ASPECTS OF SENTENCE RETRIEVAL

A Dissertation Presented

by

VANESSA GRAHAM MURDOCK

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

September 2006

Computer Science

ASPECTS OF SENTENCE RETRIEVAL

A Dissertation Presented

by

VANESSA GRAHAM MURDOCK

Approved as to style and content by:

W. Bruce Croft, Chair

James Allan, Member

David Jensen, Member

John Staudenmayer, Member

W. Bruce Croft, Department Chair
Computer Science

I dedicate this thesis to my great-grandmother Mildred Graham who completed the sixth grade, taught her husband to read and put her son through law school, and to my great-grandmother Olive Peterson who completed a Masters in Mathematics, and put her daughter through the Sorbonne. My generation owes a debt to those who came before because they prepared the way for us, so that everything was easy, and we only had to work hard to succeed. We hope to return the favor by teaching our children to be strong and courageous, thereby making us worthy of the grace bestowed upon us.

ACKNOWLEDGMENTS

Nothing worthwhile was ever achieved in isolation. I can not claim to have written this thesis without the significant influence of others. I would like to thank my advisor W. Bruce Croft. It was invaluable to have the benefit of his more than 30 years experience in the field of Information Retrieval standing behind his advising of me. I am most grateful that he allowed me to work with him for five years, and to be a part of the best information retrieval research group in the world. I would like to thank James Allan for serving on my committee. James sets a tone in the lab of friendly cooperation, which makes it possible to get more work done, and makes the work itself more enjoyable. I would like to thank David Jensen for the counsel he has given me in the time I have been at the University of Massachusetts, which was especially kind of him, considering that I am not a member of his lab. I would like to thank John Staudenmayer for serving as the outside member on my committee, to ensure the process is fair, even though he was already very busy.

My academic career was supported in part by a grant from AT&T. I would like to thank the Fellowship program for their support of my work, both monetary and academic. Srinivas Bangalore was my mentor at AT&T, and deserves the credit for inspiring me as an undergraduate to pursue research and for his part in ensuring my success. I would also like to acknowledge Charles Thompson for taking an active interest in my academic career. Charles is a person to be greatly admired for the care he takes in fostering the careers of students in his own lab, and students in the AT&T Fellowship program.

I would like to thank Andrew McCallum for advising my synthesis project, and introducing "trust exercises" into the CIIR retreats. David Fisher deserves a thank

you for his contributions to the lab both in terms of providing programming support, and providing insightful comments on papers and presentations. Kate Moruzzi, Barbara Sutherland and Sharon Mallory deserve appreciation for creating an environment where research can be done with the minimum external stress. I would like to thank our systems administrator, Andre Gauthier both for his friendship, and his patient systems support.

I would like to thank Diane Kelly who I collaborated with starting in my first year. I learned a tremendous amount about how to do good research from Diane, and I would like to thank her for her contributions to our projects. I would also like to acknowledge Xiaojun Yuan for her contributions to the project, and Nicholas J. Belkin for his support and mentoring.

Early contributors to my success were Adele Howe at Colorado State University, who was willing to work with me on an undergraduate research project as part of an independent study. She provided my first exposure to the field of information retrieval, and I learned a lot from her about what is good science. James Bieman was willing to take me on as an undergraduate research assistant, and allowed me to be co-author on what became my first paper.

I would like to thank Jerod Weinman for our excellent conversations that I found completely inspiring. I very much appreciate the friendship and conversation of students in the CIIR: Hema Raghavan, Mark Smucker, Don Metzler, Fernando Diaz, Ron Bekkerman, Trevor Strohman, Ben Carterette, Courtney Wade, Nasreen Abdul-Jaleel, Xiaoyong Liu, Xiaoyan Li, Xing Wei, Giridhar Kumaran, Jiwoon Jeon, Yun Zhou, Ao Feng, Ramesh Nallapati, Jeremy Pickens, and Victor Lavrenko. I have had many interesting and useful conversations with each of these people. Each person in the CIIR comes with his own set of strengths so that the lab is woven into a strong and beautiful fabric.

I would like to give special mention to the people who did sentence-level relevance assessments for the work on Information Provenance in Chapter 7. Relevance assessments are a tedious, but necessary, evil. The people who suffered through it included David Fisher, Kate Moruzzi, Don Metzler, Yun Zhou, Trevor Strohman, Ron Bekkerman, Xiaoyong Liu, and Xing Wei.

I would like to thank my father, Craig Murdock, who is the definition of erudite, and who I admire for his courage, perseverance, and intelligence.

I would like to thank my mother, Chris Jorgensen, who taught us to be strong and independent, and treated us as if we already were the intelligent, competent people she expected us to be.

I would like to thank my stepfather Jim Jorgensen for treating me as if I was his own child, and whose sense of humor, and sense of human I admire.

I would like to acknowledge my brother <*CENSORED*> Murdock who doesn't like to have his name published, but who has been a constant inspiration for his dedication to being independent and original. My aunt Gwendolyn Murdock has also been a role model, setting a trend as the first of three women "Dr. Murdocks" in the family.

I owe perhaps the deepest debt of gratitude to my husband, Paul. It was only through his love and support, and tolerance of the chaos I introduce into our lives, that I was able to do anything. Our son Paulot is the person I admire most of all the people I know because of his immense capacity for empathy, courage, intelligence and humor. Nothing I do in my life will ever be as meaningful and as rewarding as being his mother, and I am sure I am not worthy of the task.

I would like to thank the funding agencies that supported me during my time at the Center for Intelligent Information Retrieval, although any opinions, findings and conclusions or recommendations expressed in this material are mine and do not necessarily reflect those of the sponsors. During my time as a graduate student I was

supported in part by the Center for Intelligent Information Retrieval, in part by the National Science Foundation under grant number IIS-9907018, in part by Advanced Research and Development Activity under contract number MDA904-01-C-0984, in part by SPAWARSYSCEN-SD grant numbers N66001-99-1-8912 and N66001-02-1-8903, and in part by the Defense Advanced Research Projects Agency (DARPA) under contract number HR0011-06-C-0023.

ABSTRACT

ASPECTS OF SENTENCE RETRIEVAL

SEPTEMBER 2006

VANESSA GRAHAM MURDOCK

Bachelor of Science, COLORADO STATE UNIVERSITY

Master of Science, UNIVERSITY OF MASSACHUSETTS

Ph.D., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor W. Bruce Croft

Sentence Retrieval is the task of retrieving a relevant sentence in response to a query, a question, or a reference sentence. Tasks such as question answering, summarization, novelty detection, and information provenance make use of a sentence-retrieval module as a preprocessing step. The performance of these systems is dependent on the quality of the sentence-retrieval module. Other tasks such as information extraction and machine translation operate on sentences, either using them as training data, or as the unit of input or output (or both), and may benefit from sentence retrieval to build a training corpus, or as a post-processing step.

In this thesis we begin by demonstrating that because sentences are much smaller than documents, the performance of typical document retrieval systems on the retrieval of sentences is significantly worse. We propose several solutions to the problem of sentence retrieval, and investigate these solutions the application areas of sentence retrieval for question answering, novelty detection, and information provenance.

The context of a sentence affects its meaning, and we demonstrate that smoothing from the local context of the sentence improves retrieval when the collection to be retrieved from contains many documents of unknown relevance.

We show that statistical translation models are appropriate for tasks where the sentence to be retrieved has many terms in common with the query, but still benefits from the addition of related terms and synonyms. We show that queries of very few terms benefit from the translation approach, which incorporates related terms into the query. We show that the family of language modeling approaches, which includes statistical translation models, is not effective for discriminating between sentences that uses the same vocabulary to express the same information, and sentences that use the same vocabulary to express new information. Finally, we demonstrate a conditional model for sentence retrieval for question answering, and show that it outperforms both the translation approaches and the baseline language-modeling approach.

TABLE OF CONTENTS

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

CHAPTER 1

INTRODUCTION

Sentence Retrieval is the task of retrieving a relevant sentence in response to a query, a question, or a reference sentence. Tasks such as question answering, summarization, novelty detection, and information provenance make use of a sentence-retrieval module as a preprocessing step. The performance of these systems is dependent on the quality of the sentence-retrieval module. Other tasks such as information extraction and machine translation operate on sentences, either using them as training data, or as the unit of input or output (or both), and may benefit from sentence retrieval to build a training corpus, or as a post-processing step. In this thesis we address the question of how to retrieve sentences, and demonstrate sentence retrieval in the context of question answering, novelty detection, and information provenance.

Sentence retrieval is usually treated as a type of document retrieval. Techniques that were designed to find relevant documents in response to a user's query are employed in the service of finding relevant sentences. However, document retrieval differs from sentence retrieval in key ways that affect performance, and many of the assumptions made about document retrieval do not hold for sentence retrieval.

The most fundamental assumption about document retrieval is the assumption that a relevant document is on the general topic of the thing we are looking for. While this notion of "aboutness" is intuitive for documents, it is not sufficient for sentence retrieval. For example, if the task is question answering, a relevant sentence contains the answer to the question. It is not sufficient for the sentence to be "about"

the general topic of the question. So the first step in solving sentence retrieval is to define a notion of relevance appropriate to sentences.

The notion of relevance is captured by the way the similarity between the query and the document is modeled. Most modern information retrieval systems assume that documents that are on the same general topic as the query are relevant, and therefore look for documents that mention terms from the query. It follows that documents containing more query terms would be more relevant than documents containing fewer query terms. While a document may contain multiple mentions of a query term, a sentence is unlikely to contain more than one, especially if the term is content-bearing. Sentence-level similarity must be defined such that it captures the notion of relevance for sentences.

The most popular family of document retrieval models is the vector space model, which typically uses the "tf.idf" family of term weighting schemes. Although the various "tf.idf" approaches differ in how the term frequency (the "tf" part) and the inverse document frequency (the "idf" part) are combined, a common theme among the models in this family is that documents with a higher term frequency have a higher score. A second family of retrieval models is the language model approach, which also assigns a higher score to documents that have a higher term frequency. In both cases, documents containing multiples of the same query term will be scored higher than documents containing singleton query terms. Sentences, by contrast, usually consist of singleton query terms, and so ranking sentences based on term frequency is ineffective because the scores of the sentences have low variance. Thus in order to answer the question of how to retrieve sentences, we must answer the question of how to estimate term distributions in sentences, whose distributions are sparsely represented.

The assumption that documents containing more query terms are more relevant (or more similar to the query) extends to efforts to capture multiple ways of expressing

the same concepts. Techniques for query expansion and pseudo-relevance feedback assume that by expanding the query (perhaps with synonyms or other related terms) we can capture documents that are relevant but use a slightly different vocabulary than the original query.

In a typical document-retrieval scenario, the query contains an average of two or three terms, and the document may contain 500 terms. Expanding the query with 50 to 500 related terms makes the distribution of terms in the query and the distribution of terms in the document more comparable. Terms in the document that are not in the expanded query contribute nothing to the query term frequency counts. Terms in the expanded query that are not in the document contribute the term frequency of the background or smoothing distribution. In a relevant document, many of the terms in the document and the expanded query will contribute. In a relevant sentence, it is quite possible that only two or three terms in the expanded query will match a term in the sentence, because the sentence has so few terms to begin with. The outcome is for all sentences to have similar scores, regardless of their relevance.

If we were to expand the query with a small number of terms, we cannot be sure that we have chosen expansion terms that capture the original intent of the query. If we have chosen expansion terms on a slightly different topic we cause sentences not relevant to the original query to be retrieved by the expanded query. Thus query expansion and pseudo-relevance feedback techniques developed for document retrieval are likely not as effective for sentence retrieval. Since a sentence may contain a query term or its synonym, but is unlikely to contain both, in order to effectively retrieve sentences, we must accommodate different ways to express the same concepts, and compensate in some way for a vocabulary mismatch between the query and the sentence.

1.1 Contributions

Sentence retrieval is a new application area in the sense that it is rarely considered independent of a specific application. Rather it is considered part of an information retrieval module, and is treated as document retrieval, with trivial (if any) adjustments specific to sentences. The main contribution of this thesis is a study of sentence retrieval independent of any task. This primary contribution includes a mechanism for retrieving sentences which takes into account the notion of relevance specific to the application. The heart of this contribution is the model of similarity and the method of smoothing term statistics, which is flexible enough to capture varying notions of relevance. Secondary contributions of this work are the application of a conditional model to sentence retrieval, the representation of relevance as a spectrum, and a study of how to adjust the notion of similarity to capture varying notions of relevance.

Information provenance, which seeks the origins of information in a corpus through re-statements of the facts contained in a sentence, represents a novel application area, and as such is an application-oriented contribution of this thesis. It differs from plagiarism detection because we do not seek exact re-statements of fact or re-statements that have been edited slightly. It differs from novelty detection because we seek the same information rather than novel information on the same topic. It differs from *ad hoc* retrieval because *ad hoc* retrieval looks for documents on the same general topic, and does not characterize the similarity between the document and the query. Information provenance looks for sentences that contain the same information as a reference sentence, and seeks to characterize the originality of the sentence. A final contribution of this thesis is the creation of a corpus for further study of information provenance.

To summarize the contributions:

- A task-independent study of sentence retrieval
  - A representation of the relevance of a sentence as a spectrum

  - A model of similarity (Model-S) appropriate to a wide range of sentence retrival tasks
  - A novel application of conditional models to the task of sentence retrieval
  - A method of smoothing that significantly improves sentence retrieval
- A study of information provenance
  - An explanation of why the language model family of retrieval methods are not appropriate for information provenance.
  - A corpus for the further study of information provenance

1.2 Information Retrieval Background

Although we discuss information retrieval models in chapters 2 and 4, it is helpful to understand the way the models are evaluated, and the corpora used for the evaluation of information retrieval systems. Information retrieval systems typically consist of an index of documents, and a query which is usually provided by the user. The task of the system is to retrieve documents that are relevant to the user's query. Most often, documents are presented in a ranked list, where the documents most likely to be relevant are presented at the top of the list, and documents less likely to be relevant are presented at the bottom of the list.

Documents and queries can be any unit of information, but this thesis concerns sentence retrieval, which means that sentences are treated as documents. To avoid confusion with other types of documents, sentences are referred to as *sentences* in this thesis, and the term *document* refers to newswire articles. In general a query is any information the system uses to rank documents, and may include terms the user has provided in addition to other information about the user, such as documents the user has previously viewed, or expansion terms the system has provided. In this work

a query is either a question, a set of terms, or a sentence. The data used in this thesis was provided by NIST[1] as part of the Text REtrieval Conference (TREC)[2].

Various metrics are used to evaluate retrieval systems. Most of the work in this thesis is evaluated using *precision at N*. Precision at N is the number of relevant documents in the top N documents, averaged over all of the queries. A related metric is *mean average precision* which is the mean of the precision at the rank of each relevant document for a given query, averaged over all queries. In this thesis, the results reported for *average precision* refer to *mean average precision*. Recall is the number of relevant documents that have been retrieved, divided by the total number of relevant documents. A t-test is an appropriate significance test for precision at N and mean average precision because these metrics simply count the number of relevant sentences in a set of retrieved sentence, ignoring the rank of the sentences.

A final metric referred to in this thesis is interpolated recall-precision, which measures the precision after a certain percentage of relevant documents have been retrieved. Precision is typically calculated after 10 percent, 20 percent, etc., up to 100 percent of the relevant documents have been retrieved. *Interpolated* means that, for example, precision at recall level 0.10 is taken to be the maximum of precision at all recall points greater than or equal to 0.10. Although the t-test is commonly used in information retrieval as a test of the significance of an interpolated recall-precision result, a t-test is not appropriate because it ignores the rank of the relevant sentences. Instead the sign test is the appropriate significance test for results from metrics that consider the rank of the relevant sentences. Further information about information retrieval evaluation metrics is given in Baeza-Yates and Ribeiro-Neto [6].

[1] http://www.nist.gov

[2] http://trec.nist.gov

1.3 Thesis Overview

We begin the thesis with evidence that document retrieval techniques are not effective for sentence retrieval in Chapter 2. We follow with a discussion of relevance and similarity for sentences (Chapter 3), and a generative model for sentence-level similarity in Chapter 4. We demonstrate the model, using three application areas as examples. The task of question answering (Chapter 5) has a well-studied data set with sentence-level relevance judgments. The notion of relevance is specific: a relevant sentence contains the answer to the question, in the proper context. The notion of similarity is not well-defined, as the answer sentence must be on the same topic as the question, but there are no specific similarity requirements. The second application area, retrieval of topically relevant sentences, as represented by the relevance task of the TREC Novelty Track (Chapter 6), embodies many of the issues that arise when a sentence is treated as a document. In this chapter we explore conditions under which document retrieval techniques are successful for sentence retrieval, and conditions under which document retrieval techniques can be improved upon. Information provenance (Chapter 7) is an emerging field, and we continue preliminary work in this area by showing that a language modeling approach is not the best approach for this task.

CHAPTER 2

THE CASE AGAINST DOCUMENT RETRIEVAL FOR SENTENCES

Of the research in question answering, novelty detection, extractive summarization, and other tasks requiring a sentence or passage retrieval module, the vast majority focuses on the final step in processing. Question answering research is primarily concerned with extracting an answer token. Extractive summarization is primarily concerned with choosing sentences from a set of "relevant" sentences to represent a summary. Implicit in the design of such systems, and in the body of research devoted to them, is the assumption that sentence retrieval is document retrieval. However, many papers (for example [28, 61, 109, 22]) note that the performance of their system is affected by the quality of the retrieval module. Allan et al. [5] demonstrated that the performance of their novelty detection system was dependent on the quality of the sentence retrieval module. Harman noted in the overview of the TREC Novelty Track in 2002 [54] that the number of novel sentences was inversely proportional to the number of relevant sentences. They found that if there were many relevant sentences, fewer of them were novel, whereas if there were few relevant sentences, a greater proportion of the relevant sentences was novel.

Research in document retrieval has implicitly assumed documents are a certain typical length. Although a length assumption is never explicitly stated, all of the standard test collections, including the earlier Cranfield collection, consist of newswire articles. The newswire articles in the Aquaint collection are an average of 14 sentences in length. In TREC volumes 1 and 2, the average document is 22 sentences, in TREC volume 3, the average document length is 23 sentences, and in TREC volumes 4 and

5, the average document length is 25 sentences. Although the models also do not explicitly state a length assumption on the documents, they are designed and tested and improved with newswire documents. We investigated whether performance of a retrieval system degrades as the documents become shorter.

The definition of relevance was fixed by using question answering data. The query set was the TREC QA-track questions from the passages task from 2003, which consists of 413 questions, from the questions numbered 1894 - 2393. The top 1000 documents were retrieved from the Aquaint corpus, which consists of newswire documents from the Associated Press, Xinhua News Agency and the New York Times News Service, from the years 1998 - 2000. Of the 413 questions in the passages task, 375 had answers in the top 1000 documents. Those are the questions for which these results were computed. The sentences were judged relevant if they contained the answer token, regardless of the context. The answer tokens were detected automatically using regular expressions. The questions were stemmed using the Krovetz stemmer [66], and single characters were removed. The retrieval model was the Query Likelihood model (details of the model follow), with Jelinek-Mercer smoothing. The smoothing parameters were optimized using the relevance judgments.

The first column in Tables 2.1 and 2.2 shows the evaluation of the retrieval system when the unit of information retrieved is a complete document. For columns two through four, the documents were split into k-byte windows, which overlapped by half. That is, for two consecutive windows A and B, the second half of A was identical to the first half of B. Thus the 750-byte passages overlapped by 375 bytes, the 500-byte passages overlapped by 250 bytes, and the 250-byte passages overlapped by 125 bytes. The final column shows sentences, which were obtained by splitting the documents into sentence segments, using MXTerminator [111], a maximum-entropy based sentence segmenter freely available on the web. The results were tested for statistical significance using a two-tailed t-test, and without exception, were found

	Docs	750 bytes	500 bytes	250 bytes	Sents
Relevant	31672	99703	112544	130139	73623
Relevant and Retrieved	31672	19278	16876	13026	8639
Interpolated Recall-Precision					
at 0.00	0.497	0.421	0.379	0.327	0.268
at 0.10	0.329	0.172	0.153	0.103	0.085
at 0.20	0.270	0.119	0.092	0.061	0.052
at 0.30	0.234	0.073	0.059	0.035	0.036
at 0.40	0.201	0.048	0.042	0.028	0.028
at 0.50	0.186	0.039	0.034	0.023	0.022
at 0.60	0.162	0.023	0.021	0.012	0.007
at 0.70	0.140	0.018	0.016	0.009	0.006
at 0.80	0.126	0.015	0.013	0.007	0.004
at 0.90	0.116	0.013	0.012	0.006	0.003
at 1.00	0.109	0.012	0.010	0.005	0.003

Table 2.1. Comparison of the effects of document length on retrieval, evaluated with interpolated precision at various recall levels. Query-likelihood retrieval with Jelinek-Mercer smoothing was used on documents, overlapping passages of 750 to 250 bytes, and sentences. Results are statistically significant using a two-tailed t-test at the $p < .05$ level, compared to the document retrieval baseline.

	Docs	750 bytes	500 bytes	250 bytes	Sents
Average Precision	0.191	0.064	0.055	0.036	0.030
Precision					
at 5 docs	0.274	0.209	0.190	0.155	0.107
at 10 docs	0.232	0.186	0.166	0.136	0.098
at 15 docs	0.211	0.161	0.151	0.128	0.088
at 20 docs	0.200	0.149	0.142	0.117	0.081
at 30 docs	0.185	0.132	0.128	0.105	0.071
at 100 docs	0.140	0.096	0.093	0.074	0.052
at 200 docs	0.118	0.080	0.075	0.061	0.041
at 500 docs	0.097	0.063	0.057	0.044	0.031
at 1000 docs	0.085	0.051	0.045	0.035	0.023

Table 2.2. Comparison of the effects of document length on retrieval, evaluated with precision at n documents retrieved. Query-likelihood retrieval with Jelinek-Mercer smoothing was used on documents, overlapping passages of 750 to 250 bytes, and sentences. Results are statistically significant using a two-tailed t-test at the $p < .05$ level, compared to the document retrieval baseline.

to be significantly worse than the baseline document retrieval system at the $p < .05$ level.

2.1 Term frequency

One of the first, and arguably most popular, modern retrieval models was the vector-space model which treats the document and the query as vectors of term weights. The documents are ranked by the similarity of their vector representation to the vector representation of the query. The most common similarity metric is cosine similarity which ranks a document by the cosine of the angle between the document vector and the query vector. The term weights are most often given by the tf.idf family of term weighting schemes. There are many variations [118], but a common one is:

$$
\begin{aligned}
tf_{q_i,D} &= \frac{c(q_i; D)}{\max_l\{c(q_l; D)\}} \\
idf_{q_i} &= \log \frac{N}{n_{q_i}} \\
w_{q_i,D} &= tf_{q_i,D} \cdot idf_{q_i}
\end{aligned}
\tag{2.1}
$$

where $c(q_i; D)$ is the count of term q_i in document D, $\max_l\{c(q_l; D)\}$ is the count of the most frequent term l in document D, N is the number of documents in the collection, and n_{q_i} is the number of documents containing term q_i.

Documents that have a higher term frequency will be ranked higher than documents that have a lower term frequency. The *idf* score seeks to prevent terms that are very frequent and appear in many documents (like "the" or "and") from dominating the document score. Intuitively, if a term appears in many documents it is not as informative with regard to relevance.

One limitation of the vector-space model is that the term weights are determined heuristically. As a remedy to this various probabilistic models were proposed, including the Binary Independence Model [115] and Inference Networks [132, 112]. The query-likelihood model was first proposed for document retrieval by Ponte and Croft [103]. This model differs from earlier probabilistic models in that it does not model relevance explicitly. Query likelihood for document retrieval ranks documents by the probability that the query was generated by the same distribution of terms the document is from:

$$
\begin{aligned}
P(Q, D) &= \frac{P(Q|D)P(D)}{P(Q)} \\
P(Q|D) &\propto \prod_{i=1}^{|Q|} P(q_i|D) \\
P(q_i|D) &= \frac{c(q_i; D)}{|D|}
\end{aligned}
\tag{2.2}
$$

where Q is the query, $|Q|$ is the number of terms in the query, q_i is the i^{th} term in the query, and D is a document. In this model, we assume query terms are independent of each other given the document. Since the prior probability of a query is the same for all documents in the collection, $P(Q)$ does not affect the ranking. The prior probability of a document, $P(D)$, is typically assumed to be uniform, and in this case does not affect the ranking. The probability $P(q_i|D)$ is usually estimated as the count of terms q_i divided by the number of terms in the document, D. Clearly, the more mentions of a given term q_i in a document, the higher the document's score.

Documents in the TREC collection, volumes one through five, have an average of 458 words,[1] excluding the four subcollections with very large documents (Federal Register (1988 and 1989), Congressional Record (1993) and U.S. Patents (1993)). Documents from the Aquaint Collection, which includes newswire documents from the Associated Press, the Xinhua News Agency, and the New York Times (1998-2000), have an average of 250 words. By contrast, sentences from the Aquaint collection have an average of 18 words. Only a small number of query terms will be present in a relevant sentence, and because sentences contain so few terms to begin with, there is not much variation in scores for relevant and nonrelevant sentences. Clearly, counting the occurrence of query terms is less effective for sentences than for documents.

2.2 Pseudo-Relevance Feedback and Query Expansion

In document retrieval, query expansion and relevance feedback address the issue of vocabulary mismatch by adding related terms to the query. Query expansion creates a new query that consists of the original query terms plus new related terms. It has been studied by many researchers, starting with Maron and Kuhns in 1960 [83]. Although query expansion has been shown to benefit some queries when the expan-

[1]Numbers were obtained from [6], p. 87.

sion terms are chosen by hand, automatic query expansion has had mixed results for document retrieval [23, 91, 133, 141]. Query expansion is most successful on poorly specified queries. Expanding the query with poorly chosen terms degrades performance regardless of the query.

To determine whether query expansion was beneficial to sentence retrieval, we used the 150 topic descriptions and relevance judgments from the TREC Novelty Track in 2002, 2003, and 2004. The top 1000 documents were retrieved from the TREC Collection, volumes 4 and 5, and the Aquaint Collection. The documents were sentence segmented using MXTerminator [111]. For the baseline retrieval, the top 1000 sentences were retrieved using query likelihood, and their relevance assessed using the relevance assessments provided by NIST for the Novelty Task. The notion of relevance for the Novelty Track is similar to the notion of relevance for *ad hoc* retrieval. Description queries were used because the majority of sentence retrieval tasks take sentence-length queries.

Queries were expanded from a probabilistic dictionary of related terms learned from TREC topic titles, descriptions and narratives (excluding those TREC topics that were present in the Novelty data). This dictionary had been shown to improve retrieval performance when used in a translation model framework (presented in Chapters 4 and 6). Table 2.3 shows the results of query expansion. The metric MRR is defined as $\frac{1}{R}$, where R is the rank of the first relevant sentence, averaged over all queries. The metric *MRR at N* only considers the top N sentences, and if the first relevant sentence has rank lower than N it receives a score of zero. Query expansion degrades sentence retrieval performance significantly.

Relevance feedback collects terms from known relevant documents or clusters of related terms, and uses these terms in place of the original query. Pseudo-relevance feedback also replaces the query with terms from documents, but whereas in relevance feedback the documents or term clusters have been judged relevant by a person (for

	Query Likelihood	Query Expansion
Prec @ 1	.168	.074
Prec @ 5	.117	.043
Prec @ 10	.111	.044
Prec @ 15	.101	.044
Prec @ 20	.096	.040
Prec @ 1000	.037	.023
Recall	.506	.416

Table 2.3. The results of query expansion using topic descriptions and relevance judgments from the TREC Novelty track. Results are significant using a t-test at the $p < .05$ level.

instance, the user), in pseudo-relevance feedback the documents are automatically retrieved and assumed - but not guaranteed - to be relevant. Relevance Models were first proposed by Lavrenko and Croft [76] and have shown to benefit retrieval in a wide variety of tasks [77, 62, 75, 74]. Relevance models do an initial (query-likelihood) retrieval, using the original query. A model is constructed from the top N retrieved documents, and m content terms are sampled from the distribution of terms in the model. This set of sampled terms serves as a distribution of query terms, and the documents are re-ranked according to the likelihood they generated the new distribution of query terms. (Alternately, the m terms can be interpolated with the query terms.) N and m are parameters to be tuned.

To investigate the effects of pseudo-relevance feedback on sentence retrieval, we used the Novelty Data, as in the query expansion experiments. The top m feedback terms were chosen from the top N sentences. We investigated values of m from five to 1000, and values of N from five to 500. The best result expanded the query with the top 75 terms from the top 50 - 75 sentences. Table 2.4 shows the results. Relevance models degrade retrieval performance, though the difference is not statistically significant.

