active learning iterations in most configurations. Like uncertainty sampling, it sometimes performs poorly in early iterations when combined with logistic regression, but because of the diversity criterion it usually recovers and then improves ranking performance rapidly. When used with SVMs, this method consistently yields competitive results independently of how many queries have been labeled. The cross-over method

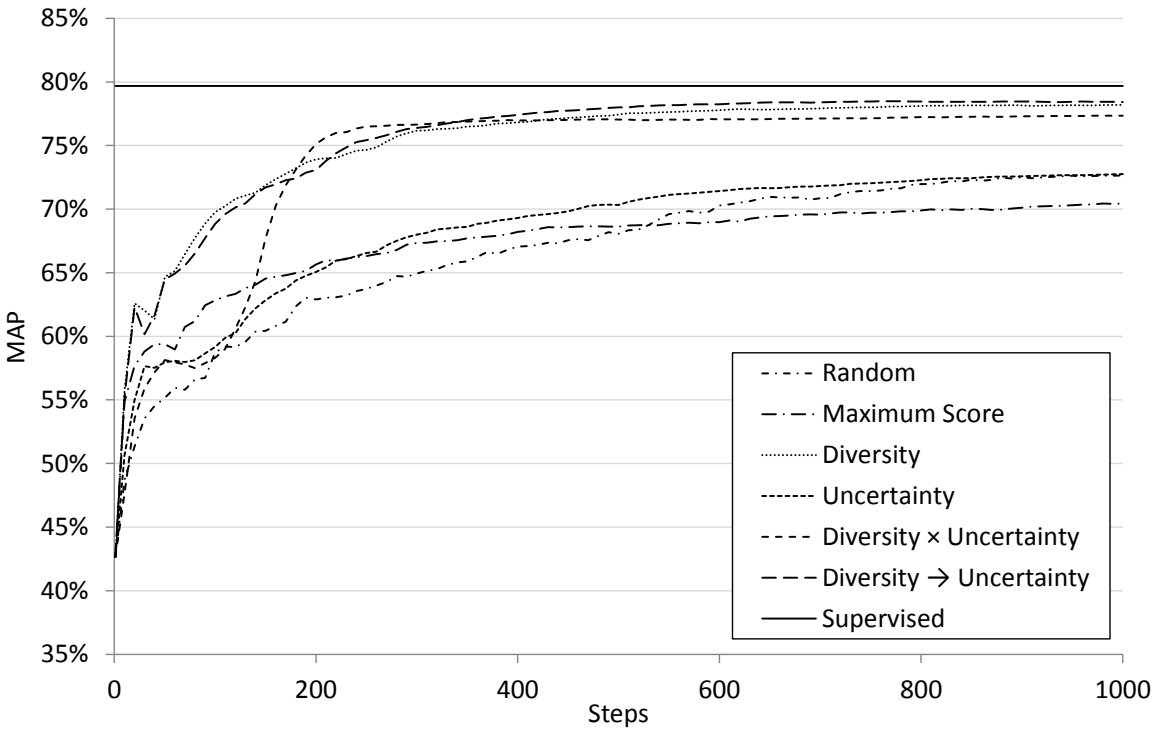


Figure 8.1: Active learning curves for logistic regression models that make independent predictions using only the original features.

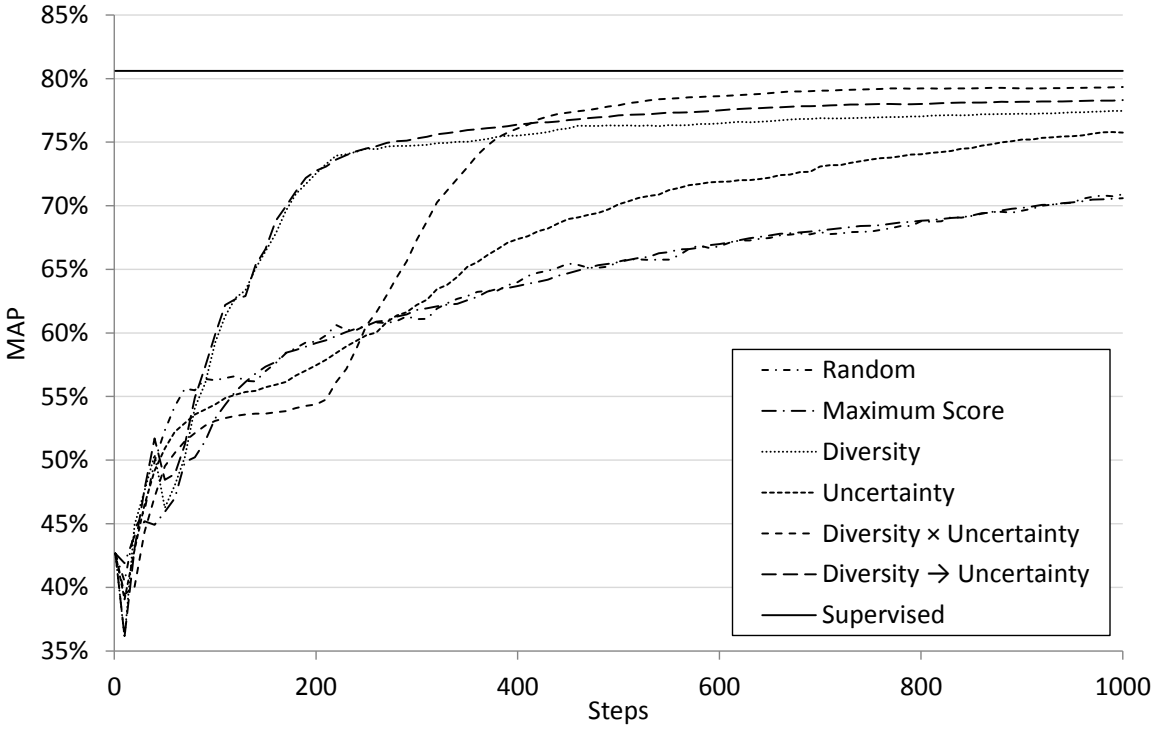


Figure 8.2: Active learning curves for logistic regression models that include features of adjacent instances to capture dependencies.

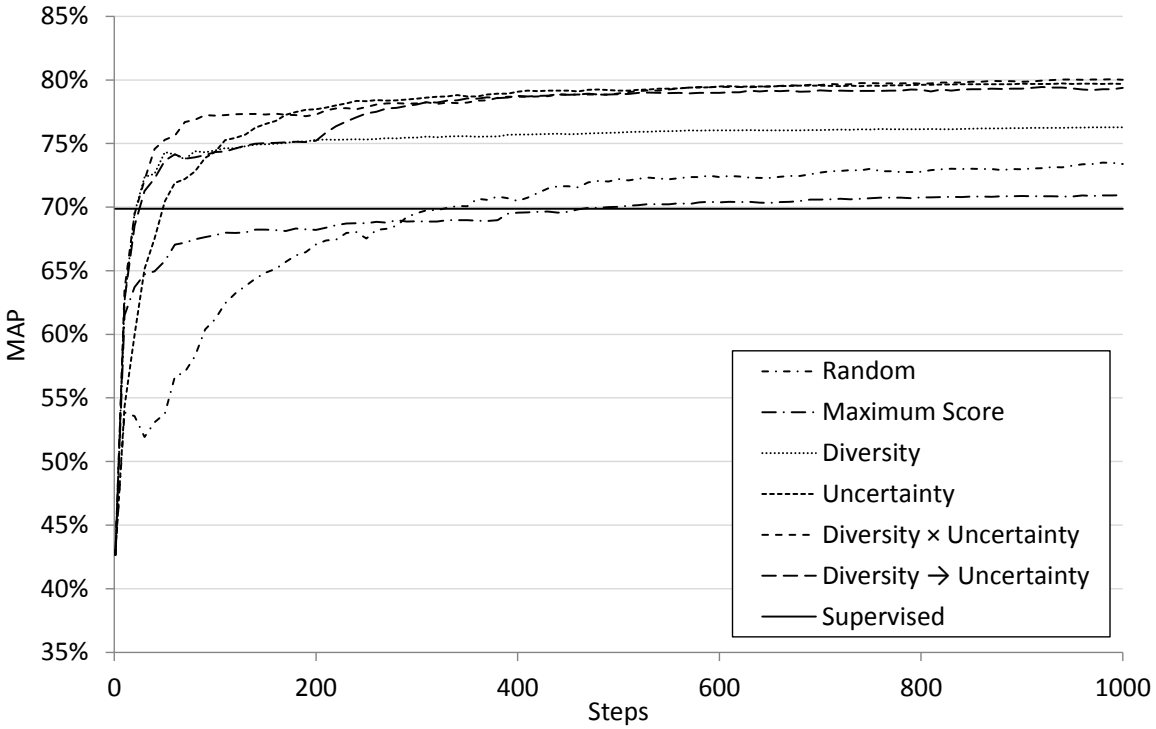


Figure 8.3: Active learning curves for SVMs with linear kernels that make independent predictions using only the original features.

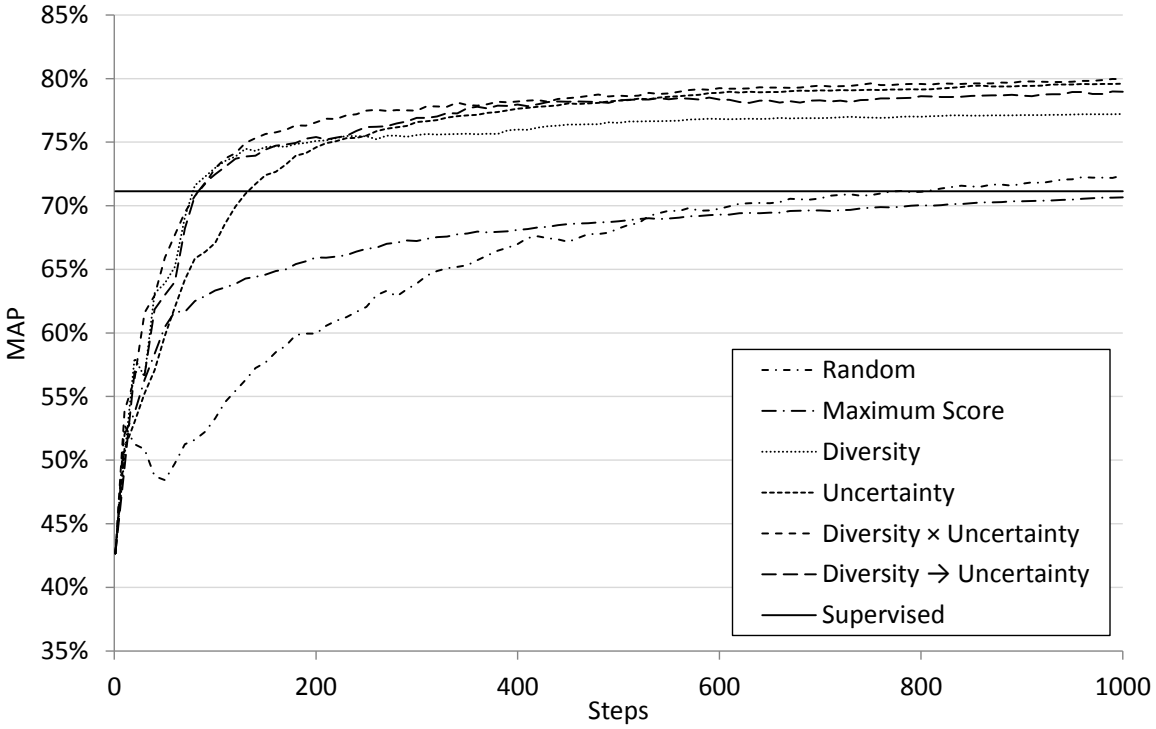


Figure 8.4: Active learning curves for SVMs with linear kernels that include features of adjacent instances to capture dependencies.

Diversity → *Uncertainty* combines the relative advantages of the individual query selection criteria by first exploring the feature space to avoid local optima and selecting hard instances in later iterations to further refine the model. Note that this hybrid approach selects the first 200 queries based on the diversity criterion and thus the learning curves for this approach and diversity sampling are initially very similar (though not identical because the first two queries in each active learning run are selected at random). Once the hybrid method switches to the uncertainty criterion, it outperforms diversity sampling in all settings. It is also more robust and on average more effective than the single-strategy uncertainty approach when using logistic regression. After 1,000 steps, the performance of the cross-over method comes close to the strategy that continuously selects queries based on a combination of diversity and uncertainty. However, we may have slightly overestimated its effectiveness since we chose a cross-over point based on the learning curves for the single-strategy methods on the same dataset.

The learning curves for LR models and SVMs with features of adjacent instances converge more slowly, which is not surprising because more training data is needed if a larger feature set is used to avoid overfitting. After 1,000 steps, many of these learning curves still have distinctly positive slopes, and thus performance could be improved further by labeling additional queries. It can also be seen that SVMs converge faster than LR models, which makes sense because the decision boundary of an SVM can be defined by a small number of support vectors. Thus a few instances that are closest to the decision boundary implied by the full training set can be sufficient to attain high accuracy. If these (or similar) instances are selected early on, performance increases quickly and queries that are selected in subsequent iterations do not improve the decision boundary much anymore.

However, when using all available training data to fit models, the performance of SVMs is much lower than after 1,000 active learning iterations and logistic regression is clearly more effective for relevance estimation. This result is surprising, but note that we evaluated the models on independent test folds, and there is no guarantee that a model trained using more data has a higher test performance. Instead, there may be outliers in the training data that were not selected as queries for active learning but affect the generalization performance of supervised models. If outliers are chosen as support vectors, they can substantially alter the decision boundary and cause overfitting. We also found that it is computationally more expensive to train SVMs than to fit logistic regression models. A single active learning step sometimes took over a minute when using SVMs and the large feature set, which would be problematic when interacting with a human annotator in realtime. We used a Java implementation of LIBSVM [Chang and Lin, 2011] and a server with 3 GHz Xeon CPUs and 32 GB RAM. However, the runtime could be reduced by selecting a subset of the features, by sampling queries from a smaller pool of training data, or by using an SVM implementation that is optimized for linear kernels.

In Table 8.1 we show for each combination of learning method, feature set and sampling strategy the percentage of positive queries that were selected, averaged over all cross-validation folds and all random restarts. It can be seen that less than 7% of the instances are relevant if queries are selected at random, which explains

	Logistic Regression		Linear SVM	
	Independent	**Adjacent**	**Independent**	**Adjacent**
Random	6.87%	6.94%	6.91%	6.84%
Maximum Score	91.63%	89.23%	91.99%	90.83%
Diversity	18.56%	14.00%	18.72%	13.93%
Uncertainty	44.89%	44.79%	44.02%	46.07%
Diversity × Uncertainty	45.38%	44.89%	48.10%	42.14%
Diversity → Uncertainty	45.91%	43.94%	44.63%	45.57%

Table 8.1: Percentage of relevant instances selected by different combinations of learning methods and query strategies. We show results for models that make independent predictions using only the original features, and models that include features of adjacent instances to capture dependencies.

the relatively low performance of this strategy. If the instance with the maximum relevance estimate is chosen in each iteration, about 89–92% of the queries selected after 1,000 steps are positive and thus the sample is again highly unbalanced. In the long run, uncertainty sampling selects a roughly equal number of positive and negative instances since it focuses on hard queries that are close to the decision boundary but can be on either side of it. The reported numbers are below 50% because in early iterations the models are still inaccurate and it is more likely that negative instances are selected if uninformed choices are made.

Diversity-based query selection yields a larger number of positive instances than the random sampling method because it selects more queries from sparse areas in the feature space. This is where most of the positive instances can be expected since there are fewer relevant text nuggets and their feature values can vary widely. Compared to uncertainty-based methods, this approach selects more positive instances in early iterations while the models used for uncertainty sampling are still unreliable, but fewer relevant instances in the long run. This is one reason why diversity sampling is initially so effective but is ultimately outperformed by methods that incorporate a measure of uncertainty. In general, we found that the fastest rates of convergence are often attained by sampling strategies that select the most balanced data, independently of the number of active learning steps. This observation consistently holds for both logistic regression and SVMs, and for both independent prediction models that use a small feature set and models that leverage features of adjacent instances.

Logistic regression models and linear SVMs trained by active learning have mean average precisions of up to 79.33% and 80.02%, respectively. In contrast, LR models with features of adjacent instances fitted to all data perform at 80.59% MAP. Thus the predictive performance of models trained on only 1,000 well-chosen queries comes close to the performance of supervised learning using over 100,000 training instances in each step of the cross-validation. We further investigated whether the small loss in relevance estimation performance when using active learning techniques instead of fitting a supervised model to pre-labeled data is noticeable in a task-based evaluation on QA datasets. For this experiment, we used a logistic regression model with ad-

	Regular J!	**Final J!**	**TREC 8–12**
Wikipedia	81.33%	63.32%	76.74%
Expansion Supervised (130,119 training instances)	86.23% *(+6.0%)*	72.21% *(+14.0%)*	82.17% *(+7.1%)*
Expansion Active Learning (1,000 training instances)	86.06% *(+5.8%)*	71.07% *(+12.2%)*	81.09% *(+5.7%)*
Wiktionary	30.39%	13.32%	29.15%
Expansion Supervised (130,119 training instances)	51.20% *(+68.5%)*	27.79% *(+108.6%)*	52.46% *(+80.0%)*
Expansion Active Learning (1,000 training instances)	49.80% *(+63.9%)*	25.13% *(+88.7%)*	49.93% *(+71.3%)*

Table 8.2: Impact of source expansion on QA search recall when using relevance models obtained through supervised learning or active learning.

jacent features because this method consistently performed well in all experiments, and because it was easy to integrate in our SE system. Queries were selected based on the strategy *Diversity* × *Uncertainty* because this method is most effective after 1,000 iterations when using logistic regression and the large feature set. In contrast to the second best strategy *Diversity* → *Uncertainty*, this approach does not require tuning a cross-over parameter.

The final model after 1,000 active learning steps was applied to expand Wikipedia and Wiktionary seed corpora using the exact same methodology as in Chapter 6. The relevance threshold in the merging phase of the SE system was adjusted to ensure that the expanded corpora have the same size as those previously generated with the supervised model. We then evaluated the impact of the expanded sources on QA search recall using both Jeopardy! and TREC datasets, and compared the performance to SE using a supervised model fitted to 130,119 labeled text nuggets. The results are summarized in Table 8.2. It can be seen that the model obtained through active learning is almost as effective for the QA search task as the supervised model trained on a much larger dataset. Thus active learning can be an interesting option in situations where it is not feasible to annotate a large amount of training data. On the other hand, if enough labeled data is available or can be annotated, it is more effective to fit a logistic regression model to the full dataset.

We found that it only takes about one person day to label 1,000 text nuggets selected by an active learning algorithm, a substantial reduction in annotation time when compared to the 5–6 person weeks it took to label the entire dataset. This result supports our initial claim that active learning can greatly facilitate the adaptation of the statistical SE approach to new domains and applications. Here we already take into account that it is more time-consuming to label 1,000 text nuggets that were selected individually than to annotate whole documents of the same total length. We also found that it is feasible to accurately label independent text nuggets as long as the annotator first familiarizes himself with the seed topics and is shown the source documents of the nuggets during the annotation process.

8.2 Sequential Models

To estimate the relevance of text nuggets with regard to a seed document, we initially fitted a logistic regression model that scores each nugget independently based on a set of relevance features (see Sections 4.3.2 and 4.3.3). However, we found that text nuggets extracted from the same document are not independent, but a nugget is more likely to be relevant if preceding or following nuggets are relevant and should therefore be evaluated in the context of surrounding nuggets. In a first attempt, we added features of adjacent nuggets to the independent LR model to capture dependencies between nuggets in an ad-hoc fashion.