Relevance models create a topic model of the query from the top N documents. The top N documents were retrieved in the first place because they had terms in

	Query Likelihood	Relevance Models
MRR at 5	.241	.217
MRR at 20	.269	.239
MRR	.274	.245
Prec @ 1	.168	.161
Prec @ 5	.117	.123
Prec @ 10	.111	.113
Prec @ 15	.101	.100
Prec @ 20	.096	.094
Prec @ 1000	.037	.037
Recall	.506	.506

Table 2.4. The results of relevances models using topic descriptions and relevance judgments from the TREC Novelty track. Results are not statistically significant.

common with the query, and since documents that are on-topic are assumed to share a common vocabulary, using the most frequent content terms from the documents provides a more enriched model of terms in the topic of the query than the original query. A relevant document will contain many of the terms in the expanded query, because the expansion terms co-occurred frequently in the top N documents (by construction). Terms in the expanded query that are not in the document will get the background probability, but for a relevant document the scores for the matching terms will dominate the scores from the non-matching terms. If the query is expanded with a few spurious terms from a different topic, there may be documents that are retrieved because of these terms, but their scores will be lower than the scores of the relevant documents because there will be fewer spurious matching terms.

A relevant sentence, by contrast, will match a few terms from the expanded query, and is unlikely to contain multiples of the same term. If the query is expanded with a few spurious terms unrelated to the topic of the query, a non-relevant sentence that matches the spurious terms will match as many terms in the query as a relevant sentence. Furthermore, since the sentence has so few terms to begin with, most of the

terms in the expanded query will receive the background score, causing the scores of relevant and nonrelevant sentences to be similar.

Finally, relevance models are designed to capture a model of the topic of the document. By their nature they capture the notion of relevance required by document retrieval: the notion of topicality. For many sentence retrieval tasks, topicality is not sufficient for relevance. For this reason, in addition to the reasons outlined above, relevance models are not ideal for sentence retrieval.

2.3 Smoothing

In the query-likelihood model, words that appear in the query that do not appear in the sentence have a probability of zero, which results in a zero probability of the query having been generated by the sentence distribution. To resolve this problem, smoothing is introduced to give a non-zero probability to unseen words. There are many smoothing strategies, but we present four common smoothing techniques: Absolute Discounting, Jelinek-Mercer smoothing, Dirichlet smoothing, and LaPlace smoothing.

Absolute Discounting redistributes some of the probability mass from the seen words to the unseen words:

$$P(Q|S) = P(S) \prod_{i=1}^{|Q|} \frac{\max(c(q_i; S) - \delta, 0)}{|S|_{total}} + \frac{\delta |S|_{unique} P(q_i|C)}{|S|_{total}} \tag{2.3}$$

where $c(q_i; S)$ is the count of the query term q_i in the sentence S, $P(q_i|C)$ is the probability that the query term q_i was generated by some other distribution, in our case the collection C. $|S|_{unique}$ is the number of unique terms in the sentence, $|S|_{total}$ is the total number of terms in the sentence, and δ is a real-valued parameter, between 0 and 1. Sentences are composed of mostly unique terms, so in this case,

$$\frac{\delta|S|_{unique}}{|S|_{total}} \approx \delta \tag{2.4}$$

In Jelinek-Mercer smoothing the sentence distribution is linearly interpolated with another distribution, in our case the distribution of terms in the collection.

$$P(Q|S) = P(S)\prod_{i=1}^{|Q|}(1-\lambda)P(q_i|S) + \lambda P(q_i|C) \tag{2.5}$$

where Q is the query, S is the sentence, C is the collection, and λ is a real-valued parameter between 0 and 1.

Dirichlet smoothing estimates the term distribution as a function of the length of the document being smoothed.

$$P(Q|S) = P(S)\prod_{i=1}^{|Q|}\frac{c(q_i;S) + \mu P(q_i|C)}{|S| + \mu} \tag{2.6}$$

where $c(q_i;S)$ is the count of word q_i from the query in the sentence S, $|S|$ is the total length of the sentence, and μ is a parameter greater than zero. LaPlace smoothing is a special case of Dirichlet smoothing where $\mu = 1$.

If the document length is small in comparison to μ, more weight will be given to the collection probabilities than if the document length is large in comparison to μ. Since μ is fixed for all documents, Dirichlet smoothing penalizes long documents less than short documents, and herein lies its advantage. This effect is demonstrated by Smucker and Allan [125].

Zhai and Lafferty [147] found that smoothing has a greater effect on long queries than on short, they also concluded that Jelinek-Mercer smoothing was more effective for long queries, with more weight given to the document probabilities than the collection probabilities.

	Average Words	Shortest	Longest
Topic Titles	4.11 ± .894	2	7
Topic Descriptions	14.4 ± 7.01	5	63
Sentences	18.39 ± 2.74	10	27
Documents	714.81 ± 1031.26	168	10,770

Table 2.5. Comparing the lengths (in words) of queries, sentences and documents. Documents have much higher variance in length than sentences. Topic descriptions and sentences are about the same length.

If we let the Dirichlet parameter μ equal the length of the sentence then Dirichlet smoothing is exactly Jelinek-Mercer smoothing where the parameters give equal weight to the document and the collection.

$$\begin{aligned} P(Q|S) &= P(S) \prod_{q \in Q} \frac{c(q;S) + |S|P(q|C)}{|S| + |S|} \\ &= P(S) \prod_{q \in Q} \frac{1}{2} P(q|S) + \frac{1}{2} P(q|C) \end{aligned} \tag{2.7}$$

which is exactly Equation 2.5, where $\lambda = .5$, as is stated in Smucker and Allan [125]. Thus, Dirichlet smoothing is a version of Jelinek-Mercer smoothing where the smoothing parameter is a function of the document length.

We investigate smoothing in the context of the TREC Novelty task because the definition of relevance for the Novelty task most closely resembles the definition of relevance for *ad hoc* document retrieval. We use the 150 topic titles and descriptions and the documents provided from the TREC Novelty Track in 2002, 2003, and 2004. Table 2.5 shows that in 150 TREC-style topic descriptions and titles, and newswire documents from the TREC Novelty Track (which are a subset of the TREC volumes four and five, and the Aquaint corpus), there is much less variance in the length of a sentence than in the length of a document, thus we do not expect to see as great a benefit in performance from Dirichlet smoothing for sentence retrieval as has been reported for document retrieval [147].

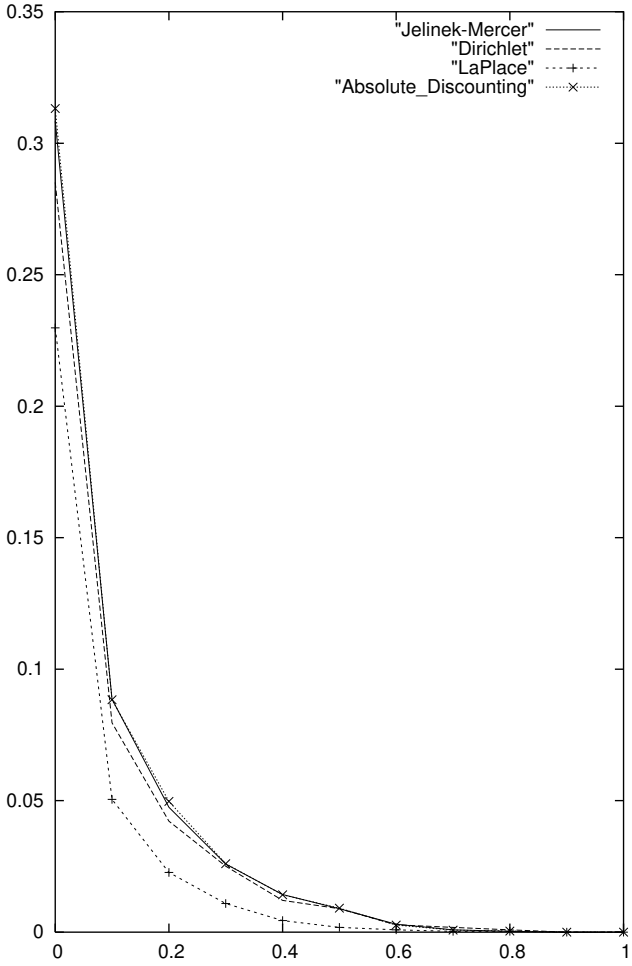

Figure 2.1. Comparing smoothing strategies for sentence retrieval on TREC-style topic descriptions. There was no significant difference between smoothing strategies, except that the result for LaPlace smoothing for topic descriptions was significantly worse using a Wilcoxon sign test at the $p < .05$ level.

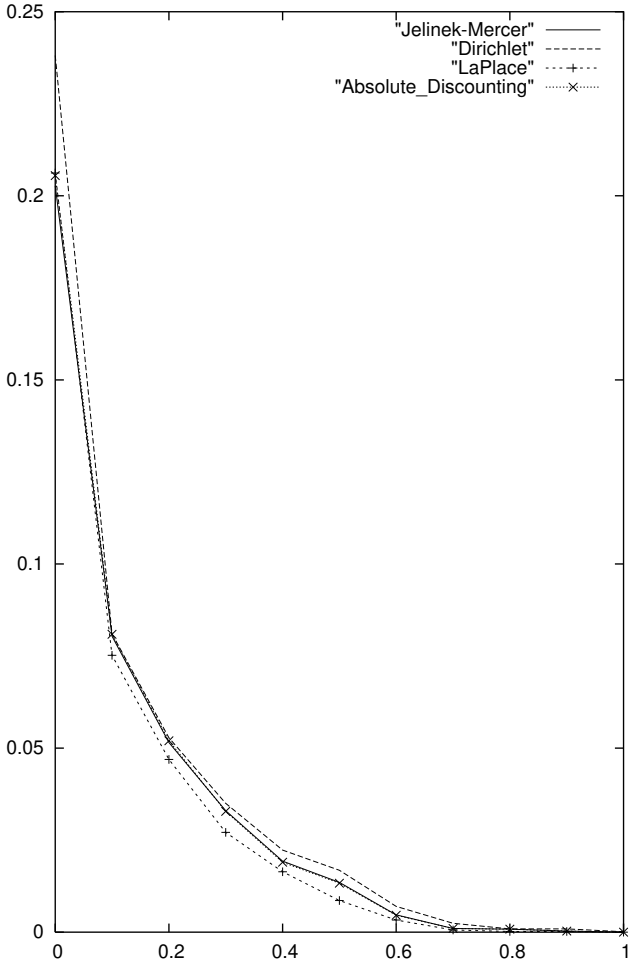

Figure 2.2. Comparing smoothing strategies for sentence retrieval on TREC-style topic titles. There was no significant difference between smoothing strategies, except that the result for LaPlace smoothing for topic descriptions was significantly worse using a Wilcoxon sign test at the $p < .05$ level.

Figure 2.2 shows a comparison of smoothing methods for TREC-style topic descriptions and titles from the TREC Novelty track. The results for LaPlace smoothing on topic descriptions are statistically significant using a Wilcoxon sign test at the $p < .05$ level. The results for topic titles showed even less difference between the different smoothing strategies. As expected, (with the exception of LaPlace smoothing, which is worse) there is no significant difference between smoothing methods for sentence retrieval.

2.4 Conclusion

As document length is reduced, retrieval performance diminishes. There are multiple factors that contribute to the lower performance on sentences. Retrieval models score documents higher that have higher term counts, without differentiating between documents that contain multiples of the same term, and documents that contain multiple unique terms. Efforts to compensate for vocabulary mismatch between the query and the document depend on documents having many terms, and are based on the assumption that a relevant document will have many terms in common with the expanded query. A relevant sentence, however, can only have a few terms in common with the expanded query, because sentences contain only a few content terms to begin with. Smoothing methods that have been shown effective for document retrieval are not as effective for sentences because sentences are much more sensitive to smoothing, and quickly become so smooth as to be indistinguishable from other less relevant sentences.

At the heart of the discrepancy between document retrieval and sentence retrieval is the nature of relevance. Document retrieval systems are designed to capture topicality, because a document that is on the topic of the query is considered relevant. Topicality is sufficient for few, if any, sentence retrieval tasks. The first step in answering the question of how to retrieve sentences, is constructing a model that is

flexible enough to capture differing flavors of relevance embodied in different sentence retrieval tasks.

CHAPTER 3

RELEVANCE AND SIMILARITY OF SENTENCES

The information retrieval community in computer science has been engaged in a fifty-year conversation about relevance. As a measurable quantity it is difficult to pin down. If we are talking about the relevance of a document, we have at least an intuitive sense of what that means, even if it is difficult to quantify. Discussions of the relevance of a document typically assume the task at hand is document retrieval. Sentence retrieval is a supporting technology for a variety of tasks, and each task has a separate notion of relevance. Any meaningful discussion of sentence-based retrieval systems must begin with a discussion of relevance.

Saracevic [119] proposed that "relevance" is a relation between an information request and a document (where a document can be loosely thought of as any piece of information). In our case a "document" is a sentence, and the relation that defines relevance is a measure of the similarity between the sentence and the information request, which may be another sentence, or a question, or a set of keywords.

To clarify our discussion of relevance and similarity in the context of sentences we propose a spectrum of relevance, and a spectrum of similarity. Tasks that utilize sentences lie in different areas of the spectrum. It is necessary to de-construct relevance and similarity in this way because the appropriate retrieval model for a given sentence-retrieval task captures the aspects of similarity embodied by a specific area of the spectrum. In the following sections we decouple relevance and similarity, and further de-construct similarity into measurable quantities.

3.1 The Relevance of a Sentence

When information retrieval as a branch of computer science was young, most discussions of relevance assumed the task at hand was document retrieval. The nature of document retrieval at that time is illustrated in Maron and Kuhns [83]. Each document in a library was read by a librarian and assigned a set of index terms from a finite set of terms that described the content of the document. The focus of information retrieval was to find the documents whose index terms most closely resembled information requested by a person, usually in a library. As large-scale indexing and search became more practical, people looked for a way to generate descriptor terms, and retrieve documents automatically. The concept of relevance was almost completely defined in the context of document retrieval.

Robertson [113], proposed that there exists an underlying continuum of relevance, and that ranking documents according to their probability of being relevant [114], rather than defining discrete categories such as "relevant" and "non-relevant," more accurately captures the continuous nature of relevance. In order to evaluate systems, an assessor places a document in a category, even though his understanding of the relevance of the document is more continuous than discrete in nature. Most relevance assessments are either binary decisions ("relevant" versus "non-relevant") or slightly more flexible categorical decisions ("highly relevant", "partially relevant", or "non-relevant") as proposed in Gebhardt [48] in 1975.

Sentence retrieval is a supporting technology for many different applications. The relevance of a sentence is dependent on the application the sentence is being used in. For example, if the task is question answering, a relevant sentence contains the answer to the question. If the task is extractive summarization, the relevant sentence provides information topically related to the query but not redundant in relation to other sentences retrieved. For each task, there may be an underlying continuum of

relevance as proposed by Robertson, but on a higher level, we propose there is a spectrum of relevance.

3.2 The Spectrum of Relevance

Useful Content

Every utterance in language has meaning, and therefore content, but not all content is useful for a given task. An example of a sentence that might have useful content only in the context of a document is "'The expectations are very low,' an official close to the negotiations said Thursday."[1] A sentence that is segmented incorrectly may lose content, making it less useful. An example of segmentation affecting the content of a sentence is "Speaking on the eve of the semiannual EC-U.S.".[2] And some sentences never had useful content, regardless of the document they appear in, as in "If I didn't tell Woodrow Wilson, why would I tell you?"[3].

If sentence retrieval is being used in a document retrieval application, this is of little consequence, because the sentence is never taken in isolation. Although a sentence that has no useful content may not hurt the performance of a document retrieval system, it certainly cannot help.

For other applications, such as question answering, it is crucial that the sentence contain useful content. To circumvent problems arising from this, the standard for research in question answering has defined a relevant sentence as one that contains

[1]From a 1988 AP newswire article (TREC volumes 1 and 2, AP880318-0287)

[2]From a 1987 Wall Street Journal newswire article(TREC volumes 1 and 2, WSJ871210-0139) sentence-segmented by MXTerminator [111].

[3]http://www.myfavoritegames.com/dragonball-z/Info/ChatLogs/Dragonball-Z-GT-ChatLog-BriceArmstrong.htm August 2006

the right answer, and appears in the proper context [135], and actual QA systems, such as Brain Boost[4], and START[5], provide a link to a source document.

We propose that "useful content" is the most general form of relevance, because a sentence that has does not contain useful information cannot be plagiarised, or answer a question, or be considered relevant to a request, even if it is similar.[6]

Tangentially Related

An example of tangentially related information would be documents about organic farming, and documents about the restaurant *Chez Panisse* in Berkeley, which uses organic produce. In the context of *ad hoc* document retrieval, tangentially related documents are less likely to be relevant. A person querying a web search engine about the menu at *Chez Panisse* is less likely to be interested in documents about organic methods of pest control. However, if the task is recommender systems, such as those that recommend book titles at the point of sale in online bookstores, tangentially related information is exactly the type of information to be retrieved by the system.

On the General Topic

The vast majority of research in document retrieval has focused on finding documents on the same general topic as a query. Setting the dividing line between documents on the general topic and documents on a tangential topic is the task of distinguishing a relevant document from a non-relevant document. An example of two sentences on the same general topic (National Parks) would be:

> 1. *National Parks tell a story that must be instilled in each generation, beginning by engaging the youngest in our society in the majesty, impor-*

[4]http://www.brainboost.com June 2006

[5]http://start.csail.mit.edu June 2006

[6]In the case of plagiarism detection, "useful content" might be defined to exclude sentences that are so common in the language that they cannot be said to be copied.

tance and splendor of America's National Parks.[7]
2. *Our national parks are home to more than 60 percent of the nation's endangered species, and represent some of the best remaining habitat for our wildlife heritage.*[8]

On the Sub-Topic

The term "topic" can be interpreted in many ways, so a "subtopic" is necessarily vague. Nevertheless, if we define a general topic, such as "National Parks", then we can define a subtopic of this general topic, such as "National Parks funding." An example of two sentences on the same subtopic would be:

1. *For decades, the National Park Service (NPS) has lacked the funds needed to manage adequately the wildlife, landscapes and historic and cultural artifacts protected within the National Park System.*[9]
2. *While congress in recent sessions has regularly increased funding for national parks, the National Park Service's budget has failed to keep pace with increasing pressures from visitation, over-development, motorized vehicle use, and air and water pollution.*[10]

Sentences that are on the same subtopic may contain redundant information. The general idea of both sentences is the same, although they might be characterized as parallel ways to express the same idea. An example of a task using information on the same subtopic is aspect retrieval, or cluster-based retrieval.

Provides Supporting Information

If we were constructing a summary of a document (or more than one document), the

[7]http://www.nationalparks.org/KidsPrograms/JuniorRangerPrograms.shtml June 2006

[8]http://www.npca.org June 2006

[9]http://www.npca.org/media_center/reports/analysis.asp June 2006

[10]http://www.forestsforever.org/nationalparksalert.html June 2006

two sentences in the above example would not both be useful to the summary, because the information content is for the most part redundant. To construct a summary from existing sentences (the task of extractive summarization), we require sentences that are on the same subtopic, but that provide new information, relative to other sentences in the summary. For example[11]:

> 1. *The National Park Service created the Junior Ranger program in the early 1960s to connect children and families with the natural and cultural history found in National Parks.*
> 2. *The program has grown and now serves nearly 330,000 children annually in 286 Parks.*

Satisfies the Request Directly

An information request might be satisfied directly with a document, or a sentence or a text token. The task of question answering requires this type of relevance. An example might be a question about how to apply for a passport, and a document detailing how to apply for a passport. In the context of factoid question answering, an example would be a question such as *How many children participate in the Junior Ranger program in the National Parks Service?* and the answer, *The (Junior Ranger) program has grown and now serves nearly 330,000 children annually in 286 Parks.*

When retrieving sentences for a particular task, the definition of relevance is dictated by the task. The notion of relevance for a particular task is captured in the type of similarity that is modeled. In document retrieval, where relevance is usually defined as "on the general topic" or "on the subtopic," models are designed to score documents highly that have many words in common with the query. Techniques have been developed to increase the degree of overlap between the query and the docu-

[11]http://www.nationalparks.org/KidsPrograms/JuniorRangerPrograms.shtml June 2006

ment. As the notion of relevance differs for each task, we must model varying types of similarity.

3.3 The Spectrum of Similarity

Similarity may be assessed by simple overlap between the request vocabulary and the document vocabulary. It may be more complex, for example incorporating information about the person posing the request, or structural information about the document, or expanded representations of the request. We examine two facets of similarity: lexical choice and structure. Although topical similarity is typically evaluated by lexical similarity, it should not be assessed solely by vocabulary overlap, because synonym terms or related words are also indicators of lexical similarity. Structure information has been effectively used in applications such as question-answering, which looks for structural patterns indicative of answer tokens, and machine translation where sentences in two languages are compared for structural elements that will indicate a proper alignment.

3.3.1 Lexical Similarity

Lexical similarity is an indicator of the degree to which two pieces of information discuss the same topic. Most language technologies have addressed lexical similarity, or the relatedness of two words, in one way or another. A document such as a news article may have multiple instances of the same word, or a word and its morphological variants, or a term and its synonym terms appearing in the same document. A sentence is unlikely to have multiple instances of any content term, or a content term and its synonyms, or a content term and its variants, making sentence retrieval much more sensitive to issues in lexical similarity.

Exact Match

If two words are exactly the same, they are clearly an exact match. Models such as the query-likelihood model, and the vector-space model with tf.idf term weighting look for exact matches between terms in a query and terms in a document. Obviously, if the terms are *exactly* the same term, the meaning of "exact match" is clear. Morphological variants (such as "runs" and "running") are also considered "exact matches" as are spelling variants (such as "email" and "e-mail"). A sentence is unlikely to contain both a term and its morphological variant, or a term and an alternate spelling of the same term.

Matching at the Synonym Level

Matching at the synonym level includes terms such as *height* and *altitude*. Documents frequently contain both a term and its synonym. A sentence is unlikely to contain both a term and its synonym.

Matching at the Related Term Level

An example of two sentences matching at the related term level are[12]:

> *How tall is Mt. Everest?*
> *The official altitude of the world's highest peak is 29,029 feet (8,848m).*

One would expect documents about Mt. Everest to contain terms such "altitude", "highest", "peak", and "feet". This example shows two sentences that have no exact matching, no synonym matching, and yet the second sentence satisfies the first directly, because terms in the second sentence are related to terms in the first.

[12]http://www.teameverest03.org/everest_info/ June 2006

Matching at the Co-occurrence Level

If two terms frequently appear in the same document, sentence, or window, they are said to *co-occur*. Sentences that contain terms that frequently co-occur in other sentences or documents, match at the co-occurrence level.

Unrelated Terms

For the purpose of the current research, terms not falling in the other categories are considered unrelated.

Stemming [104, 66] increases the level of exact matching between words and their morphological variants by conflating words to their common forms (such as reducing plurals to singular, possessive to non-possessive, past tense to infinitive). In document retrieval, stemming has been shown to improve recall [66], but can hurt precision because words with distinct senses may be conflated to the same form (such as "army" and "arm"). These mistakes are costly in sentence retrieval.

Query expansion, pseudo-feedback and translation models all find lexical similarity on the synonym, related term, and co-occurrence levels. Voorhees [133] showed that expanding the query terms with synonyms from WordNet [89], does not improve retrieval, even when the synonyms are chosen by hand, possibly because expansion terms were equally weighted. Pseudo-feedback expands the query with the top N terms from the top ranked documents for a given query. Typically terms are weighted individually by a heuristic formula (as in Rocchio feedback [116]) or by their probabilities in the document (as in Relevance Models [76]).

Depending on the formulation, sometimes the original query terms are omitted, or the number of expansion terms far exceed the original length of the query. This can lead to a problem that the terms that are indicative of the topic of the expanded query may not be indicative of the topic of the original query. Massive query expansion with

appropriately weighted expansion terms have been shown to be effective for document retrieval [23], but sentences have so few words to begin with that the effect is to add noise to the sentence retrieval process. Sentences containing expansion terms may not be as relevant to the original query as sentences containing the original query terms.

Translation models incorporate synonym term statistics into the model without explicitly expanding the query. An advantage of translation models is that query terms are scored as a combination of their synonym term scores, weighted by the probability that the query term is a synonym of the synonym term. Translation models have been used effectively to find synonym terms using a probabilistic dictionary derived from randomly chosen documents [11], or from a parallel corpus of example sentence pairs [94, 10].

3.3.2 Structural Similarity

While the relationship between sentences that are lexically similar is clear, identifying sentences whose construction is related is useful for many tasks. We present aspects of structural similarity, followed by examples where this information has been useful.

Identical Construction

The underlying structure of the sentences is identical, regardless of the actual word tokens in the sentence. Examples of sentences that share identical construction[13]:

How do you select a telephone system?
How do I get a state id?
How do I register a business name?

[13]Questions from the query logs of GovBot, which was a search engine for government and military pages used by the public, as well as civil servants, in operation at the CIIR from 1996 - 2000.

When parsed by SIFT [90], all three sentences have exactly the same parse tree:

```
 (SBARQ
(WHADVP
  (WRB how))
(SQ
  (VP
    (VB do)
    (NP
      (NPA
        (PRP i))
      (VP
        (VB get)
        (NPA
          (DT a)
          (NN state)
          (NN id)))))))
```

Clauses Re-ordered

Questions and answers are often related by wh-movement. An example of this phenomenon is:

> *When was the Triangle Shirtwaist Fire?*
> *The Triangle Shirtwaist Fire was March 25, 1911.*

The term "wh-movement" refers to the re-ordering of clauses in answers to questions beginning with "wh-" words ("who", "what", "where", "when", "why" and "how"). In the example above, the clause in common between the question and the answer "the Triangle Shirtwaist Fire" appears at the end of the question, and at the beginning of the answer.

Matching N-grams

An n-gram is not a structural unit, but an n-gram is an approximation of a phrase and a phrase is a structural unit. Thus we consider matching n-grams to be a measure of structural similarity. Although the question and answer pair:

When was the Triangle Shirtwaist Fire?
March 25, 1911 was the Triangle Shirtwaist Fire.

does not exhibit wh-movement, the two sentences do have a 5-gram in common ("was the triangle shirtwaist fire"). Sentences whose construction is matching at the n-gram level have an ordered set of N tokens in common.

Matching Patterns

Patterns have their roots in artificial intelligence. Research in question answering has focused on learning them from redundancy in text. In this body of work a surface text pattern is a generalized n-gram. For questions such as:

When was <PERSON> born?

A sentence containing the answer might have one of the forms:

On this date, in <BIRTHDATE>, <PERSON> was born.
<PERSON> (<BIRTHDATE> - <YEAR>)
<PERSON> , <BIRTHDATE> -
<PERSON> was born on <BIRTHDATE>
<PERSON> (b. <BIRTHDATE>)

Patterns are useful for relating the structure of questions to the structure of answers, but they can also create an equivalence class of answer sentences. The idea is that there are common structures for expressing certain types of information, and finding those structures in text allows us to extract the information.

Unrelated Construction

Although this is not an exhaustive list, for the purpose of the current research, we

consider sentences that do not fall into the categories above to have unrelated construction.

The parse structure of questions can be used to distinguish questions asking about a process and questions asking for a statement of fact [93]. Parse tree structure has also been used in machine translation, such as in IBM Models 4 and 5 [21], and the syntax-based models of Yamada and Knight [143], as well as in question answering by Echihabi and Marcu [38].

N-grams have been used to identify common phrases in document retrieval [43], or to align sentences in machine translation [8, 9, 21, 97], or in multi-sequence alignment in biological sequences, as in CLUSTALW [37], as well as in plagiarism detection [81, 12].

Finally, there is a large body of work in the use of patterns in question answering, which is discussed in section 5.1. Outside of the question answering literature the most notable use of patterns is to extract relations from unstructured text, such as *DIPRE* [18] and *Snowball* [2]. *DIPRE* extracts book title and author pairs, by defining a five-tuple {*order,urlprefix,prefix,middle,suffix*}, and using a list of book titles and authors as seed examples. *Snowball* starts with a set of seed relations (for example, <Microsoft,Redmond>) and bootstraps other examples of the same type of relation from text documents.

3.4 Conclusion

Defining relevance and similarity in this way is necessary because the meaning of "relevance" is dependent on the task at hand. It is also helpful because we can use the retrieval approach most suited to the type of similarity embodied by the notion of relevance.

Techniques developed for document retrieval have focused almost entirely on measuring similarity by the number of overlapping terms, or synonym terms. Standard retrieval models such as query likelihood, or the vector space model, retrieve sentences that contain the same words as the query. Since query likelihood is a "bag of words" model, any region on the spectrum of structural similarity is allowed - including negated sentences. This makes the query-likelihood approach appropriate for "exact match" similarity. Translation models have been applied to document retrieval [11] with moderate success, and to answer finding [10]. Depending on the data they train on, they capture matching at the synonym, and related term level. Translation models down-weight terms that are exact matches because part of their probability mass has been distributed to synonym terms.