We now propose a sequential model that leverages these dependencies in a more principled manner. Our sequential model resembles a hidden Markov model (HMM) with two states for relevant and irrelevant nuggets, but the probabilities of state transitions are estimated dynamically based on transition features that measure the lexical coherence of the text at nugget boundaries. In the following, we introduce these transition features (Section 8.2.1) and derive a graphical model that integrates the transition features with the previously discussed relevance features (Section 8.2.2).

In Sections 8.2.3 and 8.2.4, the sequential model is compared to the logistic regression models that solely rely on relevance features. We show that it outperforms the independent LR model on both markup-based text nuggets and sentence-level nuggets. The sequential model also yields small but consistent gains in ranking performance over the LR model with features of adjacent instances on both types of nuggets, and it uses fewer parameters and requires less time for training.

8.2.1 Transition Features

We observed that text nuggets appearing in the same document are usually not independent, but relevant nuggets are likely to be surrounded by other relevant nuggets, and irrelevant nuggets by other irrelevant ones. Thus, using the notation L_t for the label of the t-th nugget in a document, 1 for relevant and 0 for irrelevant, we would intuitively expect the following relations:

$$P(L_t = 1|L_{t-1} = 1) > P(L_t = 1|L_{t-1} = 0),$$
$$P(L_t = 0|L_{t-1} = 0) > P(L_t = 0|L_{t-1} = 1).$$

Sequential models such as hidden Markov models would use static estimates of these transition probabilities to transfer evidence of relevant text from one text nugget to subsequent nuggets.

However, we found that transitions between relevant and irrelevant text are not at all equally likely at different positions in a document. For instance, transitions are more likely if the structure and vocabulary of the text nuggets change abruptly, and less likely within a coherent window of text. To leverage this additional source of information, we developed a set of features which are indicative of the coherence of the text on either side of a given boundary. These *transition features*, along with the range of their values and intuitive motivations, are listed below. The features are computed at each boundary between text nuggets.

MarkupBoundary (binary). Whether the nugget boundary coincides with structural HTML markup, such as tags for paragraphs, table cells or items in a list. Transitions between relevant and irrelevant text often occur at such markup boundaries, since different sections in a web page are less likely to be related than contiguous text. Note that this feature is always *true* if nuggets are extracted based on markup information, and thus it is only useful for the shorter sentence-level nuggets.

TextTiling (continuous). To measure the lexical coherence of the text at nugget boundaries, we implemented four features derived from the TextTiling algorithm [Hearst, 1997]. The simplest of these features, denoted *CosSim*, measures the cosine similarity between word frequency vectors derived from windows of text on either side of a given nugget boundary. More precisely, we tokenize the text and extract a window spanning the n tokens that immediately precede a nugget boundary, and an equally sized window spanning the tokens that follow the boundary. Function words and punctuation marks are dropped, and the remaining tokens are normalized by stemming them with an implementation of the Porter stemmer [Porter, 1980] and converting them to lower case. Let W_i be the set of all normalized words that occur in the two windows that are adjacent to the i-th nugget boundary, and let $f_{i,1}(w)$ and $f_{i,2}(w)$ denote the frequencies of a word $w \in W_i$ in the two windows. Then the cosine similarity at the i-th boundary is defined as

$$CosSim(i) = \frac{\sum\limits_{w \in W_i} f_{i,1}(w)\ f_{i,2}(w)}{\sqrt{\sum\limits_{w \in W_i} f_{i,1}(w)^2 \sum\limits_{w \in W_i} f_{i,2}(w)^2}}.$$

Note that Hearst [1997] computes cosine similarities based on plain word counts, but one could also use *tf-idf* scores or other term weighting schemes. We experimented with different window sizes and found a value of $n = 50$ to be relatively effective. However, the algorithm did not appear to be overly sensitive to the choice of this parameter, and performs similarly for larger windows spanning 100 or even 200 tokens.

The cosine similarities may depend on properties of the document, such as the writing style and vocabulary size, and thus their absolute values are often poor indicators for transitions between relevant and irrelevant text. In addition, if the similarities are consistently low over a series of nugget boundaries, as illustrated in Figure 8.5 (b), this does not necessarily imply that there is a transition between relevant and irrelevant text at each boundary, but it could also be indicative of an incoherent sequence of irrelevant text. For instance, the front page of a news website may contain a list of abstracts of unrelated news articles, or a discussion board may list the titles of all current threads. Thus, it seems more prudent to compare the similarity value at a given boundary to the similarities at nearby boundaries, and look for "valleys" in the similarity scores. In Figure 8.5 (a) the depth of the valley at the i-th boundary can be measured from the local maxima at $i - 2$ and $i + 1$. More formally, Hearst proposes the following algorithm for computing the depth of the valley at the i-th nugget boundary:

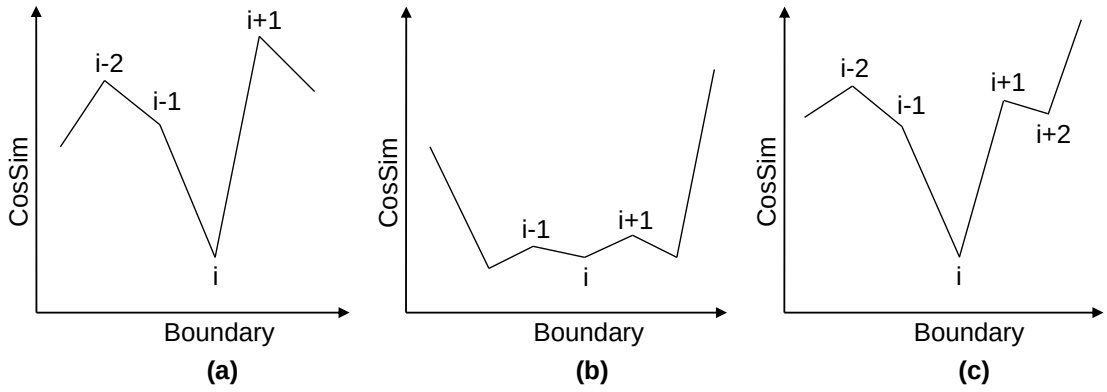


Figure 8.5: Depth computations in the TextTiling algorithm.

1. Set L = i, R = i
2. While CosSim(L - 1) $\geq$ CosSim(L)
   Set L = L - 1
3. While CosSim(R + 1) $\geq$ CosSim(R)
   Set R = R + 1
4. Set DepthL = CosSim(L) - CosSim(i)
5. Set DepthR = CosSim(R) - CosSim(i)
6. Set CosSimDepth = DepthL + DepthR

This depth score based on cosine similarities was added as another TextTiling feature, denoted *CosSimDepth*.

Hearst noted that small dips in the cosine similarity scores may result in local maxima which can affect the computation of the depth scores. This behavior is illustrated by the example in Figure 8.5 (c), where the depth computation for boundary i is interrupted by the small valley at $i + 2$. To increase the robustness of the depth feature to small perturbations in the similarity scores, Hearst proposes a smoothed version of the cosine similarity, computed as follows:

$$CosSimSmooth(i) = \frac{1}{2s+1} \sum_{j=i-s}^{i+s} CosSim(j).$$

The smoothing width s controls the amount of smoothing applied to the original similarity scores. We added *CosSimSmooth* as a third TextTiling feature, using the recommended parameter value of $s = 1$. Then, based on this feature, we computed depth scores using the smoothed similarities, yielding a forth TextTiling feature denoted *CosSimSmoothDepth*.

Figure 8.6 shows the values of the TextTiling features for a portion of a web page retrieved for the topic *Iran-Iraq War*. The values on the x-axis indicate whether a

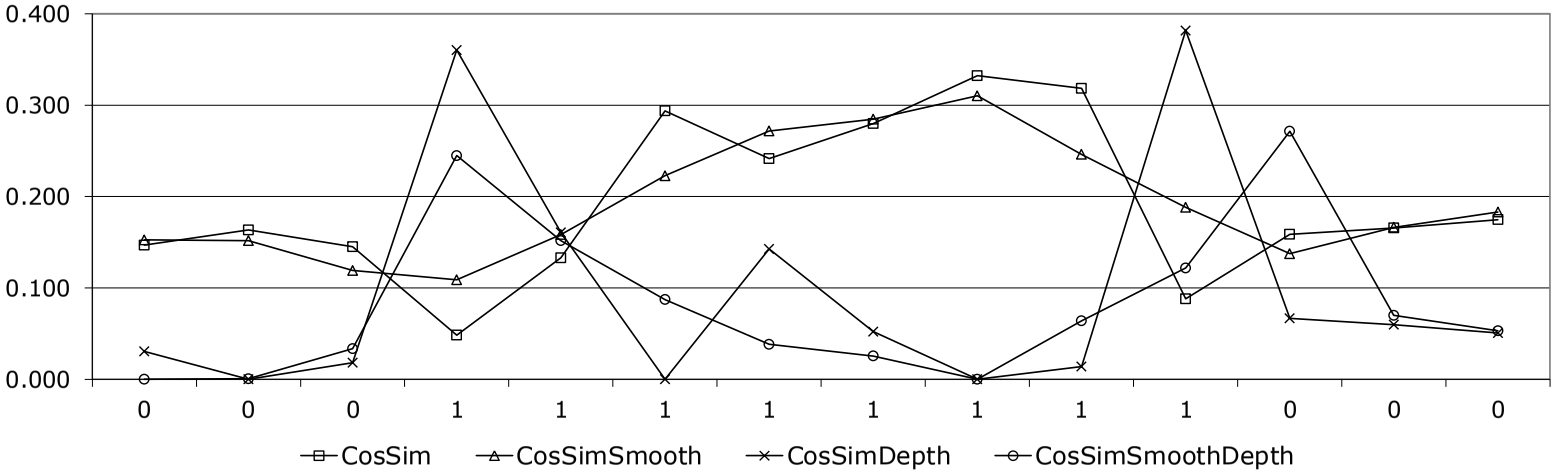
text nugget is relevant (1) or irrelevant (0), and the y-axis gives the feature values for the boundary between that text nugget and the previous nugget. The (abbreviated) nugget text is shown in the table below the plot. The nugget boundaries were determined based on structural markup in HTML documents. For the features *CosSim* and *CosSimSmooth*, a small value indicates a likely transition between relevant and irrelevant nuggets, whereas for the features *CosSimDepth* and *CosSimSmoothDepth* a large value is indicative of a transition. In this example, *CosSimSmoothDepth* perfectly predicts the transitions from irrelevant to relevant and back to irrelevant text. However, it should be noted that this is a rather "friendly" example selected for illustration purposes, and that the TextTiling approach often misses transitions or predicts transitions at wrong locations.