To capture structural similarity, features of structural similarity have been used in a variety of machine learning techniques by researchers in question answering [33, 60, 109, 110], and in Novelty Detection [65]. Structural features are commonly used in tasks not related to sentence retrieval, such as information extraction and machine translation [13, 98, 100, 69]. Models such as maximum entropy models are ideal for finding structurally similar sentences because they take any number of arbitrary real-valued feature functions, and learn a probability distribution over the examples. The score produced by a maximum-entropy classifier can be integrated into a generative retrieval model as a prior probability on the data. In the next chapter, models that capture lexical and structural similarity are presented, as is a mechanism for smoothing appropriate to sentences.

CHAPTER 4

MODELS OF SIMILARITY AT THE SENTENCE LEVEL

In this chapter we introduce models for sentence retrieval. Where these models have been used in other applications, we discuss the issues specific to sentence retrieval that differentiates modeling sentence similarity from modeling other types of information. Section 4.1 discusses the query-likelihood model and smoothing. Section 4.2 discusses translation models, both generative and discriminative. Finally, Section 4.3 discusses smoothing for sentences.

4.1 Query Likelihood

The query-likelihood model was first proposed for document retrieval by Ponte and Croft [103]. This model differs from earlier models in that it does not model relevance explicitly. Query likelihood for sentence retrieval ranks sentences by the probability that the query was generated by the same distribution of terms the sentence is from:

$$P(S|Q) \propto P(S) \prod_{i=1}^{|Q|} P(q_i|S) \tag{4.1}$$

where Q is the query, $|Q|$ is the number of terms in the query, q_i is the i^{th} term in the query, and S is a sentence. In this model, we assume query terms are independent of each other given the sentence.

In this formulation, words that appear in the query that do not appear in the sentence have a probability of zero, which results in a zero probability of the query having

been generated by the sentence distribution. To resolve this problem, smoothing is introduced to give a non-zero probability to unseen words.

In Jelinek-Mercer smoothing the sentence distribution is linearly interpolated with another distribution, in our case the distribution of terms in the collection.

$$P(S|Q) \propto P(S) \prod_{i=1}^{|Q|} (1-\lambda) P(q_i|S) + \lambda P(q_i|C) \tag{4.2}$$

where Q is the query, S is the sentence, C is the collection, and λ is a real-valued parameter between 0 and 1.

4.2 Translation Models

Machine translation has its foundations in statistical speech recognition [7, 106]. It was first used in document retrieval by Berger and Lafferty [11]. In this section we discuss two types of translation models: the IBM family of models [20, 21], and conditional models for machine translation [98, 100].

4.2.1 IBM Translation Models

Brown et al. at IBM introduced a set of five statistical translation models [20, 21] derived from models of speech recognition. The idea is that a given observation is a corrupted form of the true source. In speech recognition, the observation is the wave form of the speaker, and the source is what the speaker was actually saying. In translation the observation is the sentence we wish to translate, say in Spanish, and the source is the English translation. Translation models are trained on a parallel corpus of sentences in the source language paired with sentences in the target language. In sentence retrieval, we take the source, S, to be the sentence to be ranked, and the observation, or target, Q, to be the question or query or reference sentence. To simplify the discussion, we will refer to the target as a *query* and the source as a *sentence*.

The first model, Model 1, computes the probability of the sentence given the query, assuming that all alignments are equally likely:

$$P(S|Q) = \frac{P(m|S)}{(n+1)^m} \sum_{z_1=0}^{n} \sum_{z_2=0}^{n} \cdots \sum_{z_m=0}^{n} \prod_{i=1}^{m} P(q_i|s_{z_q}) \tag{4.3}$$

where $\sum_{z_i=0}^{n}$ is the sum over all possible, equally likely alignments of a word in the query, q, to a word in the sentence s_{z_q}.

The quantity $P(m|S)$ is the probability that a sentence of length n generates a query of length m.

Equation 4.3 can be rewritten as:

$$P(S|Q) = P(m|S) \prod_{i=1}^{m} \left(\frac{n}{n+1} P(q_i|S) + \frac{1}{n+1} P(q_i|\epsilon) \right) \tag{4.4}$$

where $P(q_i|\epsilon)$ is the probability that a term q_i has no "translation" in the sentence, and thus translates to the "empty word," and

$$P(q_i|S) = \sum_{j=1}^{n} P(q_i|s_j) P(s_j|S) \tag{4.5}$$

$P(q_i|s_j)$ is the probability that the ith term of the query, q_i, translates to the jth term of the sentence, s_j, and $P(s_j|S)$ is the probability that a term s_j was generated by the sentence S.

The IBM models differ in that each introduces progressively more sophisticated alignment algorithms between a source sentence and a target sentence. Model 2 incorporates the probability of a term at position i in the target translating to a term at position j in the source sentence. Models 3, 4, and 5 model the fertility of the source sentence (the number of terms in the target that one source term translates to), and the distortion (the distance from a word in the source to a word in the target). Model 4 introduces a preference for phrases that occur together frequently.

Models 3 and 4 are statistically deficient, and Model 5 corrects this deficiency. The parameters for Model 5 are learned from models 2, 3, and 4. The details of the five models are explained in [21]. All five models are implemented in GIZA++ [3], which is freely available on the Web.

Berger and Lafferty applied Model 1 to document retrieval [11]. The notion of an alignment between a query and a document is not intuitive so Model 1, which assumes all alignments are equally likely, is the natural choice for the task of document retrieval. Berger and Lafferty also introduce a "Model 0" which makes the simplifying assumption that words translate only to themselves, with probability 1.0. In doing so, they reduce a machine translation model to the query likelihood model. The relationship between Model 1 and query likelihood follows.

The query and the sentence are in the same vocabulary, so every word has a probability of translating to itself, if nothing else. There is no possibility that a word in the query translates to an empty string, as might be the case if we were considering two different languages where tokens in one language have no translation in the other. Thus, rather than compute the probability that a word translates to the empty word, we consider the probability that it is generated by some other sentence, and we estimate it from the collection, C:

$$P(S|Q) \approx P(m|S) \prod_{i=1}^{m} \left(\frac{n}{n+1} P(q_i|S) + \frac{1}{n+1} P(q_i|C) \right) \tag{4.6}$$

Furthermore, sentences are translated to a given query, so the length of the query is constant, and the sentences do not vary widely in length. We can ignore $P(m|S)$ and conflate $\frac{n}{n+1}$ to λ and substitute into equation 4.5:

$$P(S|Q) \approx \prod_{i=1}^{m} \lambda \left(\sum_{j=1}^{n} P(q_i|s_j) P(s_j|S) \right) + (1-\lambda) P(q_i|C) \tag{4.7}$$

If term q_i translates only to itself (with probability 1) then we have Berger and Lafferty's Model 0, which is exactly query likelihood with Jelinek-Mercer smoothing, as in equation 4.2. If we retain the length coefficient $\frac{n}{n+1}$, rather than conflating it to λ, then equation 4.6 reduces to query likelihood with Dirichlet smoothing, equation 2.6, as explained in Metzler and Croft [87]. As we have seen, Jelinek-Mercer smoothing is equivalent to Dirichlet smoothing when the μ parameter in equation 2.6 is set to the length of the sentence, which is exactly the case in Equation 4.7. Thus the introduction of Jelinek-Mercer smoothing is not artificial, rather it is a natural consequence of assuming that the alignments depend on the length of the sentence, and that each word translates only to itself with probability 1.

Translation as a concept is not intuitive when we are discussing two sentences that are in the same language. Translation models for sentence retrieval treat the two sentences as if they are in completely different languages, and learn a translation table assuming the two vocabularies are distinct. Examples of terms and their "translation" terms, are shown in Table 4.1. The examples in columns one and two are from a translation table learned from a parallel corpus of questions and answer sentences. The third and fourth columns are from a translation table learned from a parallel corpus artificially constructed from terms sampled according to their mutual information with the document, paired with sentences from the document.

While a translation model for actual machine translation would be required to generate a translation sentence, in the context of retrieval we are ranking existing sentences. We call the model "Model 1" because it assumes no alignment between the query and the sentence. The translation probabilities, however, are taken from a translation table learned prior to ranking, and need not be from a table learned with Model 1. Indeed, any probabilistic dictionary would suffice. In actuality, Model 4 produces much better translations than Model 1, and in this thesis all results reported use a translation table learned from IBM Model 4, unless otherwise noted.

QA corpus				Artificial Corpus			
zebra		galileo		planet		evolution	
kenya	.064	astronomer	.099	system	.063	curriculum	.066
libya	.024	italy	.082	surface	.060	suppress	.040
camp	.023	die	.049	earth	.047	distort	.040
safari	.023	1642	.049	jupiter	.046	clergy	.037
tanzania	.017	1564	.016	solar	.041	mandate	.034
plain	.017	inquisition	.016	voyage	.041	teach	.030
lion	.012	nasa	.016	saturn	.032	fundamentalist	.023
serengeti	.012	jupiter	.016	evolve	.023	namibia	.010
mara	.012	pisa	.016	lava	.020	theory	.010

Table 4.1. Examples of "translations" of the terms "zebra" and "galileo" from a translation dictionary trained from a corpus of question-answer pairs, and the terms "planet" and "evolution" from an artificially created corpus from a random sample of news articles from the TREC data.

The advantages of the translation approach are that it generalizes a variety of data, and it doesn't depend on preprocessing such as parsing or tagging. WordNet [89] or other external resources such as online dictionaries can be incorporated in the training process of the model. One drawback is that translation models require training data, but the training data can either be learned from existing query-sentence pairs, or it can be learned from artificially constructed data. Finally, translation models, as statistical models, have all of the benefits that statistical retrieval systems have over rule-based systems, or scoring mechanisms that are empirically determined: they generalize to unseen data, and it is straightforward to incorporate other sources of information, either at training time, or as a sentence prior at the time of ranking (in the IBM models), or directly (as with the discriminative models).

A major difference between machine translation and sentence retrieval is that machine translation assumes there is little, if any, overlap in the vocabularies of the two languages. In sentence retrieval, we depend heavily on the overlap between the two vocabularies. If we use equation 4.3, the probability of a word translating to itself is estimated as a fraction of the probability of the word translating to all other

or	0.0908211	have	0.0676525
move	0.0454105	form	0.0454105
expand	0.0454105	evidence	0.0454105
code	0.0454105	circuit	0.0454105
category	0.0454105	this	0.0452817
wrote	0.0227053	sought	0.0227053
private	0.0227053	party	0.0227053
only	0.0227053	lie	0.0227053
interpret	0.0227053	ground	0.0227053
goal	0.0227053	frustrate	0.0227053
excuse	0.0227053	enact	0.0227053
cover	0.0227053	court	0.0227053
congress	0.0227053	achieve	0.0227053
conceal	0.0227052	behavior	0.0227052
personal	0.0226616	who	0.0225724

Table 4.2. Translations for the word "relationship" learned from simulated data that illustrating that self-translations often have very low probability. In this case, "relationship" translates to itself with probability less than 0.01.

words. We would hope that the model would learn a high probability for a word translating to itself. Because the probabilities must sum to one, if there are any other translations for a given word, its self-translation probability will be less than 1.0. Table 4.2 shows translations for the word "relationship" learned from a synthetically constructed parallel corpus, described in Section 6.2.1. The self-translation does not appear in this segment of the table because the probability was less than 0.01. In fact, the probabilities shown total 0.975, so the probabilities of all other translations sum to no more than 0.025.

To accommodate this monolingual condition, at the time of ranking, we separate out the case where a word in the query has an exact match in the sentence.

Let $t_i = 1$ if there exists an s_j such that $q_i = s_j$, and 0 otherwise:

$$\sum_{1 \leq j \leq n} P(q_i|s_j)P(s_j|S) \longrightarrow t_i P(s_j|S) + (1 - t_i) \sum_{i \leq j \leq n, s_j \neq q_i} P(q_i|s_j)P(s_j|S) \qquad (4.8)$$

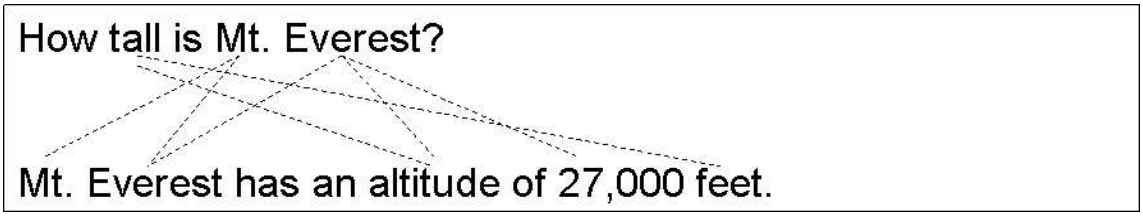

Figure 4.1. A question and answer pair, with dotted lines representing possible word pairs in a translation table under the assumptions of Model 1.

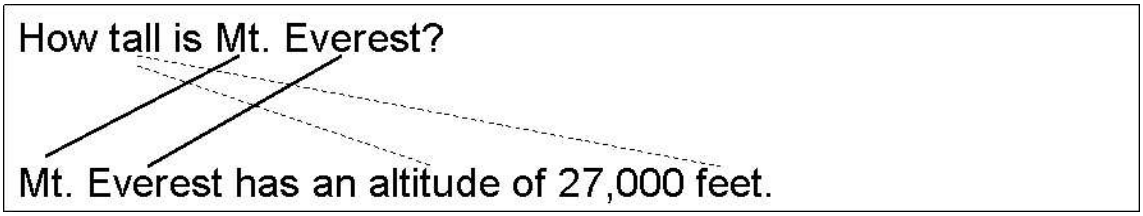

Figure 4.2. A question and answer pair, with dotted lines representing possible word pairs in a translation table. Solid lines represent matching words which have been given a probability 1.0.

Thus for sentences that have an exact matching term in the query, the matching term has probability one, and no other translations for that term are considered. Otherwise, the term probability is the sum of the translation probabilities, and all translation probabilities sum to one. In either case, the probability of a term and its translations is never greater than one for a given query-sentence pair. This model is referred to as "Model-S."

Figure 4.1 shows the question *How tall is Mt. Everest* and an answer sentence *Mt. Everest has an altitude of 27,000 feet.* The dotted lines represent word pairs that might occur in the translation table learned from Model 1. If the term *Everest* in the query is paired with the word *Everest* in sentences in the training data as often as it is paired with the words *tallest* and *elevation*, they will all have the same translation probability. In this case the word *elevation* will contribute as much to the score of a retrieved sentence as the word *Everest*, regardless of the context. Figure 4.2 shows the adjustment for Model-S that gives probability 1.0 to matching terms, and considers no other possible translations for those terms.

4.2.2 Discriminative Translation Models

In the generative model of machine translation the alignment algorithm between the source and the target is fixed, so that there is exactly one possible alignment. The fertility is fixed (in the IBM models it is limited to 25 [20]). Furthermore, it is not symmetrical. Equation 4.8 sums over j which indexes sentence terms, so it models multiple synonyms for a given query term. There is no place in the model for multiple query terms to be aligned with one sentence term. Another limitation of the generative model of machine translation is that in estimating the translations, it depends on words in the target aligning with contiguous words in the source. To accomodate gappy alignments (as in the *ne ... pas* construction in French) words are translated individually, for example the model might learn that both "ne" and "pas" translate to "not", or one of the terms is translated to the null string. Because the distortion is fixed, longer range dependencies are less likely to be captured. Sentence retrieval does not generate a new sentence, it ranks existing sentences, so we really would like to model the correlation between groups of words in both sentences, without constraints on fertility, distortion, and without being restricted to exactly one alignment.

The maximum entropy framework for machine translation was first proposed by Papineni et al. [100], to address these problems for the domain of airline reservations. In their work, the source language is an artificial formal language that shares some tokens with the target language, for example:

S_1: *What airlines fly from city1 early day1 morning?*

S_2: LIST AIRLINES SERVING FLIGHTS EARLY_MORNING FLYING_ON DAY1 FROM:CITY CITY1

Och and Ney [98] used the same approach, with the addition of an external alignment of sentence templates as a preprocessing step.

The model is an undirected graphical model, and the probability of the target (in this case a sentence, S) given the source (or query, Q) is estimated directly from the normalized sum of weighted feature functions, F_i:

$$P(Q|S) = \frac{1}{Z} e^{\sum_i \lambda_i F_i(Q,S)} \tag{4.9}$$

where

$$Z = \sum_{S \in S'} e^{\sum_i \lambda_i F_i(Q,S)} \tag{4.10}$$

Alignments can be computed as feature functions, so there is no need to commit to a single alignment strategy, or to compute the alignments in a preprocessing step. The model makes no assumptions about the independence of terms in a sentence, and the fertility and distortion are unconstrained.

These models have been used extensively in other applications, such as text segmentation [69, 84], part-of-speech tagging [121], information extraction [102], and text classification [96]. In text segmentation, information extraction, and part-of-speech tagging, the task is to assign a labeling to an input text. The labeling is external to the data, for example, the part of speech tags, or markers of the beginnings of word segments. In text classification, the label is a class, such as "e-mail" or "spam." Training is conducted on positive and negative examples of data. In the context of sentence retrieval, there are no "negative" examples, so training is done on query-sentence pairs. Furthermore, there is no external labeling. The target is the labeling of the source, and it is given, so it need not be computed at the time of decoding. Rather we are simply computing the probability of the existing labeling, given the model.

One drawback is that the computation of Z can be intractable, because Z is the sum over all possible "labelings" of a source sentence. The solution to this in other

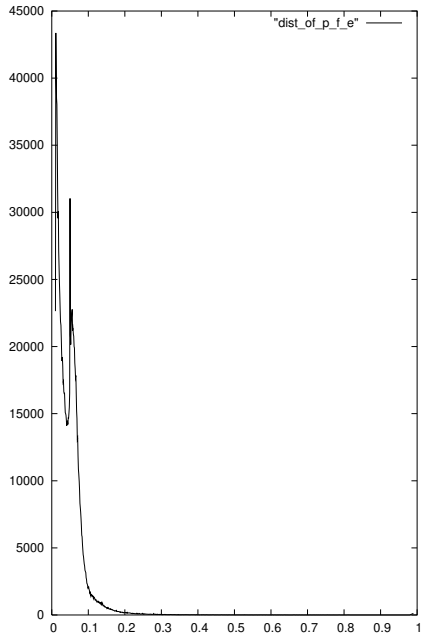

Figure 4.3. The distribution of probabilities in the translation table for IBM model 3 learned from a corpus of query-sentence pairs artificially constructed from newswire text, as described in [95]. The x-axis is the value of the probability. The y-axis is the number of times that probability occurred in the translation table.

applications, such as information extraction, is to use sampling to estimate Z. In sentence retrieval we are not finding the labeling that maximizes the probability, we are computing the probability of a specific labeling given the data. Therefore, Z can be computed directly.

4.3 Smoothing

Query terms are less likely to be present in a retrieved sentence than in a retrieved document simply because a sentence has many fewer terms. Consequently smoothing plays a larger role in the success of a sentence retrieval system. In this section we present methods for smoothing Model-S and conditional models.

4.3.1 Smoothing from the Local Context

A document is composed of one or more lexically related sentences. We would expect "good" sentences to come from "good" documents, and the term distribution in a document to be a better estimate of a constituent sentence than the term distribution in the collection as a whole. Thus we can interpolate the probability of a document having generated the query into the query likelihood framework:

$$P(S|Q) = P(S)\prod_{i=1}^{|Q|}\Big(\alpha P(q_i|S) + \beta P(q_i|D_S) + \gamma P(q_i|C)\Big) \tag{4.11}$$

where $\alpha+\beta+\gamma = 1.0$ and $P(q_i|S)$ is the probability that term q_i in the query appears in the sentence, $P(q_i|D_S)$ is the probability that the term q_i in the query appears in the document the sentence came from, and $P(q_i|C)$ is the probability that q_i appears in the collection. $P(S)$ is assumed to be uniform.

Intuitively, since documents have a discourse structure, sentences next to each other are more likely to be lexically related than sentences very far apart. To capture the local context of a sentence, we interpret the term D_S in equation 4.3.1 to be the local context of the sentence - a passage of the surrounding k sentences where k may be interpreted to encompass part or all of the document.

4.3.2 Regularization for Conditional Models

Maximum Entropy models are typically either unregularized, or regularized using a prior distribution, such as a Gaussian prior [26] or an exponential prior [50]. For the task of language modeling, Goodman [50] found that the parameters of his model resembled an exponential distribution. He proposes maximizing

$$\arg\max_{\Lambda} \prod_{j=1}^{n} P_{\Lambda}(T_j|S_j) \times \prod_{i=1}^{F} \alpha_i \exp(-\alpha_i \lambda_i) \tag{4.12}$$

and finds that this reduces the model's perplexity. Language models learn the probability of a word given the previous n words, and follow a Zipf distribution, so it seems reasonable that an exponential prior would be a better fit of a language model.

The IBM-style translation models learn the probability $P(q_i|s_j)$ that the ith term of the query, q_i, translates to the jth term of the sentence, s_j. Figure 4.3 shows the distribution of these probabilities, learned from a corpus of query-sentence pairs artificially constructed from newswire articles, as described in [95]. Unlike the language modeling task, these probabilities are for terms in different sentences, so it is not necessarily to be expected that they would follow the same distribution as the probabilities for terms in a language model. Nevertheless, the IBM Model 4 probabilities also appear to follow an exponential distribution and because the distribution of terms in both models is similarly shaped, it seems reasonable that an exponential prior would be a good fit for the sentence retrieval data as well.

4.4 Conclusion

Model-S has the benefit that it accommodates vocabulary mismatch between the query and the sentence. Unlike query expansion or pseudo-relevance feedback, terms are not "added" to the query unless they also appear in the sentence, and the probabilities for poor expansion terms are not incorporated into the sentence score. At the same time, if a term in the sentence matches a term in the query, the term is given full weight - its probability mass is not distributed among other synonym terms. For this reason Model-S is an improvement over similar models used for document retrieval. A drawback of Model-S is that it fixes an alignment and a fertility between query-sentence pairs. Conditional models make no such assumptions and allow the incorporation of arbitrary features to model structural features or non-lexical simi-

larities between query-sentence pairs. Smoothing takes into account the local context of the sentence because sentences most often appear in the context of a document.

The next three chapters demonstrate Model-S with document smoothing in applications that require sentence retrieval. Chapter 5 presents sentence retrieval for the task of question answering, for which a translation approach is ideal. Translation approaches are not a panacea for sentence retrieval and Chapters 6 and 7 discuss the conditions under which translation models are less suitable.

CHAPTER 5

SENTENCE RETRIEVAL FOR QUESTION ANSWERING

Question answering is the task of providing information that directly answers a question. It is unique among information retrieval tasks because the relevant sentence must contain information that by construction can never be provided in the question. The vast majority of research in question answering has focused on factoid questions. Factoid question answering is the task of finding the exact token or short phrase in answer to a question asking about a statement of fact.

In this chapter we demonstrate that Model-S significantly improves sentence retrieval for question answering. Further, the addition of smoothing from the local context significantly improves sentence retrieval both for sentences retrieved with Model-S and sentences retrieved with query likelihood. Finally, we show that a conditional model significantly improves results for questions beginning with the word "How", compared both to the Model-S result and the query likelihood result.

Question answering systems vary widely in their approach, but they have a typical common architecture. Questions are analyzed for their expected answer type, for example a question asking *Who is the secretary of state?* would expect a person entity as an answer; *Where is the Taj Mahal?* would expect a location entity as an answer. The question is submitted to a retrieval system to retrieve sentences (or passages) that contain the key words from the question. The candidate sentences are analyzed for lexical or structural information, and the token or phrase in the candidate sentence that is consistent with the expected answer type of the question is returned. The bulk of the research in question answering has focused on the answer

candidate selection and answer extraction modules. Clearly if the answer token or phrase does not appear in any of the candidate sentences, the system will not be able to answer the question. In this thesis we are concerned with the initial sentence retrieval step.

Systems typically use information retrieval techniques to identify passages or sentences that are similar to the question, in hopes that the answer will be located nearby. Sentences that contain the answer to a question, but do not contain terms from the question will be penalized by systems relying on terms in common with the query, and sentences that have many words in common with the question will be ranked highly regardless of whether they contain the answer token or not.

Figure 5.1 shows an example of the factoid question *When was the Red Cross founded?*. The question expects a date as an answer, and is unambiguous in meaning. Questions that expect dates as answers are among the easiest questions for an automated system to answer because dates are easily identifiable, and questions can be classified reliably as asking for dates. The sentences shown all ranked within the top 20 sentences returned by query-likelihood retrieval. None of these sentences are relevant because none contain the correct answer. After processing with stop word removal and stemming (a process which conflates words to their common forms), the question becomes *"red cross found"*. The first sentence is on the topic of the Red Cross, and the terms "red" "cross" and "found" appear in both the question and the sentence. The second sentence discusses the founding of the International Committee of the Red Cross, and contains two dates, neither of which is the correct answer to the question. The third sentence contains the words "red" and "cross" multiple times. The fourth sentence provides the data for the founding of the Chinese Red Cross Society. The last sentence is off the topic of the Red Cross completely, but contains the terms "red" and "found". Thus, even straightforward unambiguous questions present a challenge for sentence retrieval.

When was the Red Cross founded?

XIE19991221.0004-7 *The International Red Cross and Red Crescent movement is an organization whose purpose is to prevent and alleviate human suffering wherever it may be found, to protect life and health and to ensure respect for human being.*

NYT20000414.320-3 *At its founding in 1949, the International Committee for the Red Cross adopted three symbols – the red cross, the red crescent, and Iran's red lion and sun, which was dropped in 1980.*

NYT20000414.320-4 *The Red Cross has staunchly resisted adding new national or religious symbols, insisting that each new member adopt the red cross or the red crescent.*

XIE19960508.0201-4 *The Chinese Red Cross Society, which was founded in 1904, now has 150,000 grassroots organizations and over 20 million members.*

APW20000427.0309-2 *Eric "the Red" Thorvaldsson, founded Greenland colony after being kicked out of Iceland for killings.*

Figure 5.1. Five sentences that were among the top twenty sentences returned for the question "When was the Red Cross founded?" The Red Cross was founded in 1863 by Henry Dunant. The organization was named the "International Committee" and adopted the red cross emblem. The American Red Cross was founded in 1881 by Clara Barton. [59]

Questions whose answers are not a single token or named entity type (such as *What is Thalessemia?* or *How did Jimi Hendrix die?*) present additional problems of identifying the answer and defining its extent. Other questions that appear to be straightforward, such as *How far is it from Denver to Aspen?* or *Where is Glasgow?* have multiple ways to express the answer, and multiple correct answers which may or may not depend on the context of the question.[1]

We hypothesize that Model-S will improve performance for factoid question answering in two ways. The first is that the training data may contain questions that either appear in the test data, or are topically similar to questions in the test data, and as a result, the terms in the test question may "translate" to the answer token. The second is that terms in the test questions appear with related terms and synonyms in the translation table and improve the quality of the ranked list by retrieving better sentences, and therefore more of the sentences will happen to contain answer tokens. At the very least, Model-S will hurt the performance of very few questions because if the question term is absent from the translation table, Model-S reverts to query likelihood.

We hypothesize that smoothing from the document will improve performance by ranking sentences higher that come from good documents, therefore encouraging sentences that appear in the appropriate context to be ranked more highly.

Finally we hypothesize that conditional models improve results for questions that contain many common terms and stopwords. Many sentences - relevant or not - will contain these terms, and conditional models provide a mechanism for including features of the answer sentence that do not depend on the identity of a word. Fur-

[1]It is four hours from Denver to Aspen by car, or 200 miles, or 45 minutes by plane. And Glasgow is in Scotland, but it is also on the River Clyde in the West central lowlands, and it is in Iowa at latitude 40.94 and longitude -91.78.

thermore, we propose that an exponential prior is a better fit of the data than a Gaussian prior and will improve results.

The remainder of this chapter presents two approaches based on machine translation models (Section 5.2) described in Chapter 4 that significantly improve sentence retrieval for factoid question answering. The first approach (Section 5.2.1) uses Model-S. The addition of smoothing from the local context of the sentence is presented in Section 5.2.2. The second approach is presented in Section 5.2.3 and uses a conditional model. Smoothing in conditional models is accomplished by regularization, presented in Section 5.2.4. We begin with an overview of previous work in sentence retrieval for question answering.

5.1 Previous Work in Sentence Retrieval for Question Answering

The Text REtrieval Conference (TREC) provides a test bed to encourage research in question answering systems [134, 135, 136, 137, 138, 139]. Starting in 1998, the track focused largely on factoid questions. In 2001 *list* questions were added as a separate track. In 2003 *definition* questions were added, as was a passage retrieval task. In 2004 the task was altered slightly to incorporate multiple questions about a single topic, thus the topic might be "James Dean" and questions include *factoid*, *list*, and *other* questions about James Dean. The question might or might not contain the tokens "James Dean."

Sentence retrieval systems for question answering fall into three broad categories: systems that use significant linguistic analysis, systems that use surface text patterns, and systems that use term overlap or term statistics. The boundary between the use of patterns and the use of linguistic analysis is blurry, because a pattern is any linguistic regularity that is the indicator of an answer in a sentence. For the purpose of discussion, surface patterns that can be learned without any linguistic knowledge

other than named entity tags are considered separate from patterns that require extra processing, such as part-of-speech tagging, or parsing. Many systems use linguistic processing for the answer extraction component, but we are only concerned with the sentence retrieval component, so the systems described below reference only the sentence retrieval approach.

A typical example of a system using parsing to find answer sentences is IBM's submission to TREC-10 [60], which finds partial matches of dependency graphs in the parse trees of questions and answer sentences. Buchholz [22] parses and chunks the question to identify relationships between the verb chunks. The chunked question is compared to the parsed sentences, and sentences matching specific criteria are returned. The answer is identified by wh-movement markers in the matching sentences. Tanev et al. [130] match a question template (which is a generic version of the parse tree of the question, which has been re-stated as an affirmative statement) to the syntactic trees of the documents retrieved by the search engine. Regions that have sufficient similarity with the question template are used in answer extraction.

Ravichandran and Hovy [108, 58, 110] used suffix trees to find the set of longest common substrings to automatically learn surface text patterns. Greenwood and Saggion [51] use a similar approach. Kaisser and Becker [63] use hand-crafted templates to convert questions into phrases that would indicate an answer, and submit these phrases to Google as a query. They retrieve the first 40 text snippets from Google, for use in the answer extraction mechanism, and then search the AQUAINT corpus to find the sentence containing the answer (this last step is to satisfy the TREC requirements that sentences come from the corpus of documents provided to TREC participants).

A number of systems use the simple approach of retrieving documents using a standard retrieval technique (such as the vector space model), and then sentence segment the documents and retrieve sentences that contain key words from the (possibly

expanded) question [45, 25, 52, 64, 78, 109]. Several systems use a similar approach, but choose sentences that contain features of the question terms, with empirically determined feature weights [57, 32].

The WordNet [89] ontology has been used to provide synonym terms for query expansion [78, 107], as well as hypernym and hyponym relationships, useful for identifying definitions [105, 46, 51, 53, 88] or rewriting the question to turn it into a search engine query [140, 107].

In an early question answering system, Kupiec [68] proposed the use of online encyclopedias for question answering. The question was analyzed for its expected answer type, and then encyclopedia articles were heuristically scored by their degree of matching with the question. Phrases were extracted with regular expressions designed for a given answer type. Online dictionaries and encyclopedias have been used more recently to identify definitional sentences [51, 32, 55, 64, 60, 117].

A number of systems have used Web summaries to answer questions [16, 36], and then validate the answer in the TREC corpus or use the answer to expand the question before searching the TREC corpus [17, 22, 29, 32, 140, 107]. In Ravichandran and Hovy, the Internet was used to extract a database of patterns [108]. A similar approach was used by Greenwood and Saggion [51].

Translation models were first proposed for document retrieval by Berger and Lafferty [11]. Berger et al. [10] followed by using translation models for the task of answer finding. Answer finding is related to question answering, except that the system is operating on a set of questions and answers. There is no answer candidate that is not the answer to at least one question. The task is to find the existing answer that satisfies a given question. In Radev et al. [107], translation models were also used to "translate" a question into a Web query. Echihabi and Marcu [38] use translation models for the answer extraction module in their question answering system, but the

translation models are acting upon sentences retrieved by typical document retrieval methods.

5.1.1 Discussion

For the task of question answering, most research has been oriented toward extracting answers, and passage or sentence retrieval is almost an afterthought, even though in most of the papers mentioned above the negative effects of poor passage retrieval was noted. Approaches that require significant linguistic processing are potentially brittle to new question types. The more complicated the pre-processing that is done to the data, the more likely that mistakes early in the processing will negatively affect the outcome.

Systems that depend on large repositories of surface text patterns depend on having such information available, and more problematic, surface text patterns only apply to specific types of questions (such as factoid and definitional). These approaches are not effective on the TREC corpora alone, as there are not enough examples of patterns for any one question type in a corpus of that size.

WordNet [89] has been shown to be useful for identifying definitions in text, but many senses of a term are given in WordNet and it is not clear how to select the appropriate sense of a given term. WordNet was not intended for query expansion, and there is some evidence that it is not suitable for this task [133].

We propose the use of translation models for the task of sentence retrieval for question answering. Translation models capture related terms and synonyms, as might be found in WordNet or pseudo-feedback, but with appropriate weightings. Sentences contain so few terms that the context in which they appear contributes to their meaning. In the following sections we demonstrate that translation models are significantly more effective than query likelihood for sentence retrieval for question

answering, and smoothing from the local context of the sentence further improves results.

5.2 Translation Approaches to Sentence Retrieval for Question Answering

Approximately 2400 questions and answers were provided by NIST[2] as part of the TREC Question Answering Track [134, 135, 136, 137, 138, 139]. Questions numbered one through 1893, and their known correct answer sentences were used as training data. The questions from the Passages Task from TREC 2003 Question Answering Track [138], which consists of a subset of 413 of the questions numbered 1894 through 2393, were used as testing data.

For each question in the testing set, 1000 documents were retrieved from the Aquaint corpus. The Aquaint corpus consists of newswire documents from the New York Times, Xinhua News Agency and the Associated Press from the years 1998, 1999, and 2000. The documents were sentence segmented using MXterminator [111], a maximum-entropy sentence boundary detector freely available on the Web. An index of sentences was created for each question, such that each sentence was indexed as an individual "document". Where the probability of a term is estimated from a collection (as with Jelinek-Mercer smoothing for query likelihood or Model-S), it was from this collection of 1000 sentence-segmented documents. Each question index had an average of 45,000 sentences.

For an answer to be correct, it must appear in the appropriate context. A set of regular expressions is provided as part of the TREC data, and for each regular expression, a list of documents containing the answer token in the proper context is provided. The list of documents is by no means comprehensive, so the performance

[2]National Institute of Standards and Technology. http://www.nist.gov June 2006

of question answering systems is evaluated with two criteria: strict and lenient. For an answer to be correct under the strict criterion, the sentence must contain the answer token given by the regular expression, and the sentence must be in a document provided in the document list for that regular expression. Under the lenient criterion the answer token simply must be present in the sentence. Thus the strict and lenient criteria provide reasonable lower and upper bounds on performance.

The retrieved sentences would be used in further processing steps to identify the answer tokens, and it is not reasonable to conduct the processing on every sentence in the corpus. Rather we would prefer to extract the answer from the top of the ranked list. Precision at N sentences gives an indication of the quality of the top of the ranked list. For the task of question answering, we are less concerned with the overall quality of the ranked list.

The baseline in all experiments is query likelihood. This is a reasonable baseline because it represents the type of technique typically used in sentence-retrieval pre-processing modules. For each set of sentence retrieval experiments, unless otherwise noted, the parameters were set using ten-fold cross validation.

5.2.1 Model-S for Sentence Retrieval

A translation table was learned from the training data, using IBM Model 4 as implemented in GIZA++ [3].[3] The translation table for Model 4 showed performance superior to the translation table for Model 1 when trained on the same data. Words were stemmed prior to training, but were not stopped because retaining the stop words aided in the alignment between the source and the target sentences, and produced a better dictionary. Word pairs that had a probability less than .01 were eliminated from the table. The translation model trained on questions and sentences known to

[3]Work in this section is an extension of work described in [94].

Where is Glasgow?

The recession came late to Glasgow as it did to the rest of Scotland.

Bryan Jennett, a 60-year-old surgeon in Glasgow, Scotland, developed a chest pain during a flight from Washington to London.

It sold fewer than two business class seats a day in Scotland and five in the U.S.

What American composer wrote the music for West Side Story?

Bernstein was a conductor, pianist, educator, author, and composer.

Bernstein was a figure that no conductor in history has matched.

Bernstein maintained that conductors should be actors.

Figure 5.2. Examples of question-answer pairs used as translation model training data.

contain the answer. Examples of question-answer pairs used to train the translation model are shown in Figure 5.2.

A slice of the translation table is shown in Table 5.1. We can see that the terms "grunt", "tic", and "swear" translate to "tourette" and "syndrome". If the question *What condition is characterized by grunts, tics and swearing?* were present in the testing data, the sentences containing the answer "Tourette syndrome" would be more likely to be retrieved. Of the 1084 unique terms in the test set of questions, 569 were present in the translation table, however the only answer token present in the translation table is the term "red", which is the answer to the question *What color was Thomas Jefferson's hair before turning grey?*. From this we can conclude that the translation model does not improve results by "translating" a question to its answer.

Table 5.2 shows the results of retrieving sentences with Model-S as compared to query likelihood. Statistical significance is tested with respect to the baseline query-likelihood result. The results for precision at rank one were not tested for statistical significance because precision at rank one is a binary value (correct or incorrect) and neither the t-test nor the Wilcoxon sign test are appropriate. The results for precision

vitreous	material	.997
israel	history	.997
technology	material	.997
grunt	syndrome	.844
tic	syndrome	.519
swear	tourette	.513
value	prize	.493
qintex	australia	.427
spend	pound	.416
koresh	koresh	.364
monetary	worth	.331
shoe	lace	.324
lockerbie	up	.307
lockerbie	blew	.290

Table 5.1. An extract from a translation table learned from question-answer pairs provided as part of the TREC Question Answering Track from 1998 to 2002.

at 5, and precision at 20 with the lenient criteria are statistically significant using a two-tailed t-test at the $p < .05$ level. The results for precision at 15 and precision at 20 are significant using a Wilcoxon sign test at the $p < .05$ level.

Most of the questions in our testing data were unimproved by Model-S, however approximately 25% of the questions were improved at precision at rank 20. Since none of the answer tokens were present in the translation table, we believe that the improvement was due to generally better sentences being retrieved because topically related words not present in the question were given a non-zero weight. Model-S performed worse than query likelihood for approximately 16% of the questions, and the rest were unchanged. Model-S is less likely to hurt retrieval because in most cases "bad" translations are simply off the topic of the question, and the translation terms do not appear in the candidate sentences, and therefore do not affect the score.

The results under the strict criteria were not greatly affected by Model-S. Sentences appearing in documents selected by the assessors as containing the correct answer in the correct context also use the same vocabulary as the question. In this case, Model-S is of no benefit because related terms are not useful.

	Strict		Lenient	
	Query Likelihood	Model-S	Query Likelihood	Model-S
Prec @ 5	0.067	0.075	0.112	0.130⋆
Prec @ 10	0.054	0.058	0.098	0.109
Prec @ 15	0.045	0.048	0.091	0.102†
Prec @ 20	0.039	0.041	0.081	0.094⋆†
Prec @ 1	0.136	0.142	0.160	0.167

Table 5.2. The results of retrieving answer sentences with Model-S, using the strict and lenient criteria. Results indicated with a star are statistically significant at the $p < .05$ level using a two-tailed t-test. Results indicated with a dagger are statistically significant at the $p < .05$ level using a Wilcoxon sign test. The results of precision at rank one were not tested for statistical significance.

Sentences contain relatively few informative terms, and the meaning of a sentence often depends on its context. In Chapter 2 we demonstrated that larger passages have better retrieval results. In the next section we exploit the advantage of larger segments of text with smoothing, while still retaining the sentence as the unit of retrieval.

5.2.2 Document Smoothing

Sentences rarely appear as stand-alone units, and instead are almost always present in the context of longer documents. Furthermore, documents have a discourse structure, and sentences that appear next to each other in a document are often related by topic and vocabulary. Given this observation, we would expect the local context of the sentence to be a better estimate of a term's distribution than the collection as a whole, even in the case where the "collection" is a more focused set of 1000 documents retrieved for a given question.

Smoothing from the document was applied to query-likelihood retrieval and to Model-S retrieval. Table 5.3 shows the results under the strict criterion. Results for document smoothing with query likelihood are significantly better than for query likelihood. Table 5.4 shows the results of smoothing from the document under the

	QL	Model-S	QL DS	Model-S DS
Prec @ 5	0.067	0.075	0.083⋆†	0.084
Prec @ 10	0.054	0.058	0.061	0.060
Prec @ 15	0.045	0.048	0.052⋆†	0.049
Prec @ 20	0.039	0.048	0.052⋆†	0.043
Prec @ 1	0.136	0.142	0.123	0.159

Table 5.3. The result of smoothing from the document the sentence came from, evaluated with the strict criterion. Results indicated with a star are statistically significant using a two-tailed t-test at the $p < .05$ level. Results indicated with a dagger are statistically significant using a Wilcoxon sign test at the $p < .05$ level. Results for precision at rank one were not tested for statistical significance.

	QL	Model-S	QL DS	Model-S DS
Prec @ 5	0.112	0.130⋆	0.126⋆†	0.138
Prec @ 10	0.098	0.109	0.104†	0.112
Prec @ 15	0.091	0.102†	0.093	0.105†
Prec @ 20	0.081	0.094⋆†	0.087†	0.096†
Prec @ 1	0.160	0.167	0.175	0.196

Table 5.4. The result of smoothing from the document the sentence came from, evaluated with the lenient criterion. Results indicated with a star are statistically significant using a two-tailed t-test at the $p < .05$ level. Results indicated with a dagger are statistically significant using a Wilcoxon sign test at the $p < .05$ level. Results for precision at rank one were not tested for statistical significance.

lenient criterion. For both criteria, the significance results are with respect to the query-likelihood baseline. Results for precision at rank one were not tested for statistical significance.

Smoothing from the document is significantly better than the baseline query likelihood under both the strict and lenient criteria. The fact that smoothing from the document significantly improves precision under the strict criterion supports the hypothesis that smoothing from the document ranks sentences higher from documents that are topically related to the question. This is because the strict criterion requires the answer sentences to come from known "good" documents. Although the results

for smoothing from the document combined with Model-S are higher, in terms of actual scores, they are not statistically significant because the variance is higher.

Translation models rely on an alignment between the source sentence and the target sentence. They perform best when the sentences have a parallel structure, as might be the case with a sentence in Spanish and a sentence in English. In sentence retrieval, there is a greater discrepancy between the structure of a question and the structure of an answer, as well as an often large difference in the number of terms in a question and the number of terms in an answer. As a consequence, the IBM-style models rely on stopwords to guide the alignment in the sentence retrieval data, because the stopwords appear consistently in all of the sentences, and can be reliably translated.[4]

For our purposes, stopwords are not useful, and depending on them does not improve the alignment of other words, given that the word order in sentence retrieval data is not preserved between the question and the answer. IBM-style translation models are not the ideal choice for aligning data when the fertility is high or the alignments are gappy or word order is not preserved. It would be more effective to learn correlations between groups of content words, without a fixed alignment, and without the assumption of word-order preservation. Conditional models represent an improvement in machine translation because arbitrary features of the source and target sentences can be incorporated, and (possibly multiple) alignments can be calculated as features of the data. In the next section we investigate conditional models for sentence retrieval for question answering.

[4]In machine translation, dictionaries of stopwords are often incorporated in the training process to encourage the proper alignment.

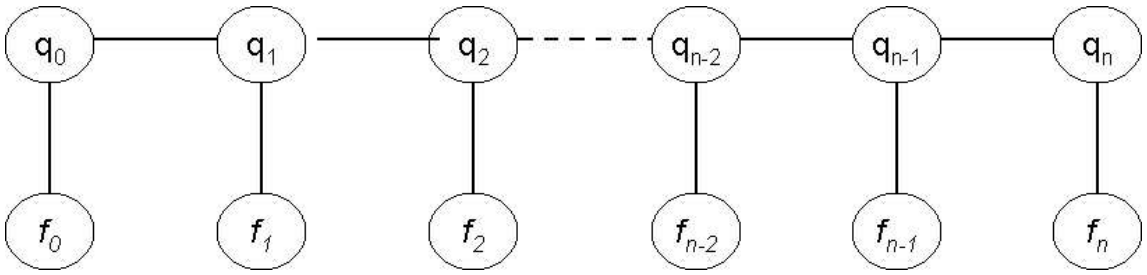

Figure 5.3. The graphical model for question-answering conditional models. The label sequence is the sequence of question words. Each question token has a feature vector associated with it that are computed over the entire answer sentence.

5.2.3 A Conditional Model for Sentence Retrieval

The conditional model computes the probability of the question given the sentence directly. We treat the question as a sequence of label tokens, and the answer sentence as an observation sequence. The task is to predict the sequence of question words given the answer sentence. Figure 5.3 shows the chain graph that represents the model. Each word in the question has a vector of features associated with it that is computed over the entire answer sentence. Only question sequences that appear in the data are considered.

We hypothesize that the conditional model will improve results for questions with many common terms. An example of such a question is *How far is it from Denver to Aspen?*. With query likelihood, many sentences mentioning Denver and Aspen will be retrieved. Model-S is unlikely to learn "200 miles" as a translation for "far". The term "mile" might be in the translation table associated with "far", but because Denver is known as the mile-high city, it is less likely documents containing the answer would be ranked higher than documents discussing Denver as the mile-high city. However, in a conditional model, a feature associating measures of distance with question words asking for a distance can be incorporated. Furthermore, whereas stopwords were necessary to compute the alignment in the IBM-style translation models, they can be completely ignored in the conditional model because no alignment is assumed.

In our data, of the 413 questions in the corpus, 364 begin with a question word *who,what,where,when,how*. Since we are predicting the sequence of question words, we trained a separate model for questions beginning with each question word. Thus, we trained a "How" model, a "Where" model, and a "Who" model. We further dissected the "What" questions into questions that began "What is the", "What is", "What are", "What do", and the rest of the "What" questions (such as "What year...", "What province..."). This eliminates all but one choice for the first token in the question sequence, and drastically reduces the choices for the second (and possibly the third) token in the question sequence. This reduces computation, and eliminates improbable question sequences that never appear in the data, such as "Where wrote" or "Who far", which would be assigned a probability if the model were trained on all of the data. Table 5.5 shows the question words, the number of questions in the training and test sets that begin with specific question words, the number of training examples, and the size of the training vocabulary, in terms of the number of unique words in the questions appearing in the training set. Training sets with a large vocabulary relative to the number of questions have more uncertainty, and take longer to train. We do not report the results for "Who" or "Where" questions because we have an insufficient number of test examples for these questions.

The features were computed over the entire answer sentence. A set of features was associated with each word in the question. Features were not associated with the first words in the question because the data was partitioned such that there was no uncertainty in the model about the labels at the beginning of the question. All features were binary features. Stopwords were given a single feature *STOP*. All words in both the question and the answer were stemmed. All content terms in the answer sentence were features of all content terms in the question. The feature *MATCH* indicated if the question word was found in the answer sentence. The feature *WN_MATCH* indicated if the question term was associated with an answer term in WordNet.

Question Words	Number Testing Questions	Number Training Questions	Number Training Examples	Training Vocab Size
what is the	36	177	3081	448
what is	15	150	4474	256
what are	9	38	806	78
what do	9	44	852	92
what (the rest)	158	238	5567	696
who is	1	46	3942	128
who was	2	60	1945	187
who (the rest)	2	88	2855	259
where is	1	73	3906	150
where (the rest)	0	40	1530	117
when	38	84	932	225
how	132	100	752	342

Table 5.5. The number of questions representing each question type in the test and training sets, and the number of examples and the number of unique words in the set of questions in the training set.

Additional features of the question type were associated with specific question words. For example, if the question began "How fast..." and words in the answer sentence indicated speed, the feature *HOW_MATCH* was associated with the term "fast". If the question began "How long..." and the answer candidate contained words indicative of length or time, the feature *HOW_MATCH* was associated to the term "long". Mentions of books, authors or writing in the answer sentence were conflated to a feature and associated with words in the question mentioning books, authors or writing. Words indicating births or deaths were treated similarly. For "Who" questions, the feature *PERSON* was associated with every content term in the question if the answer candidate contained a person entity. Years and date entities were associated with the content terms in questions beginning with "When" and "What year". Approximately twenty such features were computed on the data. Figure 5.4 shows a training example, and its features.

How fast is the speed of light?

The object is a billion light years from earth – that is the distance light travels in a billion years at the speed of 186,000 miles per second.

how	STOP
fast	HOW_MATCH WN_MATCH 000 light billion second mile 186 speed travel earth distance object
is	STOP
the	STOP
speed	WN_MATCH 000 light billion second mile 186 speed travel earth distance object speed MATCH
of	STOP
light	WN_MATCH 000 light billion second mile 186 speed travel earth distance object light MATCH

Figure 5.4. An example of the features computed from the question sequence "How fast is the speed of light" and an answer sentence.

We used the implementation of a conditional model in MALLET [85], a machine learning toolkit that is freely available on the Web, regularized with a Gaussian prior. Parameters for the Gaussian prior were optimized with the known relevant sentences, and Gaussian parameter values of 1, 100, 1000, and 5000 were tested.

At testing time, we are not generating a question sequence, given an answer sentence. Rather we are scoring an existing question sequence, given features computed over the candidate answer sentence. The answer sentences for a given question were ranked by the probability that the question would have been generated by the model, given the features of the candidate answer sentence. Table 5.6 presents the results for the strict criteria. Table 5.7 presents the results for the lenient criteria. For questions beginning with "When" the results of the conditional model were significantly worse. For questions beginning with the word "How" the results of the conditional model were significantly better.

Whereas the models for "What" questions took very little time to train, the results are not significantly better than the baseline. The reason for breaking the "What"

		Prec@5	Prec@10	Prec@15	Prec@20
what is	baseline	0.160	0.127	0.093	0.080
	conditional	0.160	0.140	0.102	0.087
what is the	baseline	0.073	0.045	0.036	0.032
	conditional	0.085	0.061	0.047	0.036
what are	baseline	0.114	0.114	0.086	0.064
	conditional	0.229	0.114	0.076	0.071
what do	baseline	0.030	0.029	0.029	0.021
	conditional	0.029	0.029	0.019	0.014
what (the rest)	baseline	0.097	0.073	0.060	0.050
	conditional	0.097	0.069	0.056	0.047
when	baseline	0.032	0.036	0.034	0.027
	conditional	0.065	0.010⋆	0.011⋆	0.013
how	baseline	0.028	0.027	0.023	0.024
	conditional	0.071⋆†	0.053⋆†	0.044⋆†	0.037†

Table 5.6. The results for the conditional model under the strict criterion, where each model has been trained on data that begins with the same question words. The results indicated with a star are statistically significant at the $p < .05$ level using a two-tailed t-test. The results indicated with a dagger are statistically significant at the $p < .05$ level using a Wilcoxon sign test.

		Prec@5	Prec@10	Prec@15	Prec@20
what is	baseline	0.267	0.220	0.169	0.137
	conditional	0.267	0.240	0.200	0.163
what is the	baseline	0.151	0.116	0.101	0.095
	conditional	0.178	0.178	0.146	0.125
what are	baseline	0.171	0.100	0.114	0.093
	conditional	0.257	0.143	0.095	0.093
what do	baseline	0.025	0.025	0.025	0.019
	conditional	0.025	0.038	0.025	0.019
what (the rest)	baseline	0.138	0.128	0.115	0.104
	conditional	0.161	0.147	0.138	0.133
when	baseline	0.094	0.079	0.084	0.066
	conditional	0.029⋆†	0.024⋆†	0.029⋆†	0.031⋆†
how	baseline	0.062	0.048	0.049	0.044
	conditional	0.126⋆†	0.090⋆†	0.077⋆†	0.067⋆†

Table 5.7. The results for the conditional model under the lenient criterion, where each model has been trained on data that begins with the same question words. The results indicated with a star are statistically significant at the $p < .05$ level using a two-tailed t-test. The results indicated with a dagger are statistically significant at the $p < .05$ level using a Wilcoxon sign test.

	Strict			
	Prec @ 5	Prec @ 10	Prec @ 15	Prec @ 20
baseline	0.0935	0.071	0.055	0.046
what is|are|do (trained separately)	0.113	0.081	0.059	0.051
what is|are|do (trained together)	0.116	0.074	0.058	0.052
	Lenient			
	Prec @ 5	Prec @ 10	Prec @ 15	Prec @ 20
baseline	0.164	0.124	0.107	0.094
what is|are|do (trained separately)	0.188	0.168	0.136	0.118
what is|are|do (trained together)	0.200	0.147	0.124	0.116

Table 5.8. Training separate models for "What is", "What is the", "What are", and "What do" questions, and aggregating the results, compared to training one model on all questions beginning with "What is (are, do)", regularizing with a Gaussian prior.

questions into subsets "What is" "What are" etc. is that it makes the time to train the model reasonable. It took approximately two weeks to train a model for "How" questions, but only took a few hours to train the model for "What is" questions. A drawback of this approach is that the amount of training data is significantly reduced, and the models may not be trained on enough data to generalize at testing time. To determine whether a larger model produced better results in the case of "What" questions, we trained a model on all of the examples for "What is the", "What is", "What are" and "What do" questions. The results for training one model on all "What" examples, compared to training four models on each type of "What" question are shown in Table 5.8. The two methods of training produce almost indistinguishable results.

The two most difficult factoid question types ("What" and "How") made up the bulk of our data, accounting for roughly 200 questions in the testing set. We examined the results for these questions separately and we show significant improvements over

	Strict		Lenient	
	baseline	conditional	baseline	conditional
Prec @ 5	0.052	0.088⋆†	0.101	0.153⋆†
Prec @ 10	0.043	0.061†	0.078	0.111†
Prec @ 15	0.035	0.049†	0.071	0.095
Prec @ 20	0.033	0.042†	0.063	0.085†

Table 5.9. The results for "What" and "How" questions using a conditional model with a Gaussian prior, using the strict and lenient criteria. Results with a star are statistically significant at the $p < .05$ level using a two-tailed t-test. Results with a dagger are statistically significant at the $p < .05$ level using a Wilcoxon sign test.

the baseline query-likelihood retrieval for these question types. Table 5.9 shows a comparison of the "What is", "What are", "What do", and "How" questions, with the baseline, under both the strict and lenient criteria.

Finally, the conditional model performs best on "How" questions, and Table 5.10 compares results for "How" questions to Model-S. The performance of the conditional model is significantly better than Model-S for precision at rank five under both the strict and lenient criteria, at the $p < .05$ level, using t-test and a Wilcoxon sign test. The conditional model is significantly better than Model-S at precision at rank 10 under the strict criterion at the $p < .05$ level, using a Wilcoxon sign test. Results for "How" questions using query likelihood are very low because "How" questions contain fewer content terms. The corpus of "How" questions has a total of 310 unique content terms for 132 questions, which gives an average of 2.35 content terms per question. The "What" questions have 697 unique content terms for 158 questions, which averages 4.41 content terms per question. The conditional model retains the advantage of Model-S to improve the quality of the retrieved results by associating related terms, and has the added advantage of including arbitrary features that relate the type of information being asked for to the type of information provided in the candidate sentence.

	Strict			
	Prec @ 5	Prec @ 10	Prec @ 15	Prec @ 20
query-likelihood	0.028	0.027	0.023	0.025
Model-S	0.041 ⋆†	0.037†	0.035	0.032
conditional	0.071 ⋆†‡	0.053⋆†‡	0.044 ⋆†	0.037 †
	Lenient			
	Prec @ 5	Prec @ 10	Prec @ 15	Prec @ 20
query-likelihood	0.062	0.048	0.049	0.044
Model-S	0.086	0.066	0.063	0.058⋆†
conditional	0.126⋆†‡	0.090⋆†	0.077⋆†	0.067⋆†

Table 5.10. Comparing the performance of the conditional model for "How" questions to query likelihood and Model-S. Results indicated with a star are statistically significant at the $p < .05$ level using a t-test, compared to the query-likelihood baseline. Results with a single dagger are statistically significant at the $p < .05$ level using a Wilcoxon sign test, compared to the query-likelihood baseline. Results with a double dagger are statistically significant at the $p < .05$ level using both a t-test and a Wilcoxon sign test, compared to the Model-S result.

5.2.4 An Exponential Prior for Sentence Retrieval

Goodman [50] proposed an exponential prior for a maximum-entropy model for the task of language modeling, based on the discovery that the parameters of his model followed an exponential distribution. We plotted the parameters of the features in the conditional model trained on 7300 training instances from the question-answering data. Figure 5.5 shows the distribution of parameter values. The x-axis is the value of a given parameter, and the y-axis is the number of parameters that have that value. Parameter values that occured only once have been omitted. The parameters for the question answering data fit an exponential distribution, which suggests that regularizing with an exponential prior - specifically a LaPlacian prior - would yield a model that was a better fit of the data.