TransitionLR (continuous). We fitted two language models to a dataset of manually labeled text nuggets introduced in Section 5.1, using the topics marked as *LM* in Table 5.1 as training data. A transition language model was trained on only those text nuggets that immediately follow a transition between relevant and irrelevant text (i.e. the first nugget in a sequence of relevant or irrelevant nuggets), and a background model was trained on all nuggets. Both models are unigrams with Good-Turing discounting. This feature is the ratio of the likelihood of the text nugget that follows a boundary under the transition model and the background model. It is intended to increase or decrease the probability of a transition if certain trigger words occur. For instance, consecutive relevant nuggets often contain words such as *however*, *thus* and *but*, or pronouns that refer to entities mentioned in previous nuggets.

Even though smoothed unigram models are very robust and require little training data, we realize that the datasets used to fit the language models (particularly the transition model) may be too sparse to achieve optimal performance. We were unable to use more training data because most of the annotated data had to be reserved for the estimation of feature weights and the evaluation of our models. An alternative to this language modeling approach may be the use of a manually compiled dictionary of discourse markers (e.g. *so*, *because*, *now*, *actually*) to detect coherent text passages. However, the web data used for source expansion is very heterogeneous, and it may be difficult to identify a comprehensive set of discourse markers that is effective across different domains and writing styles.

CharacterLengthDiff (continuous). The absolute value of the percentage difference between the number of characters in the text nugget that immediately precedes the boundary and the text nugget that immediately follows it. Documents frequently contain sequences of relevant nuggets that have approximately the same length, such as items in a list or table, or paragraphs in a narrative. A transition between relevant and irrelevant text is more likely if this feature has a large value.

3rdPersonPronoun (binary). This feature is identical to the *3rdPersonPronoun* feature in the relevance models, but it is used for a different reason in transition models. When developing relevance features we observed that a third person pronoun



Relevance	Nugget Text (abbreviated)
0	Iran-Iraq War
0	Military History Companion: Iran-Iraq war
0	Home > Library > History, Politics & Society > Military History Companion
1	On 22 September 1980, Iraqi forces crossed the Iranian border in strength, igniting what was to become one of the longest, ...
1	It has become a commonplace to view the invasion as a natural offspring of the aggressive personality of the Iraqi president, ...
1	Be that as it may, the Iraqi war strategy was fundamentally flawed. Instead of dealing a mortal blow at the Iranian armed ...
1	In January 1981 Iran carried out its first major counter-offensive since the beginning of the war, and by the end of the year Iraq ...
1	From now on the war would become a prolonged exercise in futility, reminiscent of the trench warfare of WW I, in which ...
1	By this time, the spectre of an Iranian victory had brought a group of the most unlikely bedfellows to do their utmost to ensure ...
1	Meanwhile Iran, starved of major weapons systems and subjected to a punishing economic blockade, was showing growing signs ...
1	This was soon followed by a string of military successes. In May Iraq drove the Iranians out of their positions east of Basra, and ...
0	– Efraim Karsh
0	Answers.com
0	Home Page

Figure 8.6: Example of TextTiling features for estimating transitions between relevant and irrelevant text. The plot shows the feature values at boundaries between relevant (1) and irrelevant (0) nuggets. The corresponding nugget text is given in the table below.

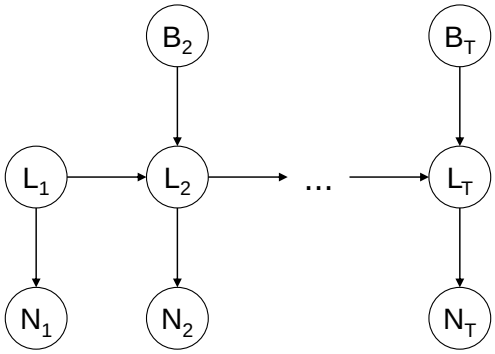


Figure 8.7: Graphical representation of the independence assumptions in the sequential model for relevance estimation.

often refers to the seed topic, and thus the text nugget that contains it is often topical. However, if a text nugget contains a pronominal coreference it is also more likely to be related to previous text, and thus a transition is less likely to occur.

8.2.2 Graphical Model

Let $N_1, ..., N_T$ be a sequence of text nuggets extracted from a document with labels $L_1, ..., L_T \in \{1 = \text{relevant}, 0 = \text{irrelevant}\}$, and let $B_2, ..., B_T$ be the boundaries between nuggets N_1 and N_2, N_2 and N_3, and so forth. If we make certain independence assumptions about these random variables, we can use dynamic programming to estimate the probability that each text nugget is relevant given *all* text nuggets in the same document. More precisely, we will estimate the probability $P(L_t = 1|N_1...N_T)$ as opposed to $P(L_t = 1|N_t)$ when using an independent logistic regression model.

The directed graphical model in Figure 8.7 illustrates our independence assumptions about the nuggets, labels and boundaries. In particular, we assume the following for $t \in \{2, ..., T\}$:

$$P(N_t|N_1...N_{t-1}, L_1...L_t, B_2...B_t) = P(N_t|L_t), \tag{8.1}$$
$$P(L_t|N_1...N_{t-1}, L_1...L_{t-1}, B_2...B_t) = P(L_t|L_{t-1}, B_t). \tag{8.2}$$

These are reasonable assumptions since the distributions of text nuggets and their labels can be expected to depend primarily, though not exclusively, on their immediate neighbors. The model can be regarded as a generalization of a hidden Markov model [Rabiner, 1989] since the probability of a transition to a label L_t does not only depend on the previous label L_{t-1} but also on the current nugget boundary B_t described by the transition features from Section 8.2.1.

The *emission probability* of nugget N_t given its label L_t can be estimated as follows according to Bayes' rule:

$$P(N_t|L_t) = \frac{P(L_t|N_t)P(N_t)}{P(L_t)} \sim \frac{P(L_t|N_t)}{P(L_t)}.$$

The prior probability $P(N_t)$ can be ignored since it is a constant that does not depend on the label, and the marginal probability $P(L_t)$ can be estimated by maximum likelihood estimation (i.e. relative frequency estimation) from a training set of labeled text nuggets as follows:

$$P(L_t = 1) = \frac{\text{Number of relevant nuggets}}{\text{Total number of nuggets}},$$
$$P(L_t = 0) = 1 - P(L_t = 1).$$

The conditional probability $P(L_t|N_t)$ can be estimated with the independent logistic regression relevance model introduced in Section 4.3.3 using the relevance feature set described in Section 4.3.2.

The *transition probability* $P(L_t|L_{t-1}, B_t)$ between labels L_{t-1} and L_t at boundary B_t can also be estimated with logistic regression models. However, a different set of features is required for predicting whether there is a transition between a relevant and an irrelevant nugget given the text on both sides of a boundary. These transition features were described in Section 8.2.1. We fit two separate transition models, one for estimating $P(L_t|L_{t-1} = 0, B_t)$ and another for estimating $P(L_t|L_{t-1} = 1, B_t)$. The first transition model is trained only on boundaries following text nuggets that were labeled as irrelevant, and the second model is trained only on boundaries following relevant text.

Given estimates of the emission probabilities and transition probabilities, a Viterbi decoder can be used to determine the most likely label sequence for a sequence of text nuggets extracted from a document. For $t \in \{1, ..., T\}$ and $i \in \{0, 1\}$, let

$$V_t(i) := \max_{L_1 ... L_{t-1}} P(N_1 ... N_t, L_1 ... L_{t-1} L_t = i).$$

Then the joint probability of the nugget sequence and the most likely state sequence can be estimated as

$$\max_{L_1 ... L_T} P(N_1 ... N_T, L_1 ... L_T) = \max_{i \in \{0,1\}} V_T(i).$$

We can compute this probability using dynamic programming as follows:

$$V_1(i) = P(N_1, L_1 = i) \sim P(L_1 = i|N_1), \tag{8.3}$$
$$V_t(i) = \left(\max_{j \in \{0,1\}} V_{t-1}(j) P(L_t = i|L_{t-1} = j, B_t) \right) P(N_t|L_t = i) \tag{8.4}$$
$$\text{for } t \in \{2, ..., T\}.$$

The most likely state sequence can be determined by keeping track of the value of $j \in \{0, 1\}$ that maximizes the right-hand side of Formula 7.4 in each iteration.