In the following experiments, parameter values between .05 and 10 were tested, and the parameters were optimized according to the known correct answers. Table 5.11 shows the results under the strict criterion. Table 5.12 shows the same results under the lenient criterion. In most cases, the exponential prior produced

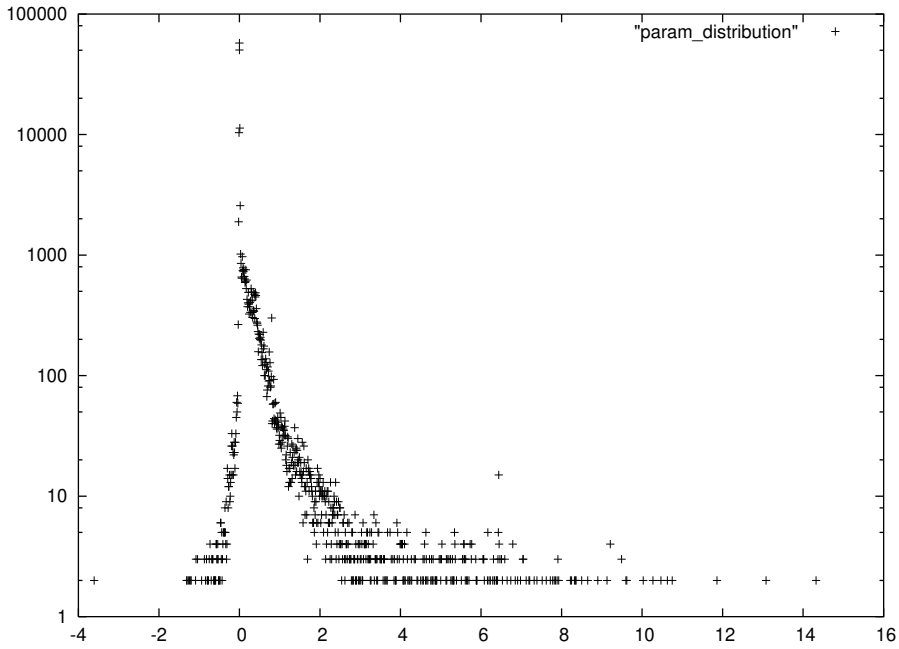

Figure 5.5. The distribution of parameters in the models trained on "What" and "How" questions. Parameter values that only occur once have been omitted.

results that were not significantly better than the results of the models regularized with a Gaussian prior.

We compared the results of aggregating the "What" questions, with training separate models, and as with the Gaussian prior, the difference between training separate smaller models and training one larger model for "What" questions was not significantly different.

Aggregating the "What" and "How" questions, which are the more difficult question types to answer, and make up the majority of questions in our data, produced equivalent results as compared to the Gaussian prior. Both were significantly better than the baseline. Table 5.13 shows a comparison of the results for "What" and "How" questions, regularizing with an exponential prior and a Gaussian prior. The results are significantly better than the baseline, but the difference between the exponential prior and the Gaussian is not significant.

		Prec@5	Prec@10	Prec@15	Prec@20
what is	baseline	0.160	0.127	0.093	0.080
	Gaussian	0.160	0.140	0.102	0.087
	Exponential	0.187	0.133	0.111	0.093
what is the	baseline	0.073	0.045	0.036	0.032
	Gaussian	0.085	0.061	0.047	0.036
	Exponential	0.091	0.058	0.051	0.039
what are	baseline	0.114	0.114	0.086	0.064
	Gaussian	0.229	0.114	0.076	0.071
	Exponential	0.171	0.086	0.057	0.057
what do	baseline	0.030	0.029	0.029	0.021
	Gaussian	0.029	0.029	0.019	0.014
	Exponential	0.057	0.043	0.029	0.029
when	baseline	0.032	0.036	0.034	0.027
	Gaussian	0.065	0.010$\star$	0.011$\star$	0.013
	Exponential	0.026	0.023	0.017	0.016
how	baseline	0.028	0.027	0.023	0.024
	Gaussian	0.071$\star\dagger$	0.053$\star\dagger$	0.044$\star\dagger$	0.037$\dagger$
	Exponential	0.063$\star\dagger$	0.055$\star\dagger$	0.042$\star\dagger$	0.035

Table 5.11. The results for the exponential prior under the strict criterion. The results indicated with a star are statistically significant at the $p < .05$ level using a two-tailed t-test. The results indicated with a dagger are statistically significant at the $p < .05$ level using a Wilcoxon sign test compared to the query-likelihood baseline.

		Prec@5	Prec@10	Prec@15	Prec@20
what is	baseline	0.267	0.220	0.169	0.137
	Gaussian	0.267	0.240	0.200	0.163
	Exponential	0.293	0.247	0.294	0.177
what is the	baseline	0.151	0.116	0.101	0.095
	Gaussian	0.178	0.178	0.146	0.125
	Exponential	0.172	0.150	0.130	0.117
what are	baseline	0.171	0.100	0.114	0.093
	Gaussian	0.257	0.143	0.095	0.093
	Exponential	0.257	0.129	0.086	0.064
what do	baseline	0.025	0.025	0.025	0.019
	Gaussian	0.025	0.038	0.025	0.019
	Exponential	0.075	0.050	0.033	0.031
when	baseline	0.094	0.079	0.084	0.066
	Gaussian	0.029⋆†	0.024⋆†	0.029⋆†	0.031⋆†
	Exponential	0.077	0.074	0.078	0.078
how	baseline	0.062	0.048	0.049	0.044
	Gaussian	0.126⋆†	0.090⋆†	0.077⋆†	0.067⋆†
	Exponential	0.126⋆†	0.093⋆†	0.079†	0.071†

Table 5.12. The results for the exponential prior under the lenient criterion. The results indicated with a star are statistically significant at the $p < .05$ level using a two-tailed t-test with respect to the query-likelihood baseline. The results indicated with a dagger are statistically significant at the $p < .05$ level using a Wilcoxon sign test.

	Strict			Lenient		
	baseline	Gaussian	Exponential	baseline	Gaussian	Exponential
Prec @ 5	0.052	0.088⋆†	0.083⋆†	0.101	0.153⋆†	0.153⋆†
Prec @ 10	0.043	0.061†	0.062	0.078	0.111†	0.113
Prec @ 15	0.035	0.049†	0.048	0.071	0.095	0.096†
Prec @ 20	0.033	0.042†	0.042	0.063	0.085†	0.088†

Table 5.13. The results for "What" and "How" questions regularizing with an exponential prior, as compared to a Gaussian prior. Results with a star are statistically significant at the $p < .05$ level using a two-tailed t-test. Results with a dagger are statistically significant at the $p < .05$ level using a Wilcoxon sign test.

	Strict			
	baseline	Model-S DS	Conditional How	Conditional What and How
Prec @ 5	0.067	0.084	0.089⋆	0.094⋆†
Prec @ 10	0.054	0.060	0.066†	0.069†
Prec @ 15	0.045	0.049	0.053†	0.055†
Prec @ 20	0.039	0.043	0.046	0.048†
Prec @ 1	0.136	0.159	0.164	0.170
	Lenient			
	baseline	Model-S DS	Conditional How	Conditional What and How
Prec @ 5	0.112	0.138	0.141⋆	0.153⋆†
Prec @ 10	0.098	0.112	0.116	0.121†
Prec @ 15	0.091	0.105†	0.108	0.113†
Prec @ 20	0.081	0.096†	0.096	0.101†
Prec @ 1	0.160	0.196	0.212	0.223

Table 5.14. Comparing the use of a conditional model for "How" questions, or "How" and "What" questions, and Model-S with document smoothing for all other questions, with the Model-S with document smoothing result. The results indicated with a star are statistically significant using a t-test at the $p < .05$ level, compared to the query-likelihood baseline. Results indicated with a dagger are statistically significant at the $p < .05$ level using a Wilcoxon sign test.

Model-S offers significant improvements for all questions, and a conditional model improves results for "How" and "What" questions. Table 5.14 shows the results of choosing the best approach, by using the conditional model for "How" and "What" questions, and Model-S for all others. Although the improvements to "What" questions alone were not statistically significant, and one could argue that the improvements to "What" and "How" questions taken together were significant only because the "How" improvements were significant, using a conditional model on "What" and "How", and Model-S with document smoothing on all other question types significantly improves results as shown in Table 5.14. When a conditional model is used for "How" questions, and Model-S is used for all other questions (including "What" questions) the results of Precision at 5 are significantly improved, and the rest of the results are better, but not significantly better.

5.3 Discussion

A perfect translation model trained on monolingual data would learn that a word always translates to itself. Therefore, for the purpose of capturing related terms and synonyms, we desire a less-than-perfect translation model. The model must be good enough to capture related terms and synonyms, but not so good as to eliminate them.

The results of retrieving sentences for the task of question answering using Model-S are significantly better than the baseline query-likelihood results under the lenient criterion. The advantage of this model is that it is elegant and efficient and requires very little in the form of external resources. As with all statistical approaches to information retrieval, it is straightforward to incorporate any number of resources, either in the training of the translation table, or as a prior probability of the data. An advantage of the Model-S approach is that it does not rely on question classification or filtering.

Smoothing from the document significantly improves retrieval over query likelihood with the strict criterion because it identifies answer sentences that are in documents that are topically related to the question. When the lenient criterion is applied, question terms that don't appear in the sentence being ranked are located in the document if the document is on the topic of the question, and smoothing from the document may help to identify answer sentences that do not happen to contain question words.

If a question term in the test set was never seen in the training set, it has no translations in the translation table, and Model-S falls back on query likelihood because it has no other mechanism to relate two sentences. The conditional model, however, allows features other than word pairs, such as features of structural or semantic relationships. If a question term in the test set was never seen in the training set, other features of the question and candidate answer sentence relate the two.

The conditional model significantly improved results for "How" questions, and "How" and "What" questions combined, which make up the bulk of our data. We are not as concerned with the poor performance on "When" questions because these questions can be selected out and handled with a different approach. A benefit of this type of sentence retrieval is that it is not necessary to classify the question by its expected answer type. Classifying question by their expected answer type is not problematic for "When" questions, but it becomes more problematic for "What" and "How" questions because there is no single named entity type these questions expect as an answer. It is possible that the same model could be used to extract the answer to the question, and features of the answer token could be incorporated into the model in addition to the features of an answer sentence. Conditional models show promise because of the facility for adding arbitrary features from any number of external sources.

Although the parameters of the model appear to follow an exponential distribution, regularizing with an exponential prior did not produce significantly different results. However, although the results were comparable to the Gaussian prior results, the time to train was much shorter. Perhaps because the gradient is steeper, the training process converges much more quickly. Given that the results are similar, the exponential prior offers this benefit over the Gaussian prior.

From a practical standpoint, we have improved the precision at rank one from one in six or seven, to one in five correct answers, and for some question types, one in three or four, but question answering systems do not act upon the top-ranked sentence. Typically they use the top 20 or 50 or 100 sentences, and depend on filtering and redundancy to identify answer tokens. Although question-answering systems typically use a retrieval system such as query likelihood to do the initial sentence retrieval, they require further processing, such as the use of gazetteers, the Internet, lists of common names, part-of-speech tagging, named-entity tagging and

other resources to find sentences more likely to contain the answer. Starting with a better set of input sentences will improve the system overall, but until the results for the sentence retrieval module have a precision at rank one equal to the state-of-the-art question answering systems that use a vast array of external resources, we cannot argue that sentence retrieval for question answering has been "solved". This is an unreasonable goal. One could, after all, do question answering and then return the sentence found to contain the answer, and claim to have improved sentence retrieval. Although we cannot claim "solving" sentence retrieval, we claim to have made a significant step in that direction.

As stated at the beginning of this chapter, the task of question answering is unique because it seeks new information not present in the question. It is not sufficient for a sentence to be on the topic of the question, for the question to be relevent. In the following chapter we investigate the use of Model-S for finding topically related sentences using data from the TREC Novelty Track.

CHAPTER 6

RETRIEVAL OF TOPICALLY RELATED SENTENCES

The TREC Novelty Track provides topics similar to topics that are part of the TREC Ad Hoc Track, as well as sentence-level relevance judgements. Because the definition of relevance as defined for the Novelty Track is similar to the notion of relevance for *Ad Hoc* document retrieval, we use the Novelty Track data. A discussion of novelty detection follows.

Novelty detection is the task of finding outliers in data that can not be explained by normal variation. It has been well studied in the pattern recognition community and has applications in medical diagnosis [73] (for example, identifying abnormalities in mammograms [129, 131]), intrusion detection [49, 82, 44, 145, 144], and new event detection [67, 15, 80, 149]. In the context of sentence retrieval, novelty detection is the task of identifying the subset of sentences relevant to a given query that present new information.

Novelty detection in text was formally defined as a track in the Text REtrieval Conference (TREC), and ran for three years from 2002 to 2004. Participants in the Novelty Track were provided a set of 50 topics (which consisted of titles, descriptions, and narratives), and a small set of documents for each topic, most of which were relevant, as well as sentence-level relevance judgments for these documents. Overviews of the TREC Novelty Tracks can be found in [54, 127, 126]. Although the task changed in the three years that it ran, the two basic subtasks were a relevance task and a novelty task. In the relevance task, systems were to retrieve the relevant sentences

from the provided documents. In the novelty task, systems were to identify the subset of relevant sentences that presented novel information.

As defined for TREC, novelty detection has somewhat conflicting goals. On one hand it looks for sentences relevant to a query, which implies the sentences are similar to the query. Once the relevant sentences are found, the sentences are compared to each other to find novel sentences, which implies the sentences would be dissimilar to each other, while remaining similar to the query.

Harman noted in the overview to the TREC Novelty Track in 2002 [54] that there is a relationship between relevance and novelty. Topics that had many relevant sentences had a small proportion of novel sentences. Topics for which there were few relevant sentences had a greater proportion of novel sentences. Relevance was described differently to assessors each year the task ran, but all of the definitions of relevance assume that, as with *ad hoc* document retrieval, sentences on the topic of the query are relevant. In this thesis we are concerned only with the relevance task. We conduct our experiments under two conditions. The first is consistent with the conditions of the TREC Novelty Track, and assumes that we are retrieving sentences from a set of 25 to 100 documents of which at least 25 are relevant. The second assumes a more realistic retrieval environment, and retrieves sentences from a subset of 1000 documents retrieved from a large corpus.[1]

In this chapter we examine the retrieval of sentences that are on the same topic as the query. We demonstrate that query likelihood and smoothing from the collection perform better than Model-S when the collection consists of a small set of mostly relevant documents. Similarly, when the collection is small and mostly relevant, smoothing from the document offers no benefit. We show that Model-S improves results significantly for short queries, given a more realistic retrieval environment where the

[1]Work in this chapter was first presented in [95].

collection is large and the relevance of the documents is unknown. Finally we show that smoothing from the local context of the sentence improves results significantly for both shorter and longer queries, when the collection is large.

The rest of the chapter is presented as follows. Section 6.1 presents previous work in sentence retrieval for novelty detection. Section 6.2 describes four ways to estimate a translation table. Section 6.3 discusses sentence retrieval from a small set of mostly relevant documents. Section 6.4 presents the results of retrieving sentences from a large corpus. The chapter ends with a discussion of the work in Section 6.5.

6.1 Previous Work in Sentence Retrieval for Novelty Detection in Text

Overviews of the TREC Novelty Track can be found in [54, 126, 127]. Most systems treated sentences as small documents and used traditional document retrieval approaches, such as the vector space model with cosine similarity and tf.idf term weights [148, 40, 1, 30], the BM25 ranking algorithm [14], or INQUERY [24, 65], which is an inference network approach to retrieval. The University of Massachusetts [71] and Carnegie Mellon University [30] both ranked sentences by the cosine similarity of the sentence vector to the query vector of tf.idf-weighted terms. Carnegie Mellon also incorporated pseudo-relevance feedback. In the vector space model, tf.idf weighting gives a higher score to longer documents because a longer document is more likely to contain query terms. To compensate for the differences in document length, the tf.idf scores can be normalized for the length of the term vectors. This similarity metric is tfc.nfx and is explained in Salton and Buckley [118]. This is the metric used in the University of Amsterdam's system [92].

Several systems use word overlap [79] or empirically driven scoring methods [35] for the relevance subtask. Other systems tried query expansion [14, 148] or pseudo-relevance feedback [1]. Meiji University [99] expanded the query with concept groups,

and then ranked the sentences by the cosine similarity between the expanded topic vector and the sentence vector.

Erkan [42] classifies sentences as relevant or not using a Maximum Entropy classifier trained on the Novelty data from 2003, and features such as the number of words in a sentence, the cosine similarity between the topic title or description and the sentence candidate, the word overlap between the topic and the sentence, and several other features based on various measures of the distance of the sentence to be ranked from the topic and from the other retrieved sentences.

The novelty subtask identifies sentences that present new information, in comparison to sentences previously seen. One approach to detecting novel sentences is to set a minimum threshold for cosine similarity with tf.idf weighted term vectors, and reject any sentence scoring higher than the threshold [1, 40, 42]. Another approach is to design a scoring method based on a function of tf.idf weights [14], or chi-squared statistics [148]. Litkowski [79] keeps a running list of text segments seen so far, and if the current sentence contains a text segment that is as yet unseen, the sentence is considered novel.

6.1.1 Discussion

In pattern classification, the identification of novel examples relies on the training set to avoid fitting the distribution to the novel events. Several systems in the Novelty Track used a training set of documents or sentences known to be relevant from data from previous TREC events. This is not ideal, because the data from TREC can differ substantially from one year to the next, thus affecting the ability of a classifier to generalize from the training set to the test set.

For the relevance subtask, we can make generalizations about the set of relevant sentences, because they share attributes common to the query and to each other. There are no attributes of the class of non-relevant sentences that can be generalized,

because they are simply any sentence that is not relevant, which is most sentences. Thus classifiers designed to discriminate between the relevant and non-relevant classes are not the ideal choice for this problem.

6.2 Translation Models for Novelty Data

Machine translation systems train on a parallel corpus of sentences in the source language paired with sentences in the target language. In the domain of question answering, a parallel corpus of questions and answer sentences was available to learn a translation table. No such corpus is available for the Novelty data. Instead we investigated several ways to create parallel data to learn a translation table. The first is to simulate a parallel corpus in much the same way that Berger and Lafferty [11] did for the task of document retrieval. The second was to create a parallel corpus of TREC topic titles, descriptions, and narratives, excluding topics that were included in the Novelty data. The third approach was to use a pair of English-Arabic, and Arabic-English lexicons to create an English-English lexicon. Finally, we used WordNet to create a probabilistic dictionary. All translation tables were trained using the implementation of IBM Model 4 in GIZA++ [3]. The details of each approach follow.

6.2.1 A Mutual Information Translation Table

Berger and Lafferty [11] created a simulated parallel corpus of queries and documents by creating a distribution of the mutual information scores of every term in a document, and then generating several queries by sampling from that distribution. The queries and the documents they were sampled from were paired to create a parallel corpus, which was used as an input to a machine translation system. In this case the parallel corpus (simulated queries paired with documents) resembled the queries and documents to be retrieved.

Following Berger and Lafferty's example, we randomly sampled 75,000 documents from the Aquaint and the TREC corpora. The Aquaint corpus consists of newswire documents from the Xinhua News Agency, the New York Times, and the Associated Press from 1998-2000. Documents were retrieved from the Aquaint Corpus for the Novelty Track in 2003 and 2004. The TREC corpus consists of newswire documents from a number of different sources including the Wall Street Journal, the San Jose Mercury News, the Financial Times, and the LA Times, from 1988 - 1994. Documents were retrieved from the TREC Corpus for the Novelty Track in 2002.

For each document, the mutual information of every term in the document was normalized to create a distribution. The mutual information of a term t in a document D is given by

$$I(t, D) = P(t, D) log \frac{P(t|D)}{P(t|C)} \tag{6.1}$$

where C is the collection, and

$$P(t, D) = P(t|D)P(D) \tag{6.2}$$

and the prior probability of a document $P(D)$ is estimated as $1/|C|$.

From this distribution a twenty-term query was sampled. The sentences in each document were ranked according to their query-likelihood score with the sampled query. The top six sentences were each paired with the neighboring sentence in the ranked list, such that every document yielded three sentence pairs. An advantage of this system is that there is no limit to the amount of data that can be generated this way.

To determine whether a translation table learned from a large amount of data yielded better results than a translation table learned from a smaller amount of data, translation models were trained on 20,000 - 200,000 sentence pairs generated from

electorate	electorate	0.997356
shipyard	deterrent	0.997352
nascar	rally	0.723774
discrepancy	women	0.691226
boston	boston	0.601832
dinosaur	dinosaur	0.581019
strauss	levi	0.501228
twilight	glitter	0.498735
puck	summer	0.498735
pastoral	dedicate	0.334979
paint	museum	0.334978
rutger	researcher	0.33249
malmo	stockholm	0.26158
mi5agent	extradition	0.246975
gala	stage	0.199654
scotsmen	tartan	0.167104
beckham	david	0.166315
january	january	0.0655233
recap	bruce	0.024299

Table 6.1. Selected terms from the translation table from synthesized data

the mutual information distribution. The results of retrieval using a translation table learned from 200,000 sentence pairs were equivalent to the retrieval results using 20,000 sentence pairs, thus the results reported for synthetic training data in this chapter are from a translation table learned from 20,000 sentence pairs.

Table 6.1 shows a slice of the translation table created from this process. Most of the terms in this translation table were self-translations. Words like "shipyard" which occurred infrequently in the data had high translation probabilities because their probability mass was spread over few translation terms. Terms such as "january" had much lower translation probabilities because they occurred often in parallel data with many other terms, each of which is given a non-zero translation probability.

6.2.2 A TREC topic Translation Table

Whereas the translation table generated from a mutual information distribution could be made arbitrarily large, so as to include any number of query terms and their

"translations", it is possible a small but targeted table such as was used for question answering would produce better results. One aspect of the mutual information translation table was that most entries in the table were self-translations. Since the TREC Ad Hoc Track data was constructed in much the same way as the Novelty Track data, it was reasonable to suppose there might be more topic and vocabulary overlap.

The TREC Ad Hoc Track data consists of topic titles, descriptions and narratives. Titles are typically between three and seven terms. Topic descriptions are roughly a sentence in length. Narratives are typically paragraph length. We created 1500 sentence pairs by aligning each topic's title with its description, each title with the narrative, and each description with the narrative. TREC topics that were contained in the Novelty data were omitted. Table 6.2 shows part of the translation table learned from the TREC data. The table was much smaller than the table learned from the mutual information distribution, and contained many fewer self-translations.

6.2.3 An English-to-Arabic-to-English Lexicon

Another source of data for a probabilistic lexicon was a pair of English-to-Arabic and Arabic-to-English lexicons. The lexicons were used in an Arabic-English cross-language document retrieval system [70], and learned from approximately three million sentence pairs from the UN corpus. The UN corpus consists of documents in Arabic and English aligned at the sentence level provided by the United Nations. To extract an English lexicon using an intermediate language, we used the formulation given in Xu et al. [142]:

$$P(e_1|e_2) = \sum_{a \in Arabic} P(e_1|a)P(a|e_2) \tag{6.3}$$

where e_1 and e_2 are the English terms, and a is a member of the set of all Arabic terms that are associated with e_1 and e_2 in the lexicons.

xenobiotic	metabolism	0.999844
perregrin	poach	0.999844
overdose	suicide	0.999844
dentistry	medical	0.834159
fearful	maul	0.742213
firewall	compute	0.724106
peugeot	automobile	0.574292
oceanfront	beachfront	0.558508
bedspread	quilt	0.499892
freelance	tax	0.345797
manipulate	develop	0.261457
presidential	candidate	0.183071
mazda	fuel	0.142507
bloodshed	africa	0.0990682
radical	social	0.0870291
bribery	official	0.0856894
contaminate	fish	0.0474268
deportation	immigrate	0.0474004
chunnel	underwater	0.03833
instigate	conflict	0.0348362

Table 6.2. Selected terms from the translation table from TREC Ad Hoc Track data

This produced a very large and high-quality translation table. Nevertheless, this dictionary was less than ideal for this task because the English words were often transliterations of Arabic words, so there might be misspellings (as in "asistance"). Or the words were transliterations of Arabic proper nouns, or words in Spanish, French or German. This dictionary was much larger than the other dictionaries, but it had many words that were not useful because they were on topics that are not discussed in the Novelty data. The topics of the documents provided by the United Nations do not cover the same topics as the topics of English-language commercial newswire articles. An extract of some of the more useful terms is shown in Table 6.3.

6.2.4 A WordNet Probabilistic Dictionary

All of the dictionaries previously described learn related terms from parallel texts. Such data allows terms such as "nascar" and "rally" to be associated with each other,

zamboni	zamboni	0.992092
ponapean	palauan	0.49905
equipment	material	0.330767
shrew	bat	0.330698
lengthiest	longest	0.249595
expediture	cost	0.247984
hotmail	communicate	0.24779
asistance	help	0.24769
petroleum	benzine	0.165349
partridge	hawk	0.165349
mortgaqe	collateral	0.165349
february	date	0.164336
nucleotide	amino	0.141728
ecotechnology	environmental	0.141728
politicide	death	0.141702
mahammad	ali	0.14145
migrant	immigrant	0.125038
aggrieve	request	0.111073
judaica	judaism	0.100146
telugu	indian	0.10002

Table 6.3. Selected terms from the translation table learned from a pair of English-Arabic, Arabic-English lexicons

or "telugu" and "indian", even though they are not synonyms. However, an artifact of the alignment is that some of the translations in the table contain non-content terms. WordNet [89] is an ontology of English terms that categorizes words by their semantic relationships. We chose to use words related by synonym, hypernym and hyponym relationships. Table 6.4 shows an extract of the probabilistic dictionary obtained from WordNet. The probabilities were arrived at by creating a parallel text of each term in WordNet paired with each term associated with it. The translation table was learned using GIZA++ [3] on the parallel data.

GIZA++ provides the facility for using a dictionary (without probabilities) to fix the alignments in the first iteration of expectation maximization, to encourage better alignments in future iterations. We investigated using the WordNet data for this purpose as well by incorporating WordNet terms as a dictionary in the training process for the mutual information data. Self-translations learned a very high probability, and all of the terms in this dictionary were synonyms, rather than related terms.

6.3 TREC Novelty Track Relevance Task

The first year the TREC Novelty Track was run, 50 topics that had between 10 and 70 relevant documents were selected from TREC topics 301-450, using the retrieval results from an actual manual run from the TREC Ad Hoc task. If a topic had fewer than 25 relevant documents, the document set was padded with non-relevant documents. If a topic had more than 25 relevant documents, a subset of 25 documents was selected. Assessors were instructed to mark a sentence relevant if it was on the topic of the query, but if two or more adjacent sentences were relevant, they were only to mark one. In spite of the fact that nearly all of the documents were relevant, the assessors marked only approximately 2% of sentences relevant.

In the following years, new topics were constructed specifically for the Novelty Track. The topic was composed by an assessor, who searched the collection for 25

zygophyllum	zygophyllum	0.994663
toxicology	toxicology	0.71048
vanessa	vanessa	0.668912
jitterbug	dance	0.537441
judaica	judaica	0.497619
uninebriated	unintoxicated	0.497332
statue	sculpture	0.497332
schnauzer	terrier	0.497332
mephistophelian	diabolic	0.397865
equivocal	ambiguous	0.372999
stegosaur	ornithischian	0.331554
mead	brew	0.331554
leopard	feline	0.331554
insouciant	casual	0.331554
hellcat	vixen	0.331554
infamous	notorious	0.248669
fungicide	chemical	0.248668
antigen	immunogen	0.248668
timpanist	percussionist	0.248666
philanthropist	altruist	0.248666
irreligious	impiety	0.198933
decry	denounce	0.165778

Table 6.4. Selected terms from the translation table created from WordNet

relevant documents. A sentence was judged relevant if it would be included in a report on the topic. In 2003, the 25 relevant documents comprised the document set. In 2004, as many documents in a ranked list as necessary were included such that 25 of them were relevant out of the top N. Thus the topics had between 25 and 100 documents, of which 25 were relevant. In 2003, 37.56% of sentences were judged relevant, and in 2004, 16.2% of sentences were relevant.

Allan et al. [5] compare the effects of the initial retrieval step on novelty detection. They show that the effectiveness of sentence retrieval is highly sensitive to the initial document retrieval. Likewise, identifying novel sentences is sensitive to the retrieval of relevant sentences Furthermore, the systems that perform best detecting novel sentences in a set of sentences that are mostly relevant are not the same systems that perform best detecting novel sentences in a set of sentences that are mostly non-relevant.

In the next section, we use both topic titles and descriptions as queries, and rank the sentences provided as part of the Novelty task. We evaluate Model-S and smoothing from the document, compared with the baseline query-likelihood performance. Query likelihood is a legitimate baseline because it is representative of the approach many systems used for the relevance task in the Novelty Track.

6.3.1 Model-S using Topic Titles and Descriptions

TREC topic titles are typically between two and seven words long. Descriptions are full sentences and average 14 words. Figure 6.1 shows an example of a TREC topic, with the title, description and narrative. Topic titles are used as queries because they more closely resemble the types of queries typically posed to search engines, whereas topic descriptions provide more information. A wider range of sentences would be relevant to the query "marine vegetation" than to the description or narrative ac-

Topic 314
Marine Vegetation
Commercial harvesting of marine vegetation such as algae, seaweed and kelp for food and drug purposes.
Recent research has shown that marine vegetation is a valuable source of both food (human and animal) and a potentially useful drug. This search will focus primarily on these two uses. Also to be considered relevant would be instances of other possible commercial uses such as fertilizer, etc.

Figure 6.1. Topic 314 from the Novelty data in 2002, which was also used in the TREC Ad Hoc track. The topic consists of a title, a description and a narrative.

companying the title, because the title is much less specific and encompasses much broader information.