The merging component of the SE system ranks text nuggets by relevance scores and constructs a pseudo-document from the most relevant nuggets. Here, we can use the probability that a particular nugget is relevant given the sequence of all nuggets

as a relevance estimate. More formally, we estimate for each nugget N_t the marginal probability $P(L_t = 1|N_1...N_T)$. For $t \in \{1, ..., T\}$ and $i \in \{0, 1\}$, let

$$\alpha_t(i) := P(N_1...N_t, L_t = i),$$
$$\beta_t(i) := P(N_{t+1}...N_T|L_t = i).$$

Then the marginal distribution of the t-th label L_t given the nugget sequence $N_1...N_T$ can be estimated as follows:

$$\begin{aligned} P(L_t = i|N_1...N_T) &= \frac{P(N_1...N_T, L_t = i)}{P(N_1...N_T)} \\ &= \frac{P(N_1...N_T, L_t = i)}{\sum_{j\in\{0,1\}} P(N_1...N_T, L_t = j)} \\ &= \frac{P(N_1...N_t, L_t = i)P(N_{t+1}...N_T|L_t = i)}{\sum_{j\in\{0,1\}} P(N_1...N_t, L_t = j)P(N_{t+1}...N_T|L_t = j)} \\ &= \frac{\alpha_t(i)\beta_t(i)}{\sum_{j\in\{0,1\}} \alpha_t(j)\beta_t(j).} \end{aligned}$$

To compute $\alpha_t(i)$ and $\beta_t(i)$, we can again use dynamic programming:

$$\alpha_1(i) = P(N_1, L_1 = i) \sim P(L_1 = i|N_1), \tag{8.5}$$

$$\alpha_t(i) = \left(\sum_{j\in\{0,1\}} \alpha_{t-1}(j)P(L_t = i|L_{t-1} = j, B_t) \right) P(N_t|L_t = i) \tag{8.6}$$

$$\text{for } t \in \{2, ..., T\},$$

$$\beta_T(i) = 1, \tag{8.7}$$

$$\beta_t(i) = \sum_{j\in\{0,1\}} P(L_{t+1} = j|L_t = i, B_{t+1})P(N_{t+1}|L_{t+1} = j)\beta_{t+1}(j) \tag{8.8}$$

$$\text{for } t \in \{1, ..., T - 1\}.$$

8.2.3 Experimental Setup

The graphical model was evaluated and compared to linear relevance models on the dataset introduced in Section 5.1. For each approach, we performed a 12-fold cross-validation over the topics marked as *CV* in Table 5.1. The text nuggets for each topic were assigned to a different fold to avoid training and testing on similar instances or even duplicates. The nuggets were ranked by their relevance estimates in descending order, and the rankings were evaluated using the same performance metrics as in the evaluation of the baselines and linear models in Chapter 5: mean average precision (MAP) and precision-recall curves.

We present evaluation results on both sentence-length text nuggets (*Sentence*) and paragraph-length nuggets delimited by structural HTML markup (*Markup*) for

Model	Sentence MAP	Markup MAP
LR Independent	71.95%	79.69%
LR Adjacent	77.19%	80.59%
Sequential	78.24%	80.71%
Interpolated	**78.81%**	**80.93%**

Table 8.3: MAP of linear and sequential relevance models.

the independent logistic regression model (*LR Independent*), the logistic regression model with features of adjacent instances (*LR Adjacent*), and the following additional relevance models:

- *Sequential.* Graphical model combining relevance and transition features, introduced in Section 8.2.2.
- *Interpolated.* Linear interpolation of *Sequential* and *LR Adjacent*, giving weight α to the sequential model and weight $1 - \alpha$ to the logistic regression model. In each step of the cross-validation, we tuned α by maximizing MAP on the training folds. The average interpolation weight was $\alpha = 0.99$ for sentence-level nuggets and $\alpha = 0.57$ for markup-based nuggets.

8.2.4 Results and Analysis

In Table 8.3 we show MAP scores for each of the statistical relevance models and for both types of text nuggets. Precision-recall curves for the best-performing linear model *LR Adjacent* and the sequential model are given in Figure 8.8. The sequential model has a higher MAP than the logistic regression models on both sentence-level text nuggets and markup-based nuggets, and the precision-recall curves illustrate that the sequential model also has higher precision than the best linear model at all but very low recall levels. It can further be seen that effective modeling of dependencies between nearby text passages is more important when ranking smaller units of text. The performance gains from using a sequential model are quite noticeable on the shorter sentence nuggets but very small on the longer paragraph nuggets.

The interpolated model only slightly outperforms the individual models in terms of MAP. The precision-recall curves are similar to the ones for the sequential model and were omitted in the plot for ease of presentation. When tuning the interpolation weight on sentence-level nuggets, the sequential model is given most of the weight ($\alpha = 0.99$) since it is more effective and the logistic regression model does not seem to add much new information. On markup-based nuggets, the two models do not only perform similarly and are given similar weights in the linear combination ($\alpha = 0.57$), but their predictions also seem to be highly correlated since the interpolation hardly improves performance.

We performed one-sided Wilcoxon signed-rank tests to determine whether the improvements in nugget ranking performance are statistically significant. In Section 5.3 we described the setup of these significance tests. Table 8.4 shows p-values for each

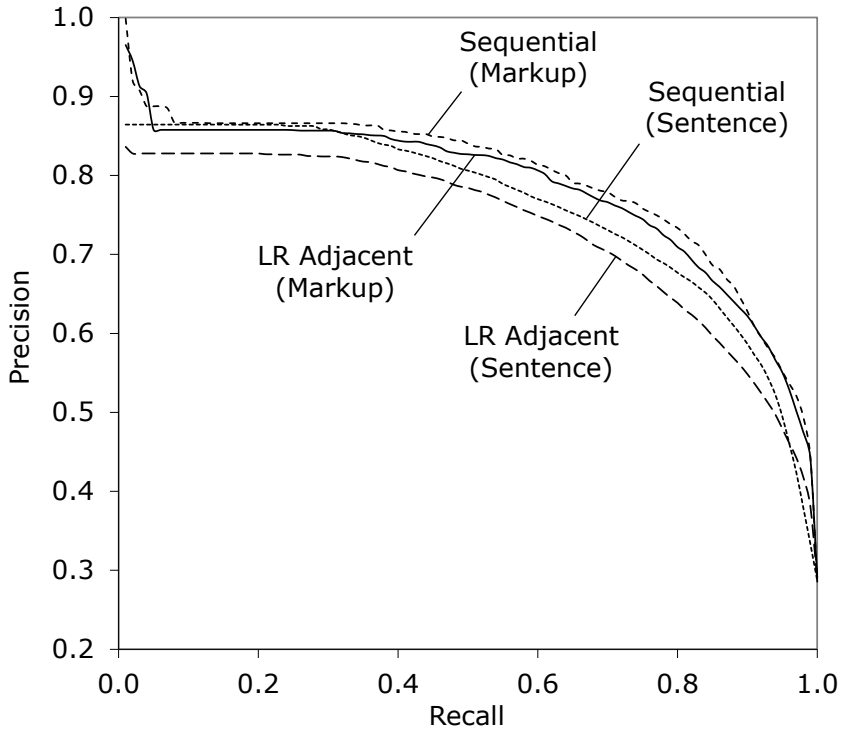


Figure 8.8: Precision-recall curves for linear and sequential relevance models.

pair of relevance models on sentence-level nuggets and markup-based nuggets. It can be seen that the models that take dependencies between text nuggets into account significantly outperform the independent LR model (at the 1% level) on sentence-length nuggets, and there is also evidence that these methods are more effective on markup nuggets. In contrast, the improvements from using a sequential model or an interpolated model instead of logistic regression with adjacent features are mostly not significant. This is because these methods did not consistently improve the rankings for the 12 topics on which the significance tests are based, but they helped for some topics and hurt (to a lesser extent) in other cases. However, one should keep in mind that we compared rankings using average precision, which is most sensitive to the highest-ranked text nuggets. The outcome of the signicance tests can be different if another metric is used, such as precision or recall at a fixed cutoff point.

Both the logistic regression models and the sequential model perform worse on sentences than on markup-based text nuggets. The lower performance of the sequential model on the shorter sentence nuggets is counterintuitive since the transition models include a binary feature that indicates whether a nugget boundary is also a markup boundary. Thus the sequential model could assign very low probabilities to transitions at sentence boundaries that do not coincide with structural markup. However, some of the features used for relevance estimation, such as cosine similarities, likelihood ratios, and the overlap of a nugget with the search engine abstract, are less suitable for shorter nuggets. In Section 5.3 we have shown that each of these relevance features, when used independently to rank text nuggets, is more effective

	LR Indep.	LR Adjacent	Sequential
LR Adjacent	S: 2.44e-4 M: 0.0134	–	–
Sequential	S: 2.44e-4 M: 0.0647	S: 0.0881 M: 0.367	–
Interpolated	S: 2.44e-4 M: 0.0171	S: 7.32e-4 M: 0.0881	S: 0.170 M: 0.212

Table 8.4: P-values for all pairs of linear and sequential relevance models, based on a one-sided Wilcoxon signed-rank test. The null hypothesis is that the ranking strategy to the left is as effective as the strategy at the top, the alternative is that it is more effective. We show p-values for both sentence-level text nuggets (S) and markup-based text nuggets (M).

for paragraphs than for sentences. This can be seen clearly when comparing the MAP scores in Figures 5.2 and 5.3. Thus the degradation in ranking performance can be attributed to the relevance model rather than the transition models.

The logistic regression model with features of adjacent text nuggets is already a very strong baseline for relevance estimation with MAP scores of over 80%, and the relevance features were specifically engineered and selected based on their impact on a linear model. It should further be noted that the theoretical upper bound on MAP in our experimental setup is not 100%. This is because some text nuggets are only partially relevant and it is therefore not possible to perfectly separate relevant from irrelevant text. Consider a simple example with two text nuggets, each consisting of one relevant token followed by one irrelevant token. No matter how the two nuggets are ranked, the relevant tokens are always separated by an irrelevant token. Since we compute precision at each relevant token (cf. Section 5.2) both possible rankings have a MAP score of $\frac{1}{2}\left(1 + \frac{2}{3}\right) = 83\%$. How closely we can approach 100% MAP in practice depends on the granularity of the text nuggets, with the shorter sentence-length nuggets leaving more headroom than the longer markup-based nuggets.

When scoring markup-based nuggets, the small improvement in MAP may not justify the additional implementation effort for the sequential relevance model. For instance, it seems unlikely that the sequential model with the current feature set would significantly improve question answering performance compared to the logistic regression model that includes features of adjacent instances. However, the sequential model could easily be extended with additional or improved transition features, and further performance gains over logistic regression seem possible, particularly when ranking smaller units of text. Furthermore, while the sequential model is conceptually more complex than *LR Adjacent*, it actually has fewer degrees of freedom and is therefore less prone to overfitting. *LR Adjacent* uses 55 relevance features, including features of adjacent instances. Thus we must optimize 55 weights and an intercept, or 56 parameters in total, if we do not perform feature selection. The sequential model, on the other hand, uses only the original 19 relevance features and 8 transition fea-

tures. Thus it comprises one relevance model with 20 parameters and two transition models with 9 parameters each (including intercepts). In total, the sequential model has only 38 parameters.