When the assessors marked relevant sentences in the documents, they had available to them all three fields from the topic. Sentences were judged by two assessors, one of whom had composed the topic and found the relevant documents. The documents were presented in their entirety, and assessors were asked to identify the relevant sentences in each document, so in addition to all of the information about the topic, the assessors had all of the information about the document. Sentences were judged in context rather than in isolation. Figure 6.2 shows three sentences judged relevant to Topic 314 from Figure 6.1. In spite of the instructions to the assessors to choose only one sentence in a contiguous group, three sentences were chosen. Taken in isolation, the second sentence would not be relevant to the topic. The relevance of the third sentence would be debatable because it is not clear what entity (referred to as "It") has three 140- to 180-foot ships, and whether mowing kelp

<s docid="LA122589-0091" num="8"> While Mendocino County's seaweed farmers wrung $73,000 in sales from the ocean in 1989, San Diego-based Kelco was harvesting millions of dollars in giant kelp.</s>

<s docid="LA122589-0091" num="9"> Kelco, a division of pharmaceuticals giant Merck & Co., is the largest company of its kind in the world.</s>

<s docid="LA122589-0091" num="10"> It has three 140- to 180-foot ships, kelp cutters that mow the tops off the fastest-growing and tallest plants in the ocean.</s>

Figure 6.2. An example of three sentences judged relevant to the topic 314 shown in example 6.1

is part of a commercial operation for food or drug production, which is required by the topic specification.

In the example in Figure 6.2 none of the sentences would score highly using query likelihood with the topic title as a query, because none of the sentences contain the words "marine" or "vegetation". In fact, query likelihood and Model-S return nearly identical results, except that lower down in the ranked list, Model-S presents sentences containing the words "algae" and "ocean". This is to be expected because Model-S gives the highest weight to query terms present in the sentence, and lesser weight to query terms and sentence terms that appear together in the translation table.

Table 6.5 shows the results for Model-S using translation tables from the TREC topics (the column labeled "TREC") and synthetic data (the column labeled "MI"). As expected, the significant improvements are in the results at the bottom of the ranked list. Model-S significantly improves the average precision, indicating that the ranked list as a whole is improved. The average precision is an appropriate metric for the Novelty task, because the entire ranked list might be examined to identify the novel sentences. The "MI" result is only slightly better than the "TREC" result, in spite of its size.

	Query Likelihood	Model-S (TREC)	Model-S (MI)
Relevant	25248	25248	25248
Rel Retrieved	24162	24328⋆	24347⋆
Prec @ 5	0.3338	0.3311	0.3230
Prec @ 10	0.3561	0.3541	0.3453
Prec @ 15	0.3608	0.3622	0.3554
Prec @ 20	0.3618	0.3574	0.3578
Prec @ 30	0.3581	0.3554	0.3687⋆
Prec @ 100	0.3423	0.3459	0.3540⋆
Prec @ 200	0.3132	0.3161⋆	0.3295⋆
Prec @ 500	0.2446	0.2481⋆	0.2546⋆
Prec @ 1000	0.1633	0.1644⋆	0.1645⋆
R-Prec	0.3339	0.3484⋆	0.3568⋆
Ave Prec	0.3260	0.3323 ⋆	0.3445⋆

Table 6.5. Comparing translation model-based retrieval with title queries, from the documents provided as part of the Novelty task. "TREC" and "MI" are two ways to estimate a translation model. Results indicated with a star are significant at the $p < .05$ level with a two-tailed t-test.

Table 6.6 presents the results of Model-S using topic descriptions as queries. The results at the top of the ranked list are significantly worse. To understand why this might happen, consider that the assessors composing the topics and marking relevant sentences were the same assessors who hand-picked the relevant documents, and wrote the descriptions. In some cases, the topic description was nearly identical to sentences appearing in the data. Figure 6.3 shows an example of topic number N55. The topic description has a high degree of overlap with sentences in the data. In situations where the query and the relevant sentences have a significant vocabulary overlap, Model-S is unlikely to perform better than query likelihood. Model-S will harm performance by promoting sentences with related terms, when the sentences with exact matching terms are relevant. In this case, the query-likelihood results for precision at N are on par with the results for document retrieval. Nevertheless, the results at the bottom of the ranked list are significantly better with Model-S, as is the Relevant-retrieved evaluation metric.

Topic N55
India and Pakistan Nuclear Tests
On May 11 and 13, 1998 India conducted five nuclear tests; Pakistan responded by detonating six nuclear tests on May 28 and 30th. This nuclear testing was condemned by the international community.
India conducted a total of five nuclear tests on May 11 and 13. In response, Pakistan claimed to have detonated six nuclear devices last week. The international community condemned the nuclear tests by the two countries

Figure 6.3. The Topic N55 description (the first box after the topic title) has a high degree of overlap with sentences from the data (the bottom box). In this situation, Model-S will not outperform query likelihood.

	Query Likelihood	Model-S (TREC)	Model-S (MI)
Relevant	25248	25248	25248
Rel Retrieved	24375	24555⋆	24524⋆
Prec @ 5	0.5230	0.4716⋆	0.4865⋆
Prec @ 10	0.4865	0.4588⋆	0.4791
Prec @ 15	0.4698	0.4468⋆	0.4586
Prec @ 20	0.4642	0.4416⋆	0.4483⋆
Prec @ 30	0.4475	0.4333	0.4376
Prec @ 100	0.3961	0.3832⋆	0.3924
Prec @ 200	0.3491	0.3491	0.3518
Prec @ 500	0.2587	0.2617⋆	0.2622⋆
Prec @ 1000	0.1647	0.1659⋆	0.1657⋆
R-Prec	0.3843	0.3797	0.3796
Ave Prec	0.3776	0.3763	0.3760

Table 6.6. Comparing translation model-based retrieval with description queries, from the documents provided as part of the Novelty task. "TREC" and "MI" are two ways to estimate a translation model. Results indicated with a star are significant at the $p < 0.05$ level with a t-test.

Considering that the assessors wrote the topics and hand-picked the documents, and then judged the sentences in context, it is reasonable to ask if exploiting the local context of the sentence improves retrieval. In the next section we present smoothing from the local context, in this case the document the sentence came from.

6.3.2 Smoothing from the Document

Documents are considered relevant because they contain relevant information. It seems reasonable to assume that relevant sentences come from relevant documents, and that the distribution of terms in documents would give a better estimate of the worth of a query term than the distribution of terms in the collection as a whole. In the experiments in this section, the collection is the set of documents provided as part of the Novelty Track data. To incorporate the local context of the sentence, we interpolate the document score with the sentence score of a given query term, as described in Chapter 4.

Table 6.7 shows the results of smoothing from the document for topic titles and topic descriptions. The precision at rank five is significantly improved for title queries, but the results overall are equivalent to the baseline result with smoothing from the collection.

Recall that the definition of relevance in this case is that the query is on the topic of the document, but the majority of documents for each query in this corpus are on-topic. In this environment, smoothing from one document gives equivalent results to smoothing from 25 on-topic documents. The lack of benefit of smoothing from the document is an artifact of the way the corpus was constructed.

The Novelty Track assumes that document retrieval has, in essence, been solved and that a retrieval system is able to return 25 relevant documents in the top 25 to 100 results. The topics were written by the same person who chose the relevant documents, and judged the sentences in the document. Because of this many of the

	Title		Description	
	Baseline	Doc Smooth	Baseline	Doc Smooth
Relevant	25248	25248	25248	25248
Rel Retrvd	24162	24325⋆	24375	24389
Prec @ 5	0.3338	0.3757⋆	0.5230	0.5216
Prec @ 10	0.3561	0.3730	0.4865	0.4845
Prec @ 15	0.3608	0.3788	0.4698	0.4703
Prec @ 20	0.3618	0.3743	0.4642	0.4635
Prec @ 30	0.3581	0.3743	0.4475	0.4475
Prec @ 100	0.3423	0.3445	0.3961	0.3959
Prec @ 200	0.3132	0.3110	0.3491	0.3495
Prec @ 500	0.2446	0.2437	0.2587	0.2590
Prec @ 1000	0.1633	0.1644⋆	0.1647	0.1648
R-Prec	0.3339	0.3401⋆	0.3843	0.3840
Ave Prec	0.3260	0.3239	0.3776	0.3773

Table 6.7. Comparing the query-likelihood baseline with smoothing from the document, using title and description queries and the set of 25 documents from the TREC Novelty track. Results indicated with a star are significant using a t-test at the $p < .05$ level.

topic descriptions are very similar to sentences in the data. When the sentences to be retrieved are very similar to the queries, Model-S can not improve upon query likelihood. Furthermore, since the collection consists mostly of relevant documents, smoothing from the document produces results equivalent to smoothing from the collection of 25 relevant documents. Model-S is effective for title queries because title queries start with very few terms, and Model-S provides additional related information.

In a more realistic retrieval setting, the collection is very large and consists of documents on a wide range of topics. In this situation we typically have no information about which documents are relevant, and which are not. In the following section we investigate Model-S and smoothing from the document retrieving from the Aquaint and TREC corpora.

6.4 Sentence Retrieval from a Large Corpus

To create a retrieval environment more representative of typical retrieval systems, where the relevance of documents is unknown, we retrieved 1000 documents for each query from the TREC corpus and the Aquaint corpus. The documents for the Novelty topics from 2002 were retrieved from the TREC corpus. Documents for the remainder of topics were retrieved from the Aquaint corpus.

The top 1000 documents contain 93% of the relevant sentences, and the top 500 documents contains 87% of relevant sentences. In addition, the topic titles produced better document retrieval results than the topic descriptions, so the sentence retrieval described in this section is from a subset of 1000 documents retrieved using query likelihood from the topic titles.

It is not possible to provide relevance judgments for every document in the Aquaint or TREC collections. Providing sentence-level judgments, even for the subset of 1000 documents, is cost and time prohibitive, so the sentence-level judgments provided for

	2002	2003	2004
% Sents Relevant	2%	37.56%	16.2%
Prec. at 1	.060	.240	.240
Prec. at 5	.048	.136	.176
Prec. at 10	.050	.108	.182
Prec. at 15	.048	.103	.169
Prec. at 20	.043	.105	.157
Prec. at 1000	.007	.051	.052
Recall	.456	.445	.614

Table 6.8. Comparing query-likelihood retrieval for each year of the TREC Novelty Track. Very few of the sentences in the 2002 data were marked relevant. The topics in 2003 and 2004 were constructed by assessors who looked for relevant documents by hand in the Aquaint corpus. Sentences were retrieved from a subset of 1000 documents retrieved from the TREC and Aquaint corpora.

the Novelty Track were applied to these experiments as well, recognizing that they cover at most 100 documents of the 1000 retrieved.

The number of sentences marked relevant by the assessors differed greatly each year that the Novelty Track ran. Table 6.8 shows the effect on retrieval, broken down by year. The retrieval is from the subset of 1000 documents whose relevance is unknown. The evaluation is clearly affected by the number of sentences marked relevant. The baselines in this section are low because sentences are considered non-relevant if they did not appear in the set of 25 documents chosen by the assessor. Thus, the results reported in this section are a lower bound on true performance.

The translation tables learned from the synthetic training data described in Section 6.2.1 and from the TREC topics, described in Section 6.2.2 produced the best results for the relevance task for the Novelty data (Section 6.3). Tables 6.9 and 6.10 show the results of Model-S using the mutual information translation table and the translation table learned from TREC data. The TREC translation table fared slightly better than the mutual information translation table, but results for both are sig-

	Query Likelihood	Model-S (TREC)	Model-S (MI)
Relevant	25248	25248	25265
Rel Retrieved	5816	5782	5614
Prec @ 5	0.077	0.078	0.081
Prec @ 10	0.082	0.076⋆†	0.074 †
Prec @ 15	0.085	0.079	0.078
Prec @ 20	0.078	0.077	0.074
Prec @ 30	0.078	0.078	0.073
Prec @ 100	0.069	0.070	0.065
Prec @ 200	0.062	0.064	0.060
Prec @ 500	0.049	0.050	0.048
Prec @ 1000	0.039	0.039	0.038
R-Prec	0.065	0.066	0.063
Ave Prec	0.025	0.026	0.024

Table 6.9. Comparing the use of a small but topically focused translation table learned from TREC topics, and a large translation table learned from the distribution of mutual information scores of terms in random documents, to the query-likelihood baseline on queries from topic titles. Results indicated with a star are significant at the $p < .05$ level using a t-test, and results indicated with a dagger are significant at the $p < .05$ level with a Wilcoxon sign test.

nificantly worse than the baseline query likelihood, for both topic titles and topic descriptions.

To understand why the results would be so poor, consider that the mutual information translation table was computed over arbitrary data that was not necessarily topically related. Therefore it contains many translations for terms that do not appear in the set of relevant sentences. Model-S will perform significantly worse than query likelihood when the "bad" translations appear in retrieved sentences. In the Novelty Track document set, the documents were on topic, and therefore "bad" translations were contained in few, if any, sentences, and Model-S in the worst case had no effect on retrieval. In the larger corpus, "bad" translations are much more likely to appear in one of the 45,000 sentences associated with a given query, and those sentences will be ranked higher. Therefore, Model-S has a much greater potential to

	Query Likelihood	Model-S (TREC)	Model-S (MI)
Relevant	25248	25248	25265
Rel Retrieved	5396	5303	4632 †
Prec @ 5	0.1216	0.1392	0.090
Prec @ 10	0.1135	0.1095	0.084 ⋆†
Prec @ 15	0.1072	0.0977	0.082 ⋆†
Prec @ 20	0.1020	0.0936	0.083
Prec @ 30	0.0939	0.0874	0.075
Prec @ 100	0.0720	0.0705	0.058 ⋆†
Prec @ 200	0.0621	0.0593 †	0.049⋆†
Prec @ 500	0.0467	0.0468	0.039 †
Prec @ 1000	0.0365	0.0358	0.031 †
R-Prec	0.0674	0.0671 †	0.052 ⋆†
Ave Prec	0.0249	0.0264	0.017 ⋆†

Table 6.10. Comparing the use of a small but topically focused translation table learned from TREC topics, and a large translation table learned from the distribution of mutual information scores of terms in random documents, to the query-likelihood baseline on queries from topic descriptions. Results indicated with a star are significant at the $p < .05$ level using a t-test, and results indicated with a dagger are significant at the $p < .05$ level with a Wilcoxon sign test.

harm performance. Since both translation tables performed badly, we must conclude that size was not the primary contributor to their failure.

Arabic is a highly inflected language, and stemming is a significant research issue for highly inflected languages [72]. Since the Arabic-English lexicons we had available were not stemmed, the results presented for the Arabic-English lexicons are compared to a query-likelihood baseline which was also not stemmed.

Tables 6.11 and 6.12 show the results on topic titles and descriptions of the English-Arabic-English translation table. The results for title queries are significantly better for precision at five documents retrieved. The ranked list overall is significantly improved, as indicated by the average precision. In addition, the results for precision at 20 through 1000 documents retrieved is also significantly better. This implies that a large, high-quality translation table improves results for title queries, when retrieving sentences from a large collection.

The results of description queries are not as dramatic, but show a trend toward improvement. Description queries contain significantly more terms to begin with, and in a translation environment, more translations of query terms would be found in the retrieved sentences. Therefore, the improvements to be gained by accommodating a vocabulary mismatch are mitigated by the retrieval of sentences containing "bad" translation terms.

As a large high-quality lexicon improved results, especially for topic titles, perhaps a smaller high-quality translation table would also improve results. The WordNet dictionary was used in two ways. The first was as a stand-alone translation table. The second as a dictionary in the training of the synthetic data to encourage proper alignments. Neither approach yielded improvements for either title or description queries. WordNet learns synonyms, but contains many unusual words that don't appear often in the data. The results for description queries are shown in Table 6.13 Results for title queries are comparable, but are not shown.

	QL	English-Arabic
Relevant	25265	25265
Rel Retrieved	5469	5690 †
Prec @ 5	0.0779	0.0993 ⋆†
Prec @ 10	0.0779	0.0872
Prec @ 15	0.0702	0.0738
Prec @ 20	0.0671	0.0742 ⋆†
Prec @ 30	0.0647	0.0700 †
Prec @ 100	0.0577	0.0644 ⋆†
Prec @ 200	0.0547	0.0592 ⋆†
Prec @ 500	0.0452	0.0479 ⋆†
Prec @ 1000	0.0367	0.0382 †
R-Prec	0.0540	0.0609 ⋆†
Ave Prec	0.0194	0.0232 ⋆†

Table 6.11. Comparing the results of Model-S with a large high quality translation table learned from a pair of English-Arabic lexicons to the query-likelihood baseline for topic titles. Results indicated with a star are statistically significant using a t-test at the $p < .05$ level, and results indicated with a dagger are statistically significant using a Wilcoxon sign test at the $p < .05$ level.

	QL	English-Arabic
Relevant	25265	25265
Rel Retrieved	5056	5257⋆
Prec @ 5	0.1154	0.1087
Prec @ 10	0.1040	0.1054
Prec @ 15	0.0971	0.1002
Prec @ 20	0.0889	0.0966†
Prec @ 30	0.0852	0.0919†
Prec @ 100	0.0667	0.0698
Prec @ 200	0.0570	0.0588
Prec @ 500	0.0441	0.0449
Prec @ 1000	0.0339	0.0353⋆
R-Prec	0.0611	0.0607
Ave Prec	0.0206	0.0218

Table 6.12. Comparing the results of Model-S with a large high quality translation table learned from a pair of English-Arabic lexicons to the query-likelihood baseline for topic descriptions. Results indicated with a star are statistically significant using a t-test at the $p < .05$ level, and results indicated with a dagger are statistically significant using a Wilcoxon sign test at the $p < .05$ level.

	QL	MI + WN	WN Only
Relevant	25248	25265	25265
Rel Retrieved	5396	4759†	5146
Prec @ 5	0.1216	0.1154	0.1141
Prec @ 10	0.1135	0.1094	0.1020
Prec @ 15	0.1072	0.1025	0.0872†
Prec @ 20	0.1020	0.0933	0.0876
Prec @ 30	0.0939	0.0828	0.0814
Prec @ 100	0.0720	0.0621	0.0662
Prec @ 200	0.0621	0.0532†	0.0594
Prec @ 500	0.0467	0.0410	0.0446
Prec @ 1000	0.0365	0.0319†	0.0345
R-Prec	0.0674	0.0582	0.0617
Ave Prec	0.0249	0.0201	0.0218

Table 6.13. Comparing the use of WordNet as a translation table, and as a dictionary during the training of a translation table from simulated data. Results indicated with a dagger are statistically significant at the $p < .05$ level using a Wilcoxon sign test. The results are not significant using a t-test.

Using WordNet as a dictionary to encourage the proper alignment in the mutual information data had a positive effect, considering how poor the mutual information data was on its own. Recall from Table 6.10 that the performance was significantly worse, and in this data, the performance is comparable to the baseline.

Since the set of relevant sentences comes from a very small set of relevant documents, we would expect smoothing from the document to improve results for retrieval from a large corpus, in spite of being ineffective on a small, mostly relevant corpus. In the next section we investigate the amount of context to smooth from, and present results for smoothing from the document when retrieving from a large corpus.

6.4.1 Smoothing from the Document

Smoothing from the document did not improve results for the description queries when retrieving from a small subset of mostly relevant documents, because estimating the background probability of a term from one relevant document was equivalent to estimating the background probability of a term from twenty-five relevant documents.

In a large corpus, the background probabilities would be estimated from a large number of documents, most of which are non-relevant. Even in the case of retrieving sentences from a smaller subset of 1000 documents, most of the documents will be non-relevant. Because of this, we would expect estimating the smoothing probability from the local context of the sentence to produce better results than smoothing from a large corpus of mostly non-relevant documents.

We investigated the appropriate amount of context from which to smooth. Table 6.14 shows the results of smoothing from the surrounding five and eleven sentences, and from the entire document, using description queries. The results are statistically significant using a t-test at the $p < .05$ level, regardless of the amount of context. The number of relevant sentences is almost double, given any number of retrieved sentences, when the sentence scores are smoothed from the entire document. Table 6.15 shows even more dramatic improvements in the ranked list for topic titles. Notice also that smoothing from the document raises the precision of title queries to be on par with the precision for description queries.

6.5 Discussion

If the document set to be retrieved from is mostly relevant and small, smoothing from the collection produces results equivalent to smoothing from the document because most of the documents are on-topic. In this situation, smoothing from the local context does not change the retrieval results by promoting sentences from "good" documents, because the majority of documents are "good".

In the Novelty Track documents, the document set was not only small and mostly relevant, but shared the vocabulary of the topic descriptions. If there is no vocabulary mismatch between the query and the sentences, Model-S will not offer a benefit over query likelihood.

	QL	5 Sents	11 Sents	Entire Doc
Relevant	25248	25248	25248	25265
Rel Retrieved	5441	6986⋆	7735⋆	9624⋆
Prec @ 5	0.1203	0.1527⋆	0.1541⋆	.2362⋆
Prec @ 10	0.1122	0.1446⋆	0.1419⋆	.2128⋆
Prec @ 15	0.1018	0.1329⋆	0.1405⋆	.2000⋆
Prec @ 20	0.0973	0.1311⋆	0.1345⋆	.1893⋆
Prec @ 30	0.0890	0.1191⋆	0.1286⋆	.1743⋆
Prec @ 100	0.0732	0.0935⋆	0.1006⋆	.1335⋆
Prec @ 200	0.0621	0.0791⋆	0.0869⋆	.1148⋆
Prec @ 500	0.0478	0.0613⋆	0.0679⋆	.0862⋆
Prec @ 1000	0.0368	0.0472⋆	0.0523⋆	.0646⋆
R-Prec	0.0672	0.0881⋆	0.0933⋆	.1226⋆
Ave Prec	0.0257	0.0410⋆	0.0485⋆	.0749⋆

Table 6.14. Comparison of smoothing context on description queries, retrieving sentences from the top 1000 documents. Results indicated with a star are significant at the $p < .05$ level using a t-test.

	Query Likelihood	Doc Smooth
Relevant	25265	25265
Rel Retrieved	5810	9924⋆
Prec @ 5	0.0765	0.2268⋆
Prec @ 10	0.0805	0.2262⋆
Prec @ 15	0.0814	0.2192⋆
Prec @ 20	0.0765	0.2124⋆
Prec @ 30	0.0765	0.2007⋆
Prec @ 100	0.0675	0.1638⋆
Prec @ 200	0.0614	0.1330⋆
Prec @ 500	0.0490	0.0963⋆
Prec @ 1000	0.0390	0.0666⋆
R-Prec	0.0646	0.1379⋆
Ave Prec	0.0243	0.0796⋆

Table 6.15. Comparison of document smoothing to query likelihood with title queries retrieving sentences from the top 1000 documents. Results indicated with a star are significant at the $p < .05$ level using a t-test.

For the more realistic retrieval environment where no document relevance information is available, Model-S improves results for both title and description queries significantly when a large high-quality translation table is available. When the translation table is noisy, as with the mutual information translation table, terms in the query may be associated with "bad" translation terms, causing non-relevant sentences to be higher in the ranked list. This is not an issue in a small corpus of relevant documents, because there are few sentences that are topically unrelated to the query.

In a large corpus where many of the documents are off the topic of the query, smoothing from the document results in a large gain in precision for both query types, but the result is especially dramatic for title queries. Title queries are improved because they contain very few words to begin with, and sentences relevant to title queries contain much more information than the query, they therefore benefit from the inclusion of added information in the form of context and translations.

Description queries are about a sentence in length, and contain more words to be translated, and perhaps mistranslated. As discussed earlier in this chapter, many of the topic descriptions are strikingly similar to relevant sentences in the data, and therefore need no additional information.

In the next chapter we examine sentence retrieval for Information Provenance. Information Provenance seeks to characterize the degree of similarity between a reference sentence and a retrieved sentence. The Information Provenance data allows us to tease out the relationship between similarity and relevance we have observed in the Novelty data.

CHAPTER 7

INFORMATION PROVENANCE

Information provenance seeks the origin of a specific piece of information in a corpus of documents. Because the documents are assumed to be of a certain typical length, they contain a broad range of information, possibly on a number of topics. Information provenance looks at the similarity between a reference sentence, and a set of candidate sentences to determine which sentences are co-derived, which sentences simply contain the same information, and which contain new information, by characterizing the degree of similarity. The unit of evaluation is a sentence because larger passages contain too much information, and smaller snippets of text cannot be assessed reliably for co-derivation.

Unlike earlier chapters, the research represented in this chapter does not demonstrate the model that performs best for the task of information provenance. Rather, this chapter is the extension of preliminary work conducted by Metzler et al. [87]. We show that models that rank sentences based on their term-frequency are not appropriate for the task of information provenance because these types of models do not have sufficient discriminative power.

A study of sentence retrieval for information provenance is a natural follow-on to the study of retrieval of topically related sentences. As discussed in Chapter 3 sentences have a more fluid sense of topicality than do documents, because they carry relatively little information. Whether a sentence appears in isolation or in context affects its meaning. The Novelty Track relevance task implicitly assumes that the relevant sentences are also similar, by the way the data was constructed,

but this need not be true in all cases. Information provenance seeks to characterize the similarity of on-topic sentences, and to distinguish between those sentences that express the same information but in a completely new way from those sentences which express new information.

Sentence retrieval for novelty detection and question answering live on opposite ends of the spectrum of relevance. Novelty detection requires a relevant sentence to be on the general topic of the query, and question answering requires a relevant sentence to satisfy the information need directly. Information provenance lives in the middle of the relevance spectrum, and requires that the sentences be on the specific subtopic as the query.

Relevant sentences in novelty detection are assumed to be very similar to the query. Question answering makes no requirements about the similarity of the sentence to the question, but in practice requires sentences that share the vocabulary of the question and appear in a context that shares the vocabulary of a question. Both tasks live on the "identity" end of the similarity spectrum. Information provenance allows us to venture away from the "identity" end of the spectrum, by characterizing the degree of similarity. The task is to find models that are capable of discerning topically related information that is dissimilar.

Based on what we know of query likelihood (and other term-frequency based models), and Model-S from the other sentence retrieval tasks investigated in this thesis, we propose that the family of language-model based retrieval approaches are not appropriate for this task. In this chapter we investigate query likelihood, and translation approaches (Model-S and Model-1) and demonstrate that they are simply not able to identify topically related information that is expressed in a dissimilar fashion.

In the remainder of the chapter we present previous work related to information provenance in Section 7.1. To evaluate retrieval models for the task of information

provenance, it was necessary to construct a new data set because there is no data available at this time. Section 7.2 describes the data that was used, and presents the relevance assessment study. Section 7.3 presents experimental results for sentence retrieval for information provenance. Finally, Section 7.4 presents a discussion of the results and the task.

7.1 Previous Work in Information Provenance

As a completely new area, there has only been one other preliminary study that we know of at the time of this writing, conducted by Metzler et al. [87]. However, there has been other work related to text re-use, such as topic detection and tracking and plagiarism detection. Previous work is presented in these areas, followed by a discussion of Metzler et al.

Text re-use covers a wide variety of technologies including topic detection and tracking, plagiarism detection, and information provenance. Plagiarism detection and topic detection and tracking have been well-studied. Information provenance is an emerging discipline concerned with finding the origins of a piece of information in a collection of documents. It differs from plagiarism detection, because the information might be re-stated in any number of ways, whereas plagiarism detection looks for exact matches in portions of documents. Information provenance differs from topic tracking in that topic tracking looks for repeated mentions of the same topic in an evolving stream of news, whereas information provenance looks backward through a stationary corpus for evolving mentions of a piece of information. Another substantial difference between topic tracking and information provenance is that topic tracking considers the topic of the document as a whole, and information provenance looks at similarity at the sub-document level. Nevertheless, some of the approaches to topic detection and tracking, and plagiarism detection may be appropriate for information provenance, and in this section we present previous work on these technologies.

7.1.1 Topic Detection and Tracking

Topic Detection and Tracking, as initially proposed in a pilot study by Allan et al. [4], has three subtasks: story segmentation, new event detection, and topic tracking. Story segmentation is the task of segmenting a stream of data into distinct stories. New event detection is the task of identifying the first mention of a news story in a stream of text. Topic tracking is the task of finding stories on a given topic in a stream of news, given a small number of stories on the same topic. New event detection is more closely related to novelty detection. The pilot study involved the University of Massachusetts, Carnegie Mellon University, and Dragon Systems. Each group investigated different approaches to each subtask.

The two basic approaches to story segmentation are to model the boundaries between topic segments, and to model the content of topics. Dragon Systems and the University of Massachusetts modeled features of topic boundaries with a Hidden Markov Model, to label the onsets of new topic segments [4]. Dharanipragada et al. [34] use a Maximum Entropy classifier trained on lexical features indicating the start of a new segment of news broadcasts, such as the n-grams "we come back" "time for" and "story and more," and other features, such the length of pauses between speakers, to determine story segments in ASR texts. Eichmann and Srinivasan [39] use agglomerative clustering, starting with each sentence as an individual cluster. Similarity between clusters is measured by the cosine similarity of tf.idf weighted terms in the cluster. The University of Massachusetts also modeled the content of topics with local context analysis [4]. Carnegie Mellon University used a conditional model trained on features of the topics [4].