In order to be usable for large-scale source expansion, a relevance model must be extremely efficient at runtime so that it can be applied to billions of text nuggets. The sequential relevance model meets this requirement since the time complexity of the Viterbi algorithm and the computation of marginal probabilities is linear in the number of text nuggets and features. Compared to the logistic regression model with features of adjacent instances, the sequential model has the advantage of requiring less time for training since it uses fewer features in its relevance component (19 instead of 55 features). The training time for the sequential model was about 2 minutes for paragraph-length text nuggets and 3 minutes for sentence-length nuggets when using all annotated data. In contrast, it took about 45 minutes to fit a logistic regression model with adjacent features to paragraph nuggets and 68 minutes to fit an LR model to sentence nuggets. Note that it takes longer to train a model on sentences since there are more sentences than paragraphs in the dataset. The runtimes were measured on a server with 3 GHz Xeon CPUs and 32 GB RAM.

Chapter 9

Conclusions

In Section 9.1 we summarize our approach and experimental results,[1] and in Section 9.2 we discuss why we believe that source expansion is an important area of QA research and in which situations this approach can be applied. Promising directions for future research are suggested in Section 9.3.

9.1 Summary

We proposed a statistical approach for source expansion (SE) and implemented an end-to-end system that extends a collection of seed documents with related information from large, external text corpora. For each seed document, our system (1) retrieves related documents from other resources such as the Web, (2) extracts self-contained text nuggets from the retrieved documents, (3) estimates the relevance of those nuggets with a statistical model, and (4) compiles a new pseudo-document from the most relevant content, avoiding lexically redundant text. This SE process augments the seed corpus with new related information, and it adds reformulations of information that was already covered in the seeds. The additional content can benefit information retrieval or extraction applications that require large, redundant and high-quality textual resources. Our implementation of the SE method is robust and efficient, enabling us to expand hundreds of thousands of seed documents that vary widely in length, style and topics with content extracted from millions of web pages. The expanded text corpora were easily integrated into Watson and OpenEphyra, two QA systems that are based on very different architectures and that leverage their information sources in different ways.

The key component of the SE approach is a statistical model for the relevance estimation task (step 3). We have seen that a large amount of training data can be labeled for this task by using an efficient annotation interface, and that high inter-annotator agreement can be achieved if a detailed set of guidelines is followed. This annotation methodology was applied to create a dataset of over 160,000 hand-labeled text nuggets, and different methods for ranking text by relevance were evaluated and

[1] For an overview of the source expansion approach and our key findings on QA datasets, please also refer to Schlaefer et al. [2011].

compared on this data. The best-performing statistical models combine various topicality features, search-based features and surface features of the retrieved text to estimate the relevance and textual quality of text nuggets and achieve MAP scores of about 81%. In comparison, the maximal marginal relevance algorithm ranks nuggets with 75% MAP, and the rankings generated by the Yahoo! search engine have 43% MAP. These results confirm that a statistical relevance model is generally more effective than a single-strategy ranking method. The amount of labeled training data required to fit statistical models can be reduced drastically with an active learning approach. We found that 1,000 instances selected based on a combination of diversity and uncertainty sampling criteria are sufficient to train a model with high ranking performance. In evaluations on question answering datasets, the effectiveness of this model comes close to a supervised model fitted to more than a hundred times more training data. We have also seen that relevant text nuggets are often surrounded by more relevant text, and that the predictive performance of a statistical model can be improved by leveraging the context of nuggets in their source documents. When ranking paragraph-length nuggets, most of this potential can be realized by extending a linear model with features of adjacent instances. However, when scoring individual sentences to allow for a more fine-grained selection of relevant content, a sequential model that predicts transitions between relevant and irrelevant text using lexical coherence features further improves performance.

The statistical source expansion approach was applied to the question answering (QA) task, and its impact on the performance of Watson and OpenEphyra was evaluated on large datasets of Jeopardy! and TREC questions. We used encyclopedias and a dictionary as seed corpora and developed strategies for selecting the most useful seed documents based on popularity estimates. The chosen seeds were expanded with related content from web pages retrieved with a search engine. This expansion method consistently improved QA performance on every (reasonably large) dataset we experimented with, independently of the QA system, seed corpus and retrieval strategy used. When we evaluated the impact on QA search results, text passages and document titles were retrieved from the original sources and the expanded content using the open-source IR systems Indri and Lucene. The SE approach significantly improved search performance on both Jeopardy! and TREC datasets, yielding gains of 4.2–8.6% in search recall. We also found that source expansion hurts relatively few questions and tends to be more robust than query expansion techniques. In end-to-end QA experiments with Watson, the expanded text corpora were also used to retrieve supporting passages for answer scoring, and the added source content had an impact on features that are based on search rankings and corpus statistics. Again, the SE method significantly improved QA performance, increasing Watson's accuracy by 7.6–12.9%. The gains in accuracy are larger than the improvements in search recall, which indicates that source expansion facilitates answer scoring by increasing the semantic redundancy of the information sources.

We have also shown that the SE approach does not require a search engine to pre-select relevant content from the Web or other large document collections. Our method is equally effective if text that is related to the topics of seed documents is extracted directly from a locally stored corpus without using a retrieval system, and

the search-based and extraction-based methods can be combined to further improve QA results. By expanding seed corpora with related content from both web search results and a local web crawl, we were able to improve Watson's QA accuracy by 9.4–19.8% on Jeopardy! and TREC datasets. In addition, we proposed an extension of the SE algorithm to seed corpora in which there exists no injective mapping between documents and topics that can be expanded. For instance, in newswire sources and web crawls, multiple topics can be covered in a single document, or the same topic may be discussed repeatedly in different documents. Such sources can be transformed into topic-oriented document collections by discovering the most important topics and building pseudo-documents about them.

Note that the improvements in question answering performance were achieved *without* retrieving more text at QA runtime. Search and candidate recall can always be increased by retrieving additional or longer search results, but this often comes at the expense of adding noise that hurts answer selection efficiency and effectiveness. It may also seem that QA results always improve if larger information sources are used, but we have seen that this is not always the case. For instance, just adding large, raw web crawls is ineffective because the vast majority of web pages do not contain useful information [Clarke et al., 2002], and even if a QA system could handle the added noise, expensive parallel hardware would be required to achieve fast response times. In addition, the proposed source expansion method outperforms other reasonable strategies for gathering additional source content. The popularity-based seed selection approach helps target topics that are relevant for QA, and the expanded documents for the most popular seeds clearly have the largest impact on search recall. We have also shown that QA search performance can degrade if too many seeds are expanded. Furthermore, even if the most relevant topics are identified, an effective model is needed to accurately estimate the relevance of related text and avoid adding noise. Our statistical relevance models outperform several strong baselines, such as search engine rankings and a multi-document summarization algorithm, both in an intrinsic evaluation and when applied to the QA task. In preliminary experiments, we also attempted compiling larger expanded sources that include text with lower relevance estimates, but this did not further improve QA search recall.

When interpreting the QA results, one should keep in mind that 100% search recall or accuracy are unattainable. In practice, the headroom is much smaller because the answer keys used to evaluate search and answer extraction performance are incomplete, and the answer selection algorithms deployed in QA systems are imperfect. In addition, the Jeopardy! datasets contain puzzles, word plays and puns that are often difficult to resolve programmatically or require specialized inference. Some Jeopardy! questions also seek answers that cannot be retrieved from existing information sources but must be derived in an additional processing step (e.g. *YOU DO THE MATH: The number of legs on a spider plus the number of legs on a fly*). TREC questions often ask for obscure facts that are unlikely to be found even when using a web search engine at QA runtime (e.g. *How much could you rent a Volkswagen bug for in 1966?*). With source expansion, Watson achieves 90.11% search recall and 72.32% accuracy on a large, blind test set of regular Jeopardy! questions. These results are hard to beat without overfitting to the test data.

9.2 Importance of Source Expansion

It is usually possible to make large improvements early in the development of a QA system (or other applications of language technologies), 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 and the total headroom for improvements is limited. Yet we found that the impact of our source expansion method did not diminish as Watson was improved over time because advances in other system 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 algorithms, 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 increase the performance of other QA systems even if they are already well tuned for their tasks. In addition, we expect that SE will be useful for other natural language processing applications that require large amounts of relevant information in a given knowledge domain, and we will give examples of tasks that may benefit from our method in Section 9.3.

Considering the significant and consistent performance impact of source expansion, it seems surprising that not more research has been devoted to the selection or generation of relevant source material for QA. One explanation is that most QA systems were developed for evaluation efforts in which reference corpora were provided to the participants. For example, in TREC evaluations each answer had to be supported with a document in a given newswire source or a blog corpus [Dang et al., 2007]. Additional resources could be leveraged to identify and score candidates, but the final answer had to be found in the reference sources to facilitate its verification by human assessors. In real-world applications, however, it usually does not matter to the user which information sources are used by a QA system, as long as questions are answered with high precision and recall. It is also possible that source acquisition has not been the focus of attention because it was thought of as a manual process that offers few opportunities for scientific work. Instead, many publications by QA researchers were concerned with algorithms for effective question analysis, retrieval, answer extraction and scoring. We hope to have shown in this thesis that statistical source expansion also involves a variety of interesting research problems in the fields of machine learning, information retrieval, and natural language processing.

While our implementation of the source expansion approach is quite comprehensive, it may not be necessary to fully replicate it in order to achieve similar results. Most of the development effort was invested in the component that estimates the relevance of text using a statistical model. For this purpose we created a large dataset of over 160,000 hand-labeled text nuggets, and we developed a diverse set of 27 relevance and transition features. Fortunately, instead of fitting a statistical model, one could get started quickly by using a single topicality feature to rank text by relevance, such as a likelihood ratio estimated with topic and background language models or cosine similarities. We have shown that the relevance estimation performance of these features comes reasonably close to the full model if the seed documents are long and

of high quality, and the cosine similarity baseline also yields large gains in QA search performance when used to expand Wikipedia articles. Apart from not having to implement the other relevance features, a single-strategy approach has the advantage of not requiring a labeled dataset to estimate feature weights. However, we have also seen that the topicality features are only effective if enough relevant seed content is available, and that the cosine similarity approach is less suitable for expanding Wiktionary entries. If the seeds are short or noisy, we recommend using a more robust relevance model that also leverages search-based features and surface characteristics of the extracted text. In that case, it is unavoidable to implement a larger feature set and to fit a model to hand-labeled data, but we can reduce the manual annotation costs considerably by using active learning instead of training a supervised model.

Our method is broadly applicable since for many knowledge domains suitable seed corpora are available. For instance, there exist encyclopedias and dictionaries that could be used as a starting point when developing systems for medical, legal or financial QA tasks. The seed corpus should at least touch on the topics that are central to a given domain, but it could be quite small and important information may be missing or hard to find. Preferably the seeds should be topic-oriented, but we have also suggested an approach for compiling pseudo-documents about topics discovered in an unstructured source. Note that the seeds may be about any topics that can be described in a query for search-based SE, or that can be looked up in a local source when using the extraction-based method. For example, a topic could be a named entity (e.g. *United Nations*) or a technical term (e.g. *volatility*), but it could also be a specific aspect of a broader topic (e.g. *corruption in developing countries*) or an event involving multiple entities (e.g. *Israeli-Palestinian peace talks*). However, the documents should not cover multiple independent topics since that would render our approach for retrieving related text and comparing it to seeds less effective.

We also expect that it does not matter if the domain is broad or narrow, as long as an external resource exists in which related content can be found. The Web provides information about most general interest topics as well as many specialized knowledge domains, including various areas of entertainment, global and regional news, technology, politics, scientific research, and even the most esoteric interests and hobbies. Local corpora may be used as a source of proprietary or confidential information (e.g. about the employees and products of a company or intellectual property) or content that is not freely available on the Web (e.g. much of the financial literature). Once seed documents have been chosen and external sources have been identified, the expansion can be performed without further manual intervention. Thus our approach can be applied to rapidly grow source content for existing applications, and it can also facilitate the adaptation of NLP systems to new domains for which the available sources have insufficient coverage or lack redundancy.

9.3 Future Research

We have demonstrated that source expansion is effective for topic-oriented seed corpora such as encyclopedias or dictionaries, but QA systems can also use a variety of

other resources that are not organized by topics, including web data and newswire corpora. If these sources are also expanded with related information, this may significantly increase the impact of our approach on QA performance and its applicability to new domains and tasks. While we proposed a method for transforming unstructured sources into seed corpora that can be used as input for the current SE algorithm, its implementation and evaluation is left as a promising direction for future research. For instance, this method could be applied to a newswire source such as the AQUAINT corpus[2], a collection of about 1 million news articles that was the reference corpus in TREC 11–15. News corpora are valuable sources of information about entities and events that have been at the center of attention in a given time period, and existing TREC datasets could be used to evaluate whether a SE system can find additional relevant content about the topics in the AQUAINT corpus.

Source expansion improves QA performance both by increasing the semantic redundancy of the knowledge sources and by adding new information that was not already covered in the sources. The ideal tradeoff between the two types of related information depends on the available text corpora and the QA task, and on how the expanded content is used. For instance, one should focus on adding redundant data if the current text corpus already has high coverage or if the source of related content is too unreliable to trust information that is not in the seeds, but it may be more effective to increase the coverage if the seed corpus is small or the QA system has low search recall. Furthermore, one could use a source with high coverage when extracting candidate answers, and a different corpus with additional reformulations of the same information to retrieve supporting evidence for answer scoring. This raises the question of how the SE algorithm can be adjusted to control the mix of redundant and new content.

The current relevance estimation approach is biased towards redundant information because it selects text nuggets that are similar to the seed documents, but we already noted that the size of the nuggets could be increased to pick up more new information along with the redundant content. In addition, instead of estimating the topicality of text nuggets using only the seeds, one could train adaptive topic models that are updated with related text that has high relevance probabilities based on an initial model. The amount of new information could also be increased by performing additional searches for related content using query terms extracted from the retrieved text, thus allowing the focus to shift towards aspects of a topic that are not already covered in the seed. Finally, one could cluster the extracted text nuggets and compute topicality features based on the clustering, such as the proximity of a nugget to the nearest centroid. This may increase the diversity of the information in the expanded documents and may help select nuggets about more obscure aspects of a topic that are not mentioned frequently.

We expect that the SE method can be adapted with relative ease to different information sources and new knowledge domains. Related text can be retrieved from the Web or other sources with a search engine, or it can be extracted directly from a local corpus. While we used HTML markup to split web pages into text nuggets, one

[2] `http://www.ldc.upenn.edu/Catalog/docs/LDC2002T31/`

could also extract paragraphs or coherent passages from documents in other formats. The relevance model in the scoring phase mostly relies on features that are based on only the seed document and the text nuggets, which can be computed for text extracted from arbitrary sources. We also used features that leverage Yahoo! search rankings and summary snippets, but similar features could be derived from the search results of other retrieval systems, and the statistical model in the experiments with the local web crawl was effective even without search-related features. The relevance model can also be extended easily with additional features that are available for new sources, such as page ranks or spam scores for a web crawl or features derived from metadata in a newswire corpus. We have shown that a new statistical model can be fitted in approximately one day using active learning. However, one could also adapt an existing relevance model that was trained on similar data to a new domain by applying transfer learning techniques. For example, relevance probabilities estimated with the existing model can be used as one of the features when fitting a new model to a different dataset. This transfer learning method integrates quite naturally with active learning. At first we can select queries the original relevance model is uncertain about, i.e. instances that are close to the old decision boundary. Once an initial set of positive and negative instances has been labeled, new models can be trained that combine the estimates from the existing model with other features, and subsequent queries can be selected based on the new decision boundaries.

Since the source expansion approach in its current form does not rely on deep natural language processing techniques, it could also be adapted to different languages with relatively minor modifications. For example, there exist versions of Wikipedia and Wiktionary in many other languages that could be expanded with content from other websites using the same methodology. The pipeline stages for retrieving web pages and extracting text nuggets based on structural markup can be applied to any language for which a search engine is available. A new relevance model can be fitted using the same learning methods as in the experiments with English sources, but some of the features may need to be removed or adapted to the new language. For instance, the topicality features are based on language models and term weighting schemes that were chosen specifically because of their effectiveness for English text, and the surface feature that measures the ratio of capital letters in a text nugget is not applicable to Asian languages. In the merging phase, the thresholds that are used to remove lexically redundant text and to control the length of the generated pseudo-documents may be adjusted to achieve optimal results. For example, more lenient thresholds may be chosen if the relevance estimates are less accurate or if the sources of related content are small. The only language-specific resources are a tokenizer and a list of function words for stopword removal, which are also available for most other languages.

While we focused on evaluating the impact of source expansion on factoid QA results, our approach can also be applied to other types of questions with more complex answers. For instance, the expanded sources could be used to extract and validate answers for list questions (e.g. *Which U.S. presidents were assassinated?*) and pseudo-documents about popular topics could also be helpful for answering definitional questions (e.g. *What is the XYZ Affair?*). In addition, the SE method can be

adapted to other information retrieval and extraction applications that benefit from large, redundant information sources. For example, the performance of a document retrieval system may be improved by expanding the documents and augmenting their index representations with related text. Our approach could also be applied to gather additional relevant content for machine reading (MR). Given a text corpus about a specific domain (e.g. *National Football League*), an MR system automatically extracts instances of a pre-defined set of relations (e.g. *players* and their *teams*). The relation extraction algorithm may be more effective if the corpus is expanded with text that contains additional instances of these relations, or additional evidence for instances that already occur in the original source. As seed topics, one could use entities that participate in relations for which the system currently has low precision or recall. Related content may be extracted from web search results, crawls of relevant websites, or domain-specific local sources.

Similarly, source expansion may improve the performance of algorithms for extracting surface patterns that express a given semantic relationship between entities from a text corpus [Ravichandran and Hovy, 2002, Pantel and Pennacchiotti, 2006]. For instance, the patterns "PERSON (DATE –" and "PERSON was born on DATE" both relate people to their dates of birth but have very different surface forms. The learned patterns can be used for relation extraction or to measure the semantic similarity between text passages. In QA systems, the patterns could be applied to generate candidate answers and to score candidates by comparing their supporting passages to the question. The effectiveness of a pattern learning algorithm may be improved by expanding the source corpus with additional relevant text. The new content must be related to the information in the original source in order to contain instances of the same entities and patterns, and it should be of high textual quality. The statistical relevance models used in the SE approach are ideal for identifying text that meets these two criteria because they combine topicality features that measure the similarity of related text to seed content with surface features to select only well-formed text. If patterns are learned for a QA system, the same seed corpora and expanded sources can be used to extract the patterns and to answer questions. This would help ensure that the patterns frequently occur in the QA search results and that they express the relationships they are intended for.

Bibliography

A. Agresti. *Categorical Data Analysis*. Wiley, 2002.

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.

H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.

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.

N. Balasubramanian and S. Cucerzan. Automatic generation of topic pages using query-based aspect models. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM)*, 2009.

P. Banerjee and H. Han. Answer credibility: A language modeling approach to answer validation. In *Proceedings of NAACL HLT*, 2009.

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

B. Billerbeck and J. Zobel. Document expansion versus query expansion for ad-hoc retrieval. In *Proceedings of the 10th Australasian Document Computing Symposium*, 2005.

M. Bilotti, P. Ogilvie, J. Callan, and E. Nyberg. Structured retrieval for question answering. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2007.