Topic tracking models the similarity between documents in an incoming stream and previously seen documents. The most successful approach has been the vector space model with cosine similarity of tf.idf-weighted term vectors [146, 31]. Other doc-

ument retrieval approaches, such as Inquery [24], language modeling, and relevance models have also been applied [41, 101, 74].

7.1.2 Plagiarism Detection

Udi Manber [81] first proposed a copy-detection system to find files that had as little as 25% duplicate material. He generates character n-gram *anchors* that cover the document, and computes the checksum fingerprint of the 50 bytes following the anchor. This fingerprint will be the same for any document containing those same 50 bytes and can be stored in an inverted index. Fingerprints are computed for incoming documents, and the index is searched for documents containing the same fingerprints. A document that contains 50% or more of the query document's fingerprints is flagged as a duplicate.

Brin et al. [19] propose a system very similar to that of Manber [81] with exact matching of the hash codes (or checksums) of text segments, rather than arbitrary 50-byte segments. Shivakumar and Garcia-Molina [122] propose a similarity function based on a function related to cosine similarity in a vector space model. Term vectors are a function of the frequencies of a particular fingerprint (computed from a text segment, which could be a word, or a phrase, or a sentence). Papers following from Stanford are variations on this theme [47, 123, 124, 27], and explore broader applications of the technology, and efficiency and scalability concerns.

An issue in fingerprinting techniques is selecting the appropriate text chunks to hash. If the document is segmented from the beginning, a copied document with formatting differences, or insertions and deletions will fail to be detected because the segmentation will differ by an offset of the number of insertions or deletions. Schleimer et al. [120] determine the text chunk by sliding a window of a pre-determined size over the document and selecting the smallest hash value in the window. Sliding windows can be computationally expensive, and as an alternative Bernstein and Zobel [12]

propose an iterative procedure based on rejecting text chunks of progressively larger size that contain unique tokens.

Hoad and Zobel [56] compare fingerprinting (as described above in the work from Stanford, and Manber) to ranking with the vector space model. They propose a variant of the cosine similarity which incorporates the difference in term frequencies between the two documents being compared. They find that ranking outperforms fingerprinting.

7.1.3 Information Provenance

Plagiarism detection looks for exact matches between documents. A clever plagiarist who substitutes unusual words for their synonym terms is less likely to be detected by such systems. Furthermore, many co-derived documents are not plagiarized or copied, but contain a legitimate restatement of the information. For example, several newspaper articles might be written from a single source, and so the basic information remains the same, without being copied. Finding the origins of information in the grey area between "exact match" and "same topic, but new information" at the sub-document level is the task of information provenance.

This grey area is defined on a six-point scale in Metzler et al. [87]. The six points are "unrelated", "on the general topic", "on the specific topic", "same facts", "minor edit" and "identical". They propose a variety of techniques for evaluating similarity at the sentence level, and investigate the performance of each similarity metric at a different category on the six point scale. The similarity metrics investigated are word overlap, tf.idf, relative frequency (as proposed in Bernstein and Zobel [12]), and variants of query likelihood. They examine similarity at both the sentence and document levels. As might be expected, at the sentence level methods designed for document retrieval worked best for sentences that were similar to the "general topic" and "specific topic" categories. Metrics counting word overlap performed well

identifying sentences at the "same facts" and "minor edit" levels. The best performing metric for "exact match" was a version of query likelihood Metzler et al. termed "Model-0" after the Berger and Lafferty term for a translation model that had been constrained to be query likelihood.

7.1.4 Discussion

The value of the work of Metzler et al. was in the presentation of the idea of information provenance, the suggestion that we should be looking at ways to view relevance as a spectrum, and to make a distinction between documents that are on the general topic of the query, and documents that are more specifically related. This represents an innovation in information retrieval because normally all forms of relevance are grouped under one umbrella. Information provenance challenges the assumption that information that is similar is relevant, and information that is relevant is similar.

The work presented in Metzler et al. represents a preliminary study. Although the authors conclude that query likelihood performs best retrieving sentences on the general topic of the query, and that Model-0 performs best for finding sentences that are an exact match, no explanation is given for why this might be true. In fact, the differences in the performance of these models are not significant enough to suggest a practical way to separate sentences on the general topic from sentences in other areas of the spectrum. Furthermore, if most of the sentences on the general topic also share vocabulary, then the same models will perform well on both the "exact match" and "general topic" categories. In Metzler et al. the relevance of the sentences was assessed by two of the authors, and reportedly the inter-assessor agreement was low [86].

No translation probabilities are used in Model-0. The Dirichlet smoothing parameter can be shown to be a function of the length of the sentences, but in Metzler et

al. it is fixed at either 1 or 2500, and is not computed as a function of the length of the sentences. Thus, while the model is reported as a translation model, in reality, it was query likelihood, with untuned Dirichlet smoothing. Query likelihood with tuned Dirichlet smoothing parameters was presented as "query likelihood". This suggests that query likelihood performs well on sentences on the general topic, as well as identical sentences, but that the categories require different smoothing parameters.

As Metzler et al. was a preliminary study, a first step in attacking the problem of information provenance is to solidify some of the ideas presented in Metzler et al. To that end, we have constructed a data set to evaluate different flavors of relevance and similarity, and conducted a relevance assessment of the data, presented in the next section.

7.2 Relevance Assessment Study

Since information provenance is an investigation of the restatement of facts, to begin with we require statements of fact. We constructed 50 queries from the question answering data from Passages task of the TREC QA Track in 2003, with the answer tokens appended to the question. The questions, answers and documents were the same as those described in Chapter 5. Sentences were retrieved from the top 1000 documents returned for these question-answer queries. From the set of retrieved sentences, 51 sentences were chosen as reference sentences that contained statements of fact, and appeared to have a mixture of related and unrelated sentences in the corpus. The final set of 51 reference sentences is presented in the Appendix.

For each of the 51 sentences, a new set of 1000 documents was retrieved, and sentence segmented with MXTerminator [111]. A separate index of sentences was created for each reference sentence. From each index of sentences, the top twenty sentences were retrieved, using query likelihood, Model-S, and Model-1, at five parameter settings for Jelinek-Mercer smoothing for each retrieval method.

The top twenty sentences from each approach for each reference sentence were pooled. They were presented to the assessors in random order, with the reference sentence. Each assessor judged between 10 and 20 reference sentences. There were a total of 3335 retrieved sentences, of which 68 were judged by two assessors, and the rest were judged by three assessors. The assessors were experts in information retrieval, and were instructed to identify identical sentences, and sentences that had minor differences as "Identity or minor edit". Restatements included subsets of the facts. New, but related information included information on the same subtopic that was not included in the reference sentence. All other information, including information on the same general topic, was to be marked "unrelated".

Figure 7.1 shows an example of a reference sentence, and sentences in each category, as marked by the assessors. Sentences that contain exact restatements or minor edits are easily identified by a number of means. More difficult, and potentially more interesting, is identifying identical information expressed in a dissimilar way, and identifying the boundaries between restatements of the same facts, and new, but related information.

Each sentence was categorized according to the majority vote, and sentences that did not have a majority were discarded. Of the 3335 sentences, for 2247, the assessors were unanimous in their judgments and 1004 had a majority vote. The remaining 84 sentences for which there was no majority were discarded. The 84 sentences were distributed across 31 of the 51 topics.

Table 7.1 shows the disagreement among assessors for the sentences that had a majority. The rows represent the majority vote, and the columns represent the dissenting vote. The greatest confusion was between the categories "New but related" and "Unrelated".

1962: Rachel Carson writes "Silent Spring," the first shot in the war against environmental pollution, particularly DDT.
Identity: 1962: Rachel Carson writes "Silent Spring," the first shot in the war against environmental pollution, particularly DDT.
Restatement: 1962 Publication of Rachel Carson's "Silent Spring" stimulates the environmental protection movement. Aides insist that Gore's commitment has remained consistent, and he'll renew that vow with a tour of the homestead of author Rachel Carson whose 1962 classic "Silent Spring" launched what has become the modern environmental movement.
New, but related: Rachel Carson's 1962 book "Silent Spring" said dieldrin causes mania. Rachel Carson, best known for her book "silent Spring" died of breast cancer in 1964.
Unrelated: Rachel, Summer's 150-year-old friend, was named for environmentalist Rachel Carson, author of "Silent Spring." Spring City's Fight Against Industrial Pollution JINAN, December 7 (Xinhua) – The Environmental Protection Bureau of Jinan, the capitol of east China's Shandong Province, took steps today to stop industrial pollution to protect its famous springs.

Figure 7.1. Example of a reference sentence, and sentences related by different flavors of relevance.

		Dissenter			
		Identity minor edit	Restatement	New related	Unrelated
Majority	Identity minor edit	31	12	2	0
	Restatement	18	203	153	50
	New related	1	178	218	234
	Unrelated	0	46	259	1795

Table 7.1. The inter-assessor agreement among sentences judged by three assessors. The row headings indicate the majority category. The column headings indicate the dissenting vote. The "new but related" category and the "unrelated" category were the most disputed.

	Identity (45 sentences)			New but Related (637 sentences)		
	QL	Model-S	Model-1	QL	Model-S	Model-1
Prec @ 5	0.411	0.411	0.411	0.180	0.160	0.168
Prec @ 10	0.237	0.237	0.237	0.230	0.232	0.192⋆
Prec @ 15	0.158	0.158	0.158	0.256	0.240	0.195⋆
Prec @ 20	0.118	0.118	0.118	0.273	0.253⋆	0.193⋆
Ave Prec	0.637	0.640	0.646	0.146	0.136	0.100⋆
	Restatement (431 sentences)			Unrelated (2138)		
	QL	Model-S	Model-1	QL	Model-S	Model-1
Prec @ 5	0.358	0.354	0.200⋆	0.143	0.167	0.306⋆
Prec @ 10	0.356	0.358	0.215⋆	0.259	0.276	0.420⋆
Prec @ 15	0.321	0.317	0.194⋆	0.322	0.354⋆	0.505⋆
Prec @ 20	0.291	0.287	0.190⋆	0.375	0.393	0.555⋆
Ave Prec	0.330	0.322	0.176⋆	0.070	0.073	0.148⋆

Table 7.2. Comparing query likelihood, Model-S, and Model-1 for each relevance category. Results indicated with a star are statistically significant using a t-test at the $p < .05$ level compared to the query-likelihood baseline.

As translation models were proposed by Metzler et al., but not investigated, in the next section we present results for Model-S and Model-1, compared to query likelihood.

7.3 Models of Similarity and Difference

Our data originated with the questions and answers used as the test set in Chapter 5. Thus, the translation table most likely to contain appropriate terms is the translation table learned from the question answering data and we used the same translation table described in Chapter 5. We used Model-S and Model-1, both of which are described in Chapter 4. Recall that Model-S assigns a probability of 1.0 for terms in the query matching terms in the retrieved sentence, and considers no other translations for that term, whereas Model-1 makes no such allowance, and terms have the probability that occurs in the translation table. If no self-translation for a given term appears in the translation table, its translation score is 0.

Table 7.2 shows the results of Model-S and Model-1 compared to query likelihood for each relevance category. There were very few identical sentences in the data, and all models had the same performance at the top of the ranked list, although the average precision differed slightly. Model-S and query likelihood perform comparably on the "restatement" category. Recall that Model-S gives a probability 1.0 to terms in the sentence that are exact matches of terms in the query. Thus if the sentences to be retrieved contain many of the content terms from the query, Model-S will perform equivalently to query likelihood.

Because Model-S behaves equivalently to query likelihood, we investigated whether Model-1 would perform differently. The results in Table 7.2 show that Model-1 performs significantly worse on the "restatement" category than query likelihood. This is to be expected if the data in the "restatement" category shares many terms with the reference sentence, especially if these terms do not appear in the translation table. The same trend is shown in the "new but related" category, but the gap between the performance of query likelihood and the performance of Model-1 is more narrow. It is possible that the translation table derived from the question-answering data is insufficient for the vocabulary of sentences, as was the case with the Novelty Track data. We tried a large high-quality translation table derived from a pair of Arabic-English lexicons, and achieved results nearly identical to the results reported here.

Model-1 performs significantly better than query likelihood and Model-S on "unrelated" sentences. Of the sentences in the "unrelated" category, almost half - 970 of them - are on the general topic of the reference sentence. The other 1167 are not. Table 7.3 shows the results of Model-1 compared to query likelihood on "unrelated" sentences, broken down by topicality. Model-1 performs significantly better than query likelihood on both types of data. However, knowing this does not tell us how to distinguish sentences with different flavors of relevance. The top 10 sentences in an average list ranked by query likelihood will contain three unrelated sentences,

	On Topic (970 sentences)	
	QL	Model-1
Prec @ 5	0.104	0.200⋆
Prec @ 10	0.174	0.250⋆
Prec @ 15	0.203	0.271⋆
Prec @ 20	0.230	0.288⋆
Ave Prec	0.082	0.114⋆
	Off Topic	
	QL	Model-1
Prec @ 5	0.046	0.121⋆
Prec @ 10	0.098	0.190⋆
Prec @ 15	0.135	0.256⋆
Prec @ 20	0.162	0.291⋆
Ave Prec	0.033	0.087⋆

Table 7.3. Comparing query likelihood to Model-1 on unrelated data that is on the general topic of the reference sentence, and unrelated data that is off the topic of the reference sentence. Results indicated with a star are statistically significant at the $p < .05$ level compared to the query-likelihood baseline, using a t-test.

of which two are on topic. The top 10 sentences ranked by Model-1 will contain four or five unrelated sentences, of which two are on topic. If the translation table for Model-1 is especially good, it will retrieve even fewer unrelated sentences.

7.4 Discussion

Information provenance is difficult because the flavors of relevance are exclusive categories, rather than varying degrees of inclusion. Looking at the example in Figure 7.1 we see that all of the sentences, regardless of their category, use the vocabulary of the reference sentence. This example is typical of the sentences in the data. Humans can identify the difference between the categories with some consistency, but even to the human assessor the difference between new, but related information, and unrelated but on-topic information is subtle. Part of the problem is that the notion of "topic" for a sentence is not well defined.

Query likelihood presents sentences higher in the ranked list that contain terms in the reference sentence. It was designed with the assumption that relevant information shares a common vocabulary. Injecting related terms in the form of a translation table, as in Model-S and Model-1, is only useful if the data contains restatements of the same information that uses a new vocabulary, and sentences containing the vocabulary of the reference sentence are also relevant. In the case of information provenance, Model-S fails because sentences containing the same vocabulary as the reference sentence cannot be excluded. Model-1 fails because it is not useful to have a model that retrieves truly unrelated sentences, and Model-1 has no mechanism to retrieve unrelated, but on-topic sentences without retrieving unrelated but off-topic sentences. Model-1 also retrieves some restatements and new, but related sentences, but if the new vocabulary is in the translation table with the vocabulary of the reference sentence, Model-1 has no way to distinguish between related and unrelated information.

In general, retrieving sentences that have a different vocabulary increases the number of off-topic sentences in the ranked list. Furthermore, most off-topic sentences actually share the same vocabulary as the reference sentence and are off-topic because they use different senses of the same words, as in the example in Figure 7.1. With no information other than term statistics, a model has no ability to distinguish between a sentence using one sense of a word and a sentence using another sense of the same word. Models based on identifying the identity of a term cannot distinguish between two separate named entities with the same identifier, as in the word "Spring" in the sentences in the example in Figure 7.1.

The family of language-model approaches to information retrieval are not suited for the task of discriminating between sentences that present new information, possibly with a new vocabulary, and sentences that present the same information, possibly with new vocabulary. We conclude that any model designed to distinguish between

these more subtle categories of relevance must incorporate information other than the distribution of terms in the sentences and their related terms and synonyms.

CHAPTER 8

CONCLUSION AND FUTURE DIRECTIONS

As stated in the introduction to this thesis we offer the following contributions:

- A task-independent study of sentence retrieval
  - A representation of the relevance of a sentence as a spectrum
  - A model of similarity (Model-S) appropriate to a wide range of sentence retrival tasks
  - A novel application of conditional models to the task of sentence retrieval
  - A method of smoothing that significantly improves sentence retrieval
- A study of information provenance
  - An explanation of why the language model family of retrieval methods are not appropriate for information provenance.
  - A corpus for the further study of information provenance

Query likelihood was designed with the assumption that relevant documents share the vocabulary of the query. It was designed this way because often the query is the only evidence we have about a user's information need. The model is elegant and effective for the task of document retrieval and represents a step forward in information retrieval because it provides a unifying mathematical framework upon which to build the technology. When the unit of retrieval is reduced to passages or sentences, retrieval performance also diminishes because the frequency of a term in a sentence is not as meaningful a metric by which to evaluate a sentence.

Sentences depend on context for meaning. Consider the example in Figure 6.2,page 96. The second and third sentences are relevant when considered as a group with the first sentence, which is no doubt why they were marked relevant by the assessors. Smoothing from the local context allows us to retrieve a sentence, as opposed to a passage or document, that requires context to give it meaning as in the example in Chapter 6. In other cases, the local context of the sentence disambiguates its meaning, as in the task of question answering, as shown in Table 5.3, page 65, where the improvements in the strict metric demonstrate that smoothing from the document improves the retrieval of sentences from contextually appropriate documents. The local context does not aid the retrieval when the sentences are being retrieved from a document set that is small and mostly relevant, as in the relevance task for the Novelty Track, shown in Table 6.7, (page 100). In this case, the collection provides sufficient context for the question.

Sentences carry only slightly more information than the typical query. In question answering, we look for sentences that contain many of the terms in the question, but require the answer also to be present in the sentence. Model-S improves results significantly under these conditions because it prefers sentences with exact matches, but gives additional weight to related terms. Model-S also improves results for queries of two or three terms, such as topic titles from the Novelty Track by adding information to the query in the form of "translation" terms. When we are looking for sentences on the general topic of the query, and the query is a full description, such as the topic descriptions for the Novelty Track, Model-S offers little benefit. As shown in Table 6.6, 98, when the relevant sentence is not required to contain information that is not in the query, and there are few sentences in the data that contain related words, related words are not helpful. Model-S fails for the task of information provenance because it can not eliminate sentences with exact matching terms. The cousin of Model-S, Model-1 retrieves fewer sentences containing exact matching terms, but because it

can not discriminate between terms related to a given topic, and terms related to a different topic, it is not useful.

For the task of information provenance, we require a different sort of model. The approaches to sentence and document retrieval that rely on term frequencies do not allow for the separation of different flavors of relevance. Such models allow us to compute the probability that a sentence was generated from the model, but it does not allow us to discriminate between different types of information that come from the same distribution of terms.

To compare the distribution of terms in a document with the distribution of terms in a query is at once the beauty and limitation of modern information retrieval methods. Models of topicality are simple and flexible enough to capture a wide range of documents similar to a query. This is especially elegant considering how impoverished the query is in content compared to a typical length document. But these retrieval models can only do so much: there is no magic behind the curtain. They can capture a narrow range of documents that use the same vocabulary as the query, or they cast a wide net and capture documents that are topically related to the query. When it comes to capturing new information, information that can never be in the query (in the case of question answering) or new, but related information (in the case of novelty detection and information provenance) they fail. They fail when the document itself is as impoverished as the query, and they fail to distinguish the difference between information that shares the same vocabulary and the same topic, and information that shares the same vocabulary but presents a new topic.

The success of conditional models for the task of question answering bespeaks the need for features of sentences other than the distribution of terms. Where the term distribution is sparse, the conditional model allows for the inclusion of other measures of similarity to distinguish relevant information from nonrelevant information.

Solving the problem of Information Provenance - separating varying levels of similarity to capture different notions of relevance - would allow us to separate the wheat from the chaff and return exactly the type of information sought, in all information retrieval tasks. We would like to adjust the granularity of the retrieval response to meet the information need of the user. To do this with subtlety and precision, we need a different model of similarity that can incorporate all of the available information about textual expression. Working with sentences allows us to combine information or distill it as needed to allow people to communicate their needs in their own language, and to respond in kind. When we can accomplish this, we will have succeeded in the original goal of information retrieval of obviating the reference librarian.

APPENDIX

REFERENCE SENTENCES FOR THE TASK OF INFORMATION PROVENANCE

1. Catherine the Great began the Hermitage collection in the 18th century, and the museum opened to the public in 1852.

2. The CN tower in Toronto, Canada has laid claim to the world tallest structure since its was built in 1976 and stands at 1,815 feet (553.3 meters).

3. Ponce de Leon was looking for the Fountain of Youth when he reached what is now St. Augustine in 1513.

4. Hank Aaron holds the major-league record with 755 home runs in a 23-season career.

5. The Milky Way, the galaxy that holds the sun and Earth, contains about 200 billion stars.

6. There are more than 12 million transfusions a year in this country.

7. Marilyn Monroe's form-fitting dress – a stunner she wore in 1962 to serenade President Kennedy – sold tonight for a record $1,267,500.

8. White-shelled eggs are produced by hens with white feathers and ear lobes; brown-shelled, by hens with red feathers and red ear lobes.

9. Geneticists have calculated, for example, that the early human population of sub-Saharan Africa was once as small as 6,900 people, and that the group that left Africa to colonize the rest of the world may have numbered as few as 500.

10. Monet, who died in 1926 at the age of 86, is perhaps the best known of the Impressionists.

11. Giant squids have 10 large tentacles lined with sucker pads and a reputation for ruthlessness.

12. The Battle of Verdun, a fruitless 300-day attempt by Germany to break through the Western Front in 1916, wiped nine villages off the map and killed 162,000 French and 143,000 German soldiers.

13. The synod recommended sainthood for Nicholas II, his wife Alexandra, their son and four daughters – all of whom were shot to death by Communist guards July 17, 1918.

14. Since opening the original Disneyland in Anaheim, Calif., in 1955, Disney has expanded to Tokyo, France, and Florida, where it as four parks in the Orlando area.

15. 1962 Rachel Carson writes "Silent Spring," the first shot in the war against environmental pollution, particularly DDT.

16. Egypt used to suffer periodic flooding from the River Nile until the completion of the Aswan High Dam.

17. Three years after the death of Jerry Garcia, his fellow members of the Grateful Dead are back together as the core of a new band, the Other Ones, that sets out to resurrect and extend the Dead's legacy.

18. At its "perigee," the closest approach to Earth, the moon is 221,456 miles away.

19. At least 20 states in recent years have designated English as the official state language, but many of the laws appear to be symbolic and do not restrict government use of other languages.

20. Previous French presidents have tried – and failed – to change the seven-year term, which has been in place since 1873.

21. Napoleon was 52 when he died May 5, 1821 on the remote volcanic island of St. Helena, where he was exiled after his final defeat at Waterloo.

22. The Tour de France has been around since 1903, stopping only for the two world wars.

23. Like single-malt whisky and specialty bourbon, premium tequila, made from 100 percent blue agave, is handmade, with subtle variations, a rich history and a complex taste.

24. Indonesia's 17,000 islands are dotted by some 500 volcanoes, of which 127 are still active and make up the Pacific rim of fire.

25. SPEED-OF-LIGHT (Undated) - For generations, physicists believed there is nothing faster than light moving through a vacuum - a speed of 186,000 miles per second.

26. Eminem is a Grammy-winning white rapper from Detroit whose real name is Marshall Mathers and whose wife is Kim.

27. Suzhou is China's largest silk producer, with its annual output accounting for one sixth of the country's total.

28. Barbie's official name is Barbie Millicent Roberts from Willows, Wis., though her creator Ruth Handler lived in Southern California.

29. When he was Assembly speaker in California, Willie Brown, a Democrat who is now the mayor of San Francisco, had an active law practice, and was frequently accused of conflicts of interest.

30. Track and field legend Carl Lewis of the United States retired at the age of 35 after winning nine Olympic gold medals and eight world championship titles in his career.

31. Under relevant U.N. Security Council resolutions, Iraq's assets in foreign countries have been frozen since it invaded Kuwait in 1990.

32. The Titanic sank in 12,000 feet of water after hitting a North Atlantic iceberg in April 1912.

33. Hyundai Motors is a subsidiary of South Korea's largest conglomerate, Hyundai, which is run by the family of Chung Mong- joon, head of the country's football association.

34. Hinckley, 42, shot at former president Ronald Reagan outside a Washington hotel on March 30, 1981, wounding the president and three others including Reagan's press secretary James Brady who was disabled.

35. Earlier Tuesday, a ten-member bench of the Supreme Court held in an unanimous decision the appointment of Justice Sajjad Ali Shah as Chief Justice of Pakistan as invalid.

36. It was Jan. 22, 1973, that the Supreme Court made abortion legal.

37. Located in Hubei Province, the Three Gorges Dam is the world's largest hydroelectric project and is scheduled for completion in 2009.

38. Researchers from San Francisco State University said Thursday that they have for the first time discovered the evidence of another solar system, suggesting that the Milky Way, which contains about 200 billion stars, may has many planetary systems.

39. Under the Panama Canal treaties, American soliders should leave by December 31, 1999, when the control of the Panama Canal passes to the Panamanian government.

40. The United States, the richest country in the world, is the biggest debtor to the United Nations, owing the world body some 1.7 billion U.S. dollars.

41. The MGM Grand Hotel & Theme Park, with 5,005 rooms, has the most rooms of any Las Vegas hotel.

42. The Suez canal, the first canal directly linking the Mediterranean Sea to the Red Sea, was opened for international navigation on November 17, 1879 and Egypt nationalized it on 26 July, 1956.

43. The insect that drills holes in wood is a carpenter bee, and even she (females drill holes to store their eggs in) would drill holes in siding or outside trim.

44. In England, for example, Nov. 5 is Guy Fawkes Day; the observance commemorates the day in 1605 when a man named Guy Fawkes was accused of attempting to bloe Houses of Parliament.

45. Diana, 36, died in a hospital early today after a car crash, which happened last night at a Paris road tunnel under the Alma bridge on the Seine River bank.

46. The stairs and plaza in front of the Philadelphia Museum of Art (the "Rocky steps" locale) are the setting Wednesday morning for "Es Un Nuevo Dia," (It's A New Day), a Hispanic-themed event welcoming Gov. George W. Bush to Philadelphia.

47. Mission controllers gave up hope today of finding the Mars Climate Orbiter and were trying to determine what caused the $125 million NASA spacecraft to disappear.

48. In October, five prison guards at Corcoran were indicted on state charges of conspiracy, accused of having an inmate rape another inmate in 1993.

49. Apart from Sierra Leone, which is ranked last, the other least developed nations, from the bottom up, are Niger, Burkina Faso, Ethiopia, Burundi, Guinea-Bissau, Mozambique, Chad, the Central African Republic and Mali.

50. What is certain is that 1 million Austrians fought beside Hitler, many Austrians believe they fought nobly or at least dutifully, Eichmann ran the Third Reich's racial pogroms from Vienna and, after the war, the Allies for their own reasons led the Austrians to think they were really victims of Hitler, and the first ones at that.

51. Defeated Hutu soldiers and militiamen, many of them responsible for the 1994 slaughter of more than 500,000 Tutsis, hid among the returning refugees and are now bent on destabilizing the country.

BIBLIOGRAPHY

[1] Abdul-Jaleel, Nasreen, Allan, James, Croft, W. Bruce, Diaz, Fernando, Larkey, Leah, Li, Xiaoyan, Metzler, Donald, Smucker, Mark D., Strohman, Trevor, Turtle, Howard, and Wade, Courtney. UMass at TREC 2004: Novelty and HARD. In *Proceedings of the Thirteenth Text Retrieval Conference (TREC)* (2004).

[2] Agichtein, Eugene, and Gravano, Luis. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the 5th ACM International Conference on Digital Libraries* (2000).

[3] Al-Onaizan, Yaser, Curin, Jan, Jahr, Michael, Knight, Kevin, Lafferty, John, Melamed, Dan, Och, Franz-Josef, Purdy, David, Smith, Noah A., and Yarowsky, David. Statistical machine translation, final report, JHU workshop, 1999.

[4] Allan, James, Carbonell, Jaime, Doddington, George, Yamron, Jonathan, and Yang, Yiming. Topic detection and tracking pilot study final report. In *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop* (1998), pp. 194–218.

[5] Allan, James, Wade, Courtney, and Bolivar, Alvaro. Retrieval and novelty detection at the sentence level. In *Proceedings of the 26th International ACM SIGIR Conference on Research and Development in Information Retrieval* (2003).

[6] Baeza-Yates, Ricardo, and Ribeiro-Neto, Berthier. *Modern Information Retrieval*. Addison Wesley, 1999.

[7] Bahl, Lalit, Jelinek, Fred, and Mercer, Robert. A maximum likelihood approach to continuous speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 5*, 2 (1983), 179–190.

[8] Bangalore, Srinivas, Bordel, German, and Riccardi, Giuseppe. Computing consensus translation from multiple machine translation systems. In *Proceedings of the Workshop on Automatic Speech Recognition and Understanding* (2001).

[9] Bangalore, Srinivas, Murdock, Vanessa, and Riccardi, Giuseppe. Bootstrapping bilingual data using consensus translation for a multilingual instant messaging system. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING)* (2002).