M. W. Bilotti and E. Nyberg. Improving text retrieval precision and answer accuracy in question answering systems. In *Proceedings of the Second Information Retrieval for Question Answering (IR4QA) Workshop at COLING*, 2008.

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

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.

S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1-7):107–117, 1998.

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

J. Carbonell and J. Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1998.

J. Carletta. Assessing agreement on classification tasks: The kappa statistic. *Computational Linguistics*, 22(2):249–254, 1996.

C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. In *ACM Transactions on Intelligent Systems and Technology*, 2011.

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

J. Chu-Carroll, J. Fan, B. Boguraev, D. Carmel, D. Sheinwald, and C. Welty. Finding needles in the haystack: Search and candidate generation. *IBM Journal of Research and Development Special Issue on Watson in Jeopardy!*, 2012a.

J. Chu-Carroll, J. Fan, N. Schlaefer, and W. Zadrozny. Textual resource acquisition and engineering. *IBM Journal of Research and Development Special Issue on Watson in Jeopardy!*, 2012b.

C. Clarke, G. Cormack, and T. Lynam. Exploiting redundancy in question answering. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2001.

C. Clarke, G. Cormack, M. Laszlo, T. Lynam, and E. Terra. The impact of corpus size on question answering performance. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2002.

K. Collins-Thompson. Robust model estimation methods for information retrieval. *LTI Technical Report*, CMU-LTI-08-010, 2008.

K. Collins-Thompson and J. Callan. Estimation and use of uncertainty in pseudo-relevance feedback. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2007.

G. Cormack, M. Smucker, and C. Clarke. Efficient and effective spam filtering and re-ranking for large web datasets. In *Information Retrieval*, pages 1–25. Springer, 2011.

N. Craswell, D. Hawking, and S. Robertson. Effective site finding using link anchor information. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2001.

H. T. Dang, J. Lin, and D. Kelly. Overview of the TREC 2006 question answering track. In *Proceedings of the Fifteenth Text REtrieval Conference*, 2006.

H. T. Dang, D. Kelly, and J. Lin. Overview of the TREC 2007 question answering track. In *Proceedings of the Sixteenth Text REtrieval Conference*, 2007.

K. Darwish and D. W. Oard. CLIR experiments at Maryland for TREC-2002: Evidence combination for Arabic-English retrieval. In *Proceedings of the Eleventh Text REtrieval Conference*, 2002.

P. Donmez, J. G. Carbonell, and P. N. Bennett. Dual strategy active learning. In *Proceedings of the European Conference on Machine Learning (ECML)*, 2007.

I. Dornescu, G. Puşcaşu, and C. Orăsan. University of Wolverhampton at CLEF 2008. In *Working Notes for the Cross Language Evaluation Forum (CLEF)*, 2008.

S. Dumais, M. Banko, E. Brill, J. Lin, and A. Ng. Web question answering: Is more always better? In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2002.

C. Fellbaum. *WordNet: An Electronic Lexical Database*. The MIT Press, 1998.

D. Ferrucci, E. Nyberg, J. Allan, K. Barker, E. Brown, J. Chu-Carroll, A. Ciccolo, P. Duboue, J. Fan, D. Gondek, E. Hovy, B. Katz, A. Lally, M. McCord, P. Morarescu, B. Murdock, B. Porter, J. Prager, T. Strzalkowski, C. Welty, and W. Zadrozny. Towards the open advancement of question answering systems. *IBM Technical Report*, RC24789, 2009.

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.

J. L. Fleiss. *Statistical Methods for Rates and Proportions (2nd Edition)*. Wiley, 1981.

J. S. Garofolo, E. M. Voorhees, C. G. P. Auzanne, V. M. Stanford, and B. A. Lund. 1998 TREC-7 spoken document retrieval track overview and results. In *Proceedings of the Seventh Text REtrieval Conference*, 1998.

J. Goldstein, M. Kantrowitz, V. Mittal, and J. Carbonell. Summarizing text documents: Sentence selection and evaluation metrics. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1999.

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.

I. J. Good. The population frequencies of species and the estimation of population parameters. *Biometrika*, 40(3-4):237–264, 1953.

S. Gopal and Y. Yang. Multilabel classification with meta-level features. In *Proceedings of the 33rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2010.

D. Graff, C. Cieri, S. Strassel, and N. Martey. The TDT-3 text and speech corpus. In *Proceedings of the DARPA Broadcast News Workshop*, 1999.

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.

D. Harman. Overview of the TREC 2002 novelty track. In *Proceedings of the Eleventh Text REtrieval Conference*, 2002.

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

A. Hickl, K. Roberts, B. Rink, J. Bensley, T. Jungen, Y. Shi, and J. Williams. Question answering with LCC's CHAUCER-2 at TREC 2007. In *Proceedings of the Sixteenth Text REtrieval Conference*, 2007.

S. E. Johnson, P. Jourlin, K. Spärck Jones, and P. C. Woodland. Spoken document retrieval for TREC-8 at Cambridge University. In *Proceedings of the Eighth Text REtrieval Conference*, 1999.

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

M. Kaisser, S. Scheible, and B. Webber. Experiments at the University of Edinburgh for the TREC 2006 QA track. In *Proceedings of the Fifteenth Text REtrieval Conference*, 2006.

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.

J. Ko, L. Si, and E. Nyberg. Combining evidence with a probabilistic framework for answer ranking and answer merging in question answering. In *Information Processing & Management*, 2010.

J. R. Landis and G. G. Koch. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174, 1977.

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.

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 (MSDR)*, 2003.

B. Magnini, M. Negri, R. Prevete, and H. Tanev. Comparing statistical and content-based techniques for answer validation on the Web. In *Proceedings of the VIII Convegno AI*IA*, 2002.

C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

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

O. A. McBryan. GENVL and WWWW: Tools for taming the Web. In *Proceedings of the First International World Wide Web Conference*, 1994.

A. McCallum and K. Nigam. Employing EM and pool-based active learning for text classification. In *Proceedings of the Fifteenth International Conference on Machine Learning*, 1998.

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

T. M. Mitchell. Generalization as search. *Artificial Intelligence*, 18(2):203–226, 1982.

A. Nenkova, L. Vanderwende, and K. McKeown. A compositional context sensitive multi-document summarizer: Exploring the factors that influence summarization. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2006.

D. W. Oard and F. C. Gey. The TREC-2002 Arabic/English CLIR track. In *Proceedings of the Eleventh Text REtrieval Conference*, 2002.

J. O'Connor. Citing statements: Computer recognition and use to improve retrieval. *Information Processing & Management*, 18(3):125–131, 1982.

B. Pang and L. Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd ACL Conference*, 2004.

P. Pantel and M. Pennacchiotti. Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics (COLING/ACL)*, 2006.

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.

M. F. Porter. An algorithm for suffix stripping. *Program, 14(3):130-137*, 1980.

X. Qiu, B. Li, C. Shen, L. Wu, X. Huang, and Y. Zhou. FDUQA on TREC 2007 QA track. In *Proceedings of the Sixteenth Text REtrieval Conference*, 2007.

L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

D. Ravichandran and E. Hovy. Learning surface text patterns for a question answering system. In *Proceedings of the 40th ACL Conference*, 2002.

J. Rocchio. Relevance feedback in information retrieval. *The SMART Retrieval System: Experiments in Automatic Document Processing*, pages 313–323, 1971.

N. Schlaefer and J. Chu-Carroll. Question answering (book chapter). In *Multilingual Natural Language Processing Applications: From Theory to Practice, Eds. D. Bikel and I. Zitouni*. Prentice Hall, 2012.

N. Schlaefer, P. Gieselmann, and G. Sautter. The Ephyra QA system at TREC 2006. In *Proceedings of the Fifteenth Text REtrieval Conference*, 2006.

N. Schlaefer, J. Ko, J. Betteridge, G. Sautter, M. Pathak, and E. Nyberg. Semantic extensions of the Ephyra QA system in TREC 2007. In *Proceedings of the Sixteenth Text REtrieval Conference*, 2007.

N. Schlaefer, J. Chu-Carroll, E. Nyberg, J. Fan, W. Zadrozny, and D. Ferrucci. Statistical source expansion for question answering. In *Proceedings of the 20th ACM Conference on Information and Knowledge Management (CIKM)*, 2011.

D. Shen, M. Wiegand, A. Merkel, S. Kazalski, S. Hunsicker, J. L. Leidner, and D. Klakow. The Alyssa system at TREC QA 2007: Do we need Blog06? In *Proceedings of the Sixteenth Text REtrieval Conference*, 2007.

S. Siegel. *Nonparametric Statistics for the Behavioral Sciences*. McGraw-Hill, 1956.

A. Singhal and F. Pereira. Document expansion for speech retrieval. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1999.

A. Singhal, J. Choi, D. Hindle, D. D. Lewis, and F. Pereira. AT&T at TREC-7. In *Proceedings of the Seventh Text REtrieval Conference*, 1998.

A. Singhal, S. Abney, M. Bacchiani, M. Collins, D. Hindle, and F. Pereira. AT&T at TREC-8. In *Proceedings of the Eighth Text REtrieval Conference*, 1999.

T. Tao, X. Wang, Q. Mei, and C. Zhai. Language model information retrieval with document expansion. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL*, 2006.

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

S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, 2:45–66, 2001.

M. van Zaanen. Multi-lingual question answering using OpenEphyra. In *Working Notes for the Cross Language Evaluation Forum (CLEF)*, 2008.

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

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

J. Wiebe and E. Riloff. Creating subjective and objective sentence classifiers from unannotated texts. In *Proceedings of the 6th International Conference on Intelligent Text Processing and Computational Linguistics*, 2005.

F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6): 80–83, 1945.

M. Zhang, R. Song, C. Lin, S. Ma, Z. Jiang, Y. Jin, Y. Liu, and L. Zhao. Expansion-based technologies in finding relevant and new information: THU TREC2002 novelty track experiments. In *Proceedings of the Eleventh Text REtrieval Conference*, 2002.