[10] Berger, Adam, Caruana, Rich, Cohn, David, Freitag, Dayne, and Mittal, Vibhu. Bridging the lexical chasm: Statistical approaches to answer-finding. In *Proceedings of the 23rd Annual Conference on Research and Development in Information Retrieval (ACM SIGIR)* (2000), pp. 192–199.

[11] Berger, Adam, and Lafferty, John. Information retrieval as statistical translation. In *Proceedings of the 22nd Annual Conference on Research and Development in Information Retrieval (ACM SIGIR)* (1999).

[12] Bernstein, Yaniv, and Zobel, Justin. A scalable system for identifying coderivative documents. In *Proceedings of the 11th Symposium on String Processing and Information Retrieval* (2004).

[13] Bikel, Daniel, Schwartz, Richard, and Weischedel, Ralph. An algorithm that learns what's in a name. *Machine Learning Special Issue on Natural Language Learning* (1999).

[14] Blott, Stephen, Boydell, Oisin, Camous, Fabrice, Ferguson, Paul, Gaughan, Georgina, Gurrin, Cathal, Jones, Gareth J. F., Murphy, Noel, O'Connor, Noel, Smeaton, Alan F., Smyth, Barry, and Wilkins, Peter. Experiments in terabyte searching, genomic retrieval and novelty detection for TREC-2004. In *Proceedings of the Thirteenth Text Retrieval Conference (TREC)* (2004).

[15] Brants, Thorsten, Chen, Francine, and Farahat, Ayman. A system for new event detection. In *Proceedings of the 26th International ACM SIGIR Conference on Research and Development in Information Retrieval* (2003).

[16] Brill, Eric, Dumais, Susan, and Banko, Michele. An analysis of the AskMSR question-answering system. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (2002).

[17] Brill, Eric, Lin, Jimmy, Banko, Michele, Dumais, Susan, and Ng, Andrew. Data intensive question answering. In *Proceedings of the Tenth Text REtrieval Conference (TREC)* (2001).

[18] Brin, Sergey. Extracting patterns and relations from the world wide web. In *WebDB Workshop at the 6th International Conference on Extending Database Technology (EDTB)* (1998).

[19] Brin, Sergey, Davis, James, and Garcia-Molina, Hector. Copy detection mechanisms for digital documents. In *Proceedings of the International Conference on Management of Data (SIGMOD)* (1995), pp. 398–409.

[20] Brown, Peter F., Cocke, John, Della Pietra, Stephen A., Della Pietra, Vincent J., Jelineck, Fredrick, Lafferty, John D., Mercer, Robert L., and Roossin, Paul S. A statistical approach to machine translation. *Computational Linguistics 16*, 2 (1990), 79–85.

[21] Brown, Peter F., Della Pietra, Stephen A., Della Pietra, Vincent J., and Mercer, Robert L. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics 19*, 2 (1993), 263 – 311.

[22] Buchholz, Sabine. Using grammatical relations, answer frequencies and the world wide web for TREC question answering. In *Proceedings of the Tenth Text Retrieval Conference (TREC)* (2001).

[23] Buckley, Chris, Salton, Gerard, Allan, James, and Singhal, Amit. Automatic query expansion using SMART : TREC 3. In *NIST Special Publication 500-225: The Third Text Retrieval conference (TREC 3)* (1995), D. K. Harmon, Ed., pp. 69–80.

[24] Callan, Jamie, Croft, W. Bruce, and Harding, Steven M. The INQUERY retrieval system. In *Proceedings of the 3rd International Conference on Database and Expert Systems Applications* (1992), pp. 78–83.

[25] Cardie, Claire, Ng, Vincent, Pierce, David, and Buckley, Chris. Examining the role of statistical and linguistic knowledge sources in a general-knowledge question-answering system. In *Proceedings of the Sixth Applied Natural Language Processing Conference (ANLP)* (2000).

[26] Chen, Stanley F., and Rosenfeld, Ronald. A gaussian prior for smoothing maximum entropy. Tech. Rep. CMU-CS-99-108, Carnegie Mellon University, School of Computer Science, 1999.

[27] Cho, Junghoo, Shivakumar, Narayanan, and Garcia-Molina, Hector. Finding replicated web collections. In *Proceedings of the International Conference on Management of Data (SIGMOD)* (2000).

[28] Clarke, Charles L. A., Cormack, Gordon V., and Lynam, Thomas R. Exploiting redundancy in question answering. In *Proceedings of the 24th Annual Conference on Research and Development in Information Retrieval (ACM SIGIR)* (2001).

[29] Clarke, Charles L. A., Cormack, Gordon V., Lynam, Thomas R., Li, C. M., and McLearn, Greg L. Web reinforced question answering. In *Proceedings of the Tenth Text Retrieval Conference (TREC)* (2001).

[30] Collins-Thompson, Kevyn, Ogilvie, Paul, Zhang, Yi, and Callan, Jamie. Information filtering, novelty detection and named-page finding. In *Proceedings of the Eleventh Text Retrieval Conference (TREC)* (2002).

[31] Connell, Margaret, Feng, Ao, Kumaran, Giridhar, Raghavan, Hema, Shah, Chirag, and Allan, James. UMass at TDT 2004. In *Working Notes of the Topic Detection and Tracking (TDT) Evaluation* (2004).

[32] Cui, Hang, Li, Keya, Sun, Renxu, Chua, Tat-Seng, and Kan, Min-Yen. National University of Singapore at the TREC-13 question answering main task. In *Proceedings of the Thirteenth Text Retrieval Conference (TREC)* (2004).

[33] Czuba, Krzysztof, Prager, John M., and Chu-Carroll, Jennifer. A machine learning approach to introspection in a question answering system. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)* (2002), pp. 265 – 272.

[34] Dharanipragada, Satya, Franz, Martin, McCarley, J. S., Ward, Todd, and Zhu, W.-J. Segmentation and detection at IBM. In *Topic Detection and Tracking*, James Allan, Ed. Kluwer Academic Publishers, 2002, ch. 7.

[35] Dkaki, Taoufiq, and Mothe, Josiane. TREC novelty track at IRIT-SIG. In *Proceedings of the Thirteenth Text Retrieval Conference (TREC)* (2004).

[36] Dumais, Susan, Banko, Michele, Brill, Eric, Lin, Jimmy, and Ng, Andrew. Web question answering: Is more always better? In *Proceedings of the 25th Annual Conference on Research and Development in Information Retrieval (ACM SIGIR)* (2002).

[37] Durbin, Richard, Eddy, Sean, Krogh, Anders, and Mitchison, Graeme. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1998.

[38] Echihabi, Abdessamad, and Marcu, Daniel. A noisy-channel approach to question answering. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL)* (2003).

[39] Eichmann, David, and Srinivasan, Padmini. A cluster-based approach to broadcast news. In *Topic Detection and Tracking*, James Allan, Ed. Kluwer Academic Publishers, 2002, ch. 8.

[40] Eichmann, David, Zhang, Yi, Bradshaw, Shannon, Qui, Xin Ying, Zhou, Li, Srinivasan, Padmini, Sehgal, Adtiya Kumar, and Wang, Hudon. Novelty, question answeirng and genomics: The University of Iowa response. In *Proceedings of the Thirteenth Text Retrieval Conference (TREC)* (2004).

[41] Elsayed, Tamer, Oard, Douglas W., and Doermann, David. TDT-2004: Adaptive topic tracking at Maryland. In *Working Notes of the Topic Detection and Tracking (TDT) Evaluation* (2004).

[42] Erkan, Güneş. The University of Michigan in novelty 2004. In *Proceedings of the Thirteenth Text Retrieval Conference (TREC)* (2004).

[43] Evans, David A., and Zhai, ChengXiang. Noun-phrase analysis in unrestricted text for information retrieval. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics (ACL)* (1996).

[44] Fan, Wei, Miller, Matthew, Stolfo, Salvatore J., Lee, Wenke, and Chan, Philip K. Using artificial anomalies to detect unknown and known network intrusions. In *Proceedings of the 1st IEEE International Conference on Data Mining (ICDM)* (2001).

[45] Ferret, Oliver, Grau, Brigitte, Hurault-Plantet, Martine, Illouz, Gabriel, Monceaux, Laura, Robba, Isabelle, and Vilnat, Anne. Finding an answer based on the recognition of the question focus. In *Proceedings of the Tenth Text Retrieval Conference (TREC)* (2001).

[46] Gaizauskas, Robert, Greenwood, Mark A., Hepple, Mark, Roberts, Ian, and Saggion, Horacio. The University of Sheffield's TREC 2004 Q&A experiments. In *Proceedings of the Thirteenth Text Retrieval Conference (TREC)* (2004).

[47] Garcia-Molina, Hector, Gravano, Luis, and Shivakumar, Narayanan. dSCAM: Finding document copies across multiple databases. In *Proceedings of the 4th International Conference on Parallel and Distributed Information Systems (PDIS)* (1996).

[48] Gebhardt, Friedrich. A simple probabilistic model for the relevance assessments of documents. *Information Processing and Management 11*, 1-2 (1975), 59–65.

[49] Gonzalez, Fabio, and Dasgupta, Dipankar. An immunogenetic approach to intrusion detection. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)* (2002).

[50] Goodman, Joshua. Exponential priors for maximum entropy models. In *Proceedings of the North American Association of Computational Linguistics (NAACL)* (2004).

[51] Greenwood, Mark A., and Saggion, Horacio. A pattern based approach to answering factoid, list and definition questions. In *Proceedings of RIAO* (2004).

[52] Han, Kyoung-Soo, Chung, Hoojun, Kim, Sang-Bum, Song, Young-In, Lee, Joo-Young, and Rim, Hae-Chang. Korea University question answering system at TREC 2004. In *Proceedings of the Thirteenth Text Retrieval Conference (TREC)* (2004).

[53] Harabagiu, Sanda, and Moldovan, Dan. An intelligent system for question answering. In *Proceedings of the 5th International Conference on Intelligent Systems* (Reno, NV, 1996), pp. 71–75.

[54] Harman, Donna. Overview of the TREC 2002 novelty track. In *Proceedings of the Eleventh Text Retrieval Conference (TREC)* (2002).

[55] Hildebrandt, Wesley, Katz, Boris, and Lin, Jimmy. Answering definition questions using multiple knowledge sources. In *Proceedings of HLT-NAACL* (2004).

[56] Hoad, Timothy C., and Zobel, Justin. Methods for identifying versioned and plagiarised documents. *Journal of the American Society of Information Science and Technology 54*, 3 (2003), 203–215.

[57] Hovy, Eduard, Hermjakob, Ulf, and Lin, Chin-Yew. The use of external knowledge in factoid question answering. In *Proceedings of the Tenth Text Retrieval Conference (TREC)* (2001).

[58] Hovy, Eduard, Hermjakob, Ulf, and Ravichandran, Deepak. A question/answer typology with surface text patterns. In *Proceedings of the Conference on Human Language Technologies (HLT)* (2002).

[59] http://www.wramc.amedd.army.mil/wramc/redcross/historyARC.htm.

[60] Ittycheriah, Abraham, Franz, Martin, and Roukos, Salim. IBM's statistical question answering system - TREC-10. In *Proceedings of the Tenth Text Retrieval Conference (TREC)* (2001).

[61] Ittycheriah, Abraham, Franz, Martin, Zhu, Wei-Jing, and Ratnaparkhi, Adwait. IBM's statistical question answering system. In *Proceedings of the Ninth Text Retrieval Conference (TREC)* (2000), pp. 229 – 235.

[62] Jeon, Jiwoon, Lavrenko, Victor, and Manmatha, R. Automatic image annotation and retrieval using cross-media relevance models. In *Proceedings of the 26th Annual Conference on Research and Development in Information Retrieval (ACM SIGIR)* (2003).

[63] Kaisser, Michael, and Becker, Tilman. Question answering by searching large corpora with linguistic methods. In *Proceedings of the Thirteenth Text Retrieval Conference (TREC)* (2004).

[64] Katz, Boris, Bilotti, Matthew, Felshin, Sue, Fernandes, Aaron, Hildebrandt, Wesley, Katzir, Roni, Lin, Jimmy, Loreto, Daniel, Marton, Gregory, Mora, Federico, and Usuner, Ozlem. Answering multiple questions on a topic from heterogeneous resources. In *Proceedings of the Thirteenth Text Retrieval Conference (TREC)* (2004).

[65] Kim, Soo-Min, Ravichandran, Deepak, and Hovy, Eduard. ISI novelty track system for TREC 2004. In *Proceedings of the Thirteenth Text Retrieval Conference (TREC)* (2004).

[66] Krovetz, Robert. Viewing morphology as an inference process. In *Proceedings of the 16th International ACM SIGIR Conference on Research and Development in Information Retrieval* (1993).

[67] Kumaran, Giridhar, and Allan, James. Using names and topics for new event detection. In *Proceedings of the Conference on Human Language Technologies and Empirical Methods in Natural Language Processing (HLT/EMNLP)* (2005).

[68] Kupiec, Julian. Murax: A robust linguistic approach for question answering using an on-line encyclopedia. In *Proceedings of the 16th Annual Conference on Research and Development in Information Retrieval (ACM SIGIR)* (1993), pp. 181–190.

[69] Lafferty, John, McCallum, Andrew, and Pereira, Fernando. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conference on Machine Learning* (2001).

[70] Larkey, Leah. Personal communication, January 2005.

[71] Larkey, Leah, Allan, James, Connell, Margie, Bolivar, Alvaro, and Wade, Courtney. UMass at TREC 2002: Cross language and novelty tracks. In *Proceedings of the Eleventh Text Retrieval Conference (TREC)* (2002), p. 721.

[72] Larkey, Leah S., Ballesteros, Lisa, and Connell, Margaret E. Improving stemming for arabic information retrieval: Light stemming and co-occurrence analysis. In *Proceedings of the 25th Annual Conference on Research and Development in Information Retrieval (SIGIR)* (2002).

[73] Laurikkala, Jorma, Juhola, Martti, and Kentala, Erna. Informal identification of outliers in medical data. In *Proceedings of the 5th International Workshop on Intelligent Data Analysis in Medicine and Pharmacology (IDAMAP) at the 14th European Conference on Artificial Intelligence* (2000).

[74] Lavrenko, Victor, Allan, James, DeGuzman, Edward, LaFlamme, Daniel, Pollard, Veera, and Thomas, Stephen. Relevance models for topic detection and tracking. In *Proceedings of the Conference on Human Language Technologies(HLT)* (2002).

[75] Lavrenko, Victor, Choquette, Martin, and Croft, W. Bruce. Cross-lingual relevance models. In *Proceedings of the 25th Annual Conference on Research and Development in Information Retrieval (ACM SIGIR)* (2002).

[76] Lavrenko, Victor, and Croft, W. Bruce. Relevance-based language models. In *Proceedings of the 24th International ACM SIGIR Conference on Research and Development in Information Retrieval* (2001).

[77] Lavrenko, Victor, Manmatha, R., and Jeon, Jiwoon. A model for learning the semantics of pictures. In *Proceedings of the 17th Annual Conference on Neural Information Processing Systems* (2003).

[78] Lee, Gary Geunbae, Seo, Junyun, Lee, Seungwoo, Jung, Hanmin, Cho, Bong-Hyun, Lee, Changki, Kwak, Byung-Kwan, Cha, Jeongwon, Kim, Dongseok, An, JooHui, Kim, Harksoo, and Kim, Kyungsun. SiteQ: Engineering high performance QA system using lexico-semantic pattern matching and shallow nlp. In *Proceedings of the Tenth Text Retrieval Conference (TREC)* (2001).

[79] Litkowski, Kenneth. Evolving XML and dictionary strategies for question answering and novelty. In *Proceedings of the Thirteenth Text Retrieval Conference (TREC)* (2004).

[80] Makkonen, Juha, Ahonen-Myka, Helena, and Salmenkivi, Marko. Applying semantic classes in event detection and tracking. *Information Retrieval 7* (2004), 347–368.

[81] Manber, Udi. Finding similar files in a large file system. In *Proceedings of the Winter USENIX Technical Conference* (1994).

[82] Manikopoulos, Constantine, and Papavassiliou, Symeon. Network intrusion and fault detection: a statistical anomaly approach. *IEEE Communications Magazine 40*, 10 (2002), 76–82.

[83] Maron, Melvin Earl, and Kuhns, J. L. On relevance, probabilistic indexing and information retrieval. *Journal of the ACM 7*, 3 (1960), 216–244. Reprinted in [128].

[84] McCallum, Andrew, and Feng, Fangfang. Chinese word segmentation with conditional random fields and integrated domain knowledge. Tech. rep., University of Massachusetts, 2003. CIIR Tech Report.

[85] McCallum, Andrew Kachites. MALLET: A machine learning for language toolkit, 2002. http://www.cs.umass.edu/ mccallum/mallet.

[86] Metzler, Donald. Personal communication, June 2005.

[87] Metzler, Donald, Bernstein, Yaniv, Croft, W. Bruce, Moffat, Alistair, and Zobel, Justin. Similarity measures for tracking information flow. In *Proceedings of the Conference on Information and Knowledge Management* (2005). to appear.

[88] Miliaraki, Spyridoula, and Androutsopoulos, Ion. Learning to identify single-snippet answers to definition questions. In *Proceedings of the Conference on Computational Linguistics (COLING)* (2004).

[89] Miller, George A. WordNet: A lexical database. *Communications of the ACM 38*, 11 (1995), 39–41.

[90] Miller, S., Crystal, M., Fox, H., Ramshaw, L., Schwartz, R., Stone, R., Weischedel, R., and the Annotation Group. Algorithms that learn to extract information–BBN: Description of the SIFT system as used for MUC-7. In *Proceedings of the Seventh Message Understanding Conference (MUC-7)* (1998).

[91] Mitra, Mandar, Singhal, Amit, and Buckley, Chris. Improving automatic query expansion. In *Proceedings of the 21st Annual Conference on Research and Development in Information Retrieval (ACM SIGIR)* (1998), pp. 206–214.

[92] Monz, Christof, Kamps, Jaap, and de Rijke, Maarten. The University of Amsterdam at TREC 2002. In *Proceedings of the Eleventh Text Retrieval Conference (TREC)* (2002).

[93] Murdock, Vanessa, and Croft, W. Bruce. Task orientation in question answering. In *Proceedings of the 25th Annual Conference on Research and Development in Information Retrieval (ACM SIGIR)* (2002).

[94] Murdock, Vanessa, and Croft, W. Bruce. Simple translation models for sentence retrieval in factoid question answering. In *Proceedings of the Information Retrieval for Question Answering Workshop at SIGIR 2004* (2004).

[95] Murdock, Vanessa, and Croft, W. Bruce. A translation model for sentence retrieval. In *Proceedings of the Conference on Human Language Technologies and Empirical Methods in Natural Language Processing (HLT/EMNLP)* (2005).

[96] Nigam, Kamal, Lafferty, John, and McCallum, Andrew. Using maximum entropy for text classification. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI) Workshop on Information Filtering* (1999).

[97] Och, Franz Josef, and Ney, Hermann. Improved statistical alignment models. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL)* (2000), pp. 440–447.

[98] Och, Franz Josef, and Ney, Hermann. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics* (2002).

[99] Ohgaya, Ryosuke, Shimmura, Akiyoshi, and Takagi, Tomohiro. Meiji University web and novelty track experiments at TREC 2003. In *Proceedings of the Twelth Text Retrieval Conference (TREC)* (2003).

[100] Papineni, Kishore, Roukos, Salim, and Ward, R. Todd. Maximum likelihood and discriminative training of direct translation models. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (1998), pp. 189–192.

[101] Papka, Ron, Allan, James, and Lavrenko, Victor. UMass approaches to detection and tracking at TDT2. In *Working Notes of the Topic Detection and Tracking (TDT) Evaluation* (2002).

[102] Pinto, David, McCallum, Andrew, Wei, Xing, and Croft, W. Bruce. Table extraction using conditional random fields. In *Proceedings of the ACM SIGIR* (2003).

[103] Ponte, Jay, and Croft, W. Bruce. A language modeling approach to information retrieval. In *Proceedings of the 21st Annual Conference on Research and Development in Information Retrieval (ACM SIGIR)* (1998).

[104] Porter, Martin F. An algorithm for suffix stripping. *Program 14*, 3 (1980), 130–137. Reprinted in [128].

[105] Prager, John M., Chu-Carroll, Jennifer, and Czuba, Krzysztof. Use of WordNet hypernyms for answering what-is questions. In *Proceedings of the Tenth Text Retrieval Conference (TREC)* (2001).

[106] Rabiner, Lawrence R. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE 77*, 2 (1989), 257–286.

[107] Radev, Dragomir R., Qi, Hong, Zheng, Zhiping, Blair-Goldensohn, Sasha, Zhang, Zhu, Fan, Weiguo, and Prager, John. Mining the web for answers to natural language questions. In *Proceedings of the 10th International Conference on Information Knowledge Management (CIKM)* (Atlanta, GA, 2001).

[108] Ravichandran, Deepak, and Hovy, Eduard. Learning surface text patterns for a question answering system. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)* (Philadelphia, PA, 2002).

[109] Ravichandran, Deepak, Hovy, Eduard, and Och, Franz Josef. Statistical QA - classifier vs. re-ranker: What's the difference? In *Proceedings of the ACL Workshop on Multilingual Summarization and Question Answering - Machine Learning and Beyond* (2003).

[110] Ravichandran, Deepak, Ittycheriah, Abraham, and Roukos, Salim. Automatic derivation of surface text patterns for a maximum entropy based question answering system. In *Proceedings of HLT-NAACL* (2003).

[111] Reynar, Jeffrey C., and Ratnaparkhi, Adwait. A maximum entropy approach to identifying sentence boundaries. In *Proceedings of the 5th Conference on Applied Natural Language Processing (ANLP)* (1997). http://www.cis.upenn.edu/ãdwait/statnlp.html.

[112] Ribeiro-Neto, Berthier A., and Muntz, Richard. A belief network model for IR. In *Proceedings of the 19th Annual Conference on Research and Development in Information Retrieval (SIGIR)* (1996).

[113] Robertson, Steven E. The probabilistic character of relevance. *Information Processing and Management 13*, 4 (1977), 247–251.

[114] Robertson, Steven E. The probability ranking principle in IR. *Journal of Documentation 33* (1977), 294–304. Reprinted in [128].

[115] Robertson, Steven E., and Jones, Karen Sparck. Relevance weighting of search terms. *Journal of the American Society for Information Sciences 27*, 3 (1976), 129–146.

[116] Rocchio, Joseph J. Relevance feedback in information retrieval. In *The SMART Retrieval System - Experiments in Automatic Document Processing*, Gerard Salton, Ed. Prentice Hall, Englewood Cliffs, NJ, 1971.

[117] Saggion, Horacio. Identifying definitions in text collections for question answering. In *Proceedings of the Conference on Language Resources and Evaluation (LREC)* (2004).

[118] Salton, Gerard, and Buckley, Christopher. Term-weighting approaches in automatic text retrieval. *Information Processing and Management 24*, 5 (1988), 513 – 523.

[119] Saracevic, Tefko. Relevance: A review of and a framework for the thinking on the notion in information science. *Advances in Librarianship 6* (1976). Reprinted in [128].

[120] Schleimer, Saul, Wilkerson, Daniel S., and Aiken, Alex. Winnowing: Local algorithms for document fingerprinting. In *Proceedings of the International Conference on Management of Data (SIGMOD)* (2003).

[121] Sha, Fei, and Pereira, Fernando. Shallow parsing with conditional random fields. In *Proceedings of HLT-NAACL* (2003).

[122] Shivakumar, Narayanan, and Garcia-Molina, Hector. SCAM: A copy detection mechanism for digital documents. In *Proceedings of the 2nd International Conference on Theory and Practice of Digital Libraries* (1995).

[123] Shivakumar, Narayanan, and Garcia-Molina, Hector. Building a scalable and accurate copy detection mechanism. In *Proceedings of the 1st ACM Conference on Digital Libraries* (1996).

[124] Shivakumar, Narayanan, and Garcia-Molina, Hector. Finding near-replicas of documents on the web. In *Proceedings of the Workshop on Web Databases (WebDB)* (1998).

[125] Smucker, Mark, and Allan, James. An investigation of dirichlet prior smoothing's performance advantage. Tech. Rep. IR-391, The University of Massachusetts, The Center for Intelligent Information Retrieval, 2005.

[126] Soboroff, Ian. Overview of the TREC 2004 novelty track. In *Proceedings of the Thirteenth Text Retrieval Conference (TREC)* (2004).

[127] Soboroff, Ian, and Harman, Donna. Overview of the TREC 2003 novelty track. In *Proceedings of the Twelfth Text Retrieval Conference (TREC)* (2003).

[128] Sparck Jones, Karen, and Willett, Peter, Eds. *Readings in Information Retrieval*. Morgan Kaufmann Publishers, 1997.

[129] Spence, Clay, Parra, Lucas, and Sajda, Paul. Detection, synthesis and compression in mammographic image analysis with a hierarchical image probability model. In *IEEE Workshop on Mathematical Methods in Biomedical Image Analsysis* (2001), pp. 3–10.

[130] Tanev, Hristo, Kouylekov, Milen, and Magnini, Bernardo. Combining linguistic processing and web mining for question answering: ITC-irst at TREC-2004. In *Proceedings of the Thirteenth Text Retrieval Conference (TREC)* (2004).

[131] Tarassenko, Lionel. Novelty detection for the identification of masses in mammograms. In *Proceedings of the 4th IEEE International Conference on Artificial Neural Networks* (1995), vol. 4, pp. 442–447.

[132] Turtle, Howard, and Croft, W. Bruce. Inference networks for document retrieval. In *Proceedings of the 13th Annual Conference on Research and Development in Information Retrieval (SIGIR)* (1990).

[133] Voorhees, Ellen M. On expanding query vectors with lexically related words. In *Proceedings of the Text REtrieval Conference (TREC)* (1993), pp. 223–232.

[134] Voorhees, Ellen M. The TREC-8 question answering track report. In *Proceedings of the Eighth Text Retrieval Conference (TREC)* (1999).

[135] Voorhees, Ellen M. Overview of the TREC-9 question answering track. In *Proceedings of the Ninth Text Retrieval Conference (TREC)* (2000).

[136] Voorhees, Ellen M. Overview of the TREC 2001 question answering track. In *Proceedings of the Tenth Text Retrieval Conference (TREC)* (2001).

[137] Voorhees, Ellen M. Overview of the TREC 2002 question answering track. In *Proceedings of the Eleventh Text Retrieval Conference (TREC)* (2002).

[138] Voorhees, Ellen M. Overview of the TREC 2003 question answering track. In *Proceedings of the Twelfth Text Retrieval Conference (TREC)* (2003).

[139] Voorhees, Ellen M. Overview of the TREC 2004 question answering track. In *Proceedings of the Thirteenth Text Retrieval Conference (TREC)* (2004).

[140] Wu, Lide, Huang, Xuanjing, You, Lan, Zhang, Zhushuo, Li, Xin, and Zhou, Yaqian. FDUQA on TREC2004 QA track. In *Proceedings of the Thirteenth Text Retrieval Conference (TREC)* (2004).

[141] Xu, Jinxi, and Croft, W. Bruce. Query expansion using local and global document analysis. In *Proceedings of the 19th Annual Conference on Research and Development in Information Retrieval (ACM SIGIR)* (1996), pp. 4–11.

[142] Xu, Jinxi, Fraser, Alexander, and Weischedel, Ralph. Empirical studies in strategies for arabic retrieval. In *Proceedings of the 25th Annual Conference on Research and Development in Information Retrieval (ACM SIGIR)* (2002).

[143] Yamada, Kenji, and Knight, Kevin. A syntax-based statistical translation model. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL)* (2001).

[144] Yeung, Dit-Yan, and Chow, Calvin. Parzen-window network intrusion detectors. In *Proceedings of the Conference on Pattern Recognition* (2002).

[145] Yeung, Dit-Yan, and Ding, Yuxin. Host-based intrusion detection using dynamic and static behavioral models. *Pattern Recognition 36*, 1 (2003), 229–243.

[146] Yu, Man-Quan, Luo, Wei-Hua, Zhou, Zhao-Tao, and Bai, Shuo. ICT's approaches to HTD and tracking at TDT2004. In *Working Notes of the Topic Detection and Tracking (TDT) Evaluation* (2004).

[147] Zhai, ChengXiang, and Lafferty, John. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (2001), pp. 334–342.

[148] Zhang, Hua-Ping, Xu, Hong-Bo, Bai, Shuo, Wang, Bin, and Cheng, Xue-Qi. Experiments in TREC 2004 novelty track at CAS-ICT. In *Proceedings of the Thirteenth Text Retrieval Conference (TREC)* (2004).

[149] Zhang, Jian, Ghahramani, Zoubin, and Yang, Yiming. A probabilistic model for online document clustering with application to novelty detection. In *Advances in Neural Information Processing Systems 17*, Lawrence K. Saul, Yair Weiss, and Léon Bottou, Eds. MIT Press, Cambridge, MA, 2005, pp. 1617–1624.