Language Modeling Approaches to Question Answering

A Dissertation

Submitted to the Faculty

of

Drexel University

by

Protima Banerjee

in partial fulfillment of the

requirements for the degree

of

Doctor of Philosophy

July 2009

Acknowledgements

I would like to thank the members of my dissertation committee for their participation in my dissertation proposal and final defenses, for reviewing my papers, and for the many comments and suggestions which improved this work. I would particularly like to thank my committee chair, Dr. Hyoil Han, for years of advice, support and the academic and personal guidance without which my studies would have floundered long ago. The members of my dissertation committee are:

Dr. Hyoil Han, Advisor and Chairperson
Dr. Il-Yeol Song
Dr. Chaomei Chen
Dr. Jung-ran Park
Drexel University, College of Information Science and Technology
Philadelphia, PA 19104

Dr. Daniel McFarlane
Lockheed-Martin, Advanced Technology Labs
Cherry Hill, NJ 08002

I would also like to thank my management team at Lockheed-Martin Corporation for appreciating my academic goals, and for allowing me the flexibility to balance my academic and professional interests. Most importantly, I would like to thank my husband Christopher and my daughter Maya for their never-ending understanding, encouragement and assistance.

Table of Contents

List of Tables

List of Figures

Abstract

Language Modeling Approaches to Question Answering

Protima Banerjee

Hyoil Han, Ph.D.

In today's environment of information overload, Question Answering (QA) is a critically important research area. QA is the task of automatically extracting a precise answer from one or more data sources to a question posed in natural language. A two-stage strategy is typically adopted when designing a QA system; the first stage is an Information Retrieval (IR) process which returns a set of candidate documents relevant to the question and the second stage narrows the information contained in those passages down to a single response (sentence or entity) that answers the question, typically using Information Extraction (IE) or Natural Language Processing methods. This research proposes novel techniques for QA by enhancing the user's original query with latent semantic information from the corpus. This enhanced query is then applied to both the first and second stages of the QA architecture. To build the enhanced query, we propose the Aspect-Based Relevance Language Model as an approach that uses statistical language modeling techniques to measure the likelihood of relevance of a concept (or aspect as defined by Probabilistic Latent Semantic Analysis) to a question. We then use terms from the aspects that have the highest likelihood of relevance to design a model for a semantic Question Context, which includes sense-disambiguated terms than amplify the user's query. Question Context is incorporated into the first state of QA as query

expansion to improve recall. We then derive a novel measure called Answer Credibility from the Question Context. Answer Credibility may be thought of as a statistical measure of the reliability of a candidate answer with respect to a question and the source text from which the candidate answer was derived. We incorporate Answer Credibility in the Answer Validation process; the answer with the highest score after the application of Answer Credibility is returned to the user. Our techniques show performance improvements over state-of-the-art approaches, and have the advantage that they use statistical techniques to derive semantic information to aid the process of QA.

1. INTRODUCTION

In today's environment of information overload, Question Answering (QA) is a critically important research area. To make effective use of the massive amounts of readily available data, humans need efficient tools that will bypass irrelevant information to find the precise "nuggets" of data that answer their specific questions. The user of a QA system may range from a homeowner who is searching for a method to remove carpet stains to an intelligence analyst searching for signs of potential terrorists in a major metropolis. In either case, the human ability to synthesize, analyze and evaluate all of the available information sources is limited by both time and cognitive capacity; in order for us to make effective use of information we require automated tools to help us make sense of data, especially when our questions are time-critical. It is the goal of automated QA systems to streamline information processing, and to ensure that the human user is presented with the right information in a timely manner.

It should be noted that Question Answering, while related to Information Retrieval (IR) and Information Extraction (IE), is distinct from both these fields. QA goes beyond entity identification, which is the primary task of an IE system. A QA system must pinpoint the specific entity that answers a question, and often times answering a question correctly requires the proper identification of complex relationships between entities. In a similar vein, a QA system transcends traditional IR; a QA system returns snippets of valuable information rather than entire passages or documents which then need to be scanned by a user. Another key distinction between QA and IR is the importance of identifying the absence of relevant information. If a QA system is unable

to answer a given question with the data available to it, this must be communicated to the user. In other words, in QA a wrong answer is worse than no answer at all.

A two-stage strategy is typically adopted when designing a QA system; the first stage is an IR process which returns a set of candidate documents and/or passages relevant to the question and the second stage narrows the information contained in those passages down to a single response (sentence or entity) that answers the question. This proposal describes the design of a novel system which will incorporate statistical language modeling techniques into both the first and second QA stages, and then evaluate the effectiveness of our techniques using metrics that have been well-established in the TREC Question Answering Track (E. M. Voorhees & D. Harman, 2005).

Our approach to QA focuses on the problem of sparse queries; that is, a typical query submitted to a QA system contains relatively few terms, and virtually no supporting information that amplifies those terms. The document corpus, on the other hand, is a rich source of textual information about the specific terms that appear in the query. As such, we propose to enhance the user's original query with latent semantic information from the document corpus. This enhanced query is then applied to both the first and second stages of the QA architecture. This enhanced query is then applied to both the first and second stages of the QA architecture. To build the enhanced query, we propose the Aspect-Based Relevance Language Model as an approach that uses statistical language modeling techniques to measure the likelihood of relevance of a concept (or aspect as defined by Probabilistic Latent Semantic Analysis) to a question. We then use terms from the aspects that have the highest likelihood of relevance to design a model for a semantic Question Context, which includes sense-disambiguated terms than amplify the

user's query. Question Context is incorporated into the first state of QA as a query expansion and document smoothing mechanism to improve recall. We then derive a novel measure called Answer Credibility from the Question Context. Conceptually, Answer Credibility may be thought of as a statistical measure of the reliability of a candidate answer with respect to a question and the source text from which the candidate answer was derived. We incorporate Answer Credibility in the Answer Validation process in the second stage of QA to set the answer score. The answer with the highest score after the application of Answer Credibility is returned to the user as the final response of our QA system.

Our research focuses on Question Answering on topics in the open-domain due to availability of evaluation testbeds. We believe that opportunities for application of our techniques into one or more specialty domain areas exist in the future as they have shown significant improvements over the baseline.

1.1 Research Questions

The goal of the proposed research is to investigate and evaluate the incorporation of language modeling techniques into a two-stage Question Answering system. A two-stage Question Answering system can be broadly decomposed into four processes: Candidate Document Selection, Answer Extraction, Answer Validation and Response Generation. In this work, we explore the incorporation of language modeling techniques into the Candidate Document Selection and Answer Validation Question Answering processes. We choose to focus on these processes for the following reasons:

- Candidate Document Selection makes up the bulk of the processing of the first QA stage; in Candidate Document Selection a set of documents which are likely to include the candidate answer are selected from a document corpus. This stage is likely to involve some type of IR mechanism. If we are to consider applying statistical language modeling methodologies to the first stage of Question Answering, we must address how these techniques can be incorporated into Candidate Document Selection.
- To date, the bulk of research related to the second QA stage has focused on Answer Extraction – specifically using mature IE techniques. Furthermore, the Answer Extraction processes focuses on narrowing down large segments of text into one or more specific chunks that might be considered the (exact) correct answer to a question. Statistical language modeling is a generative technique, and is not well-suited to the task of narrowing down large segments of text into very fine-grained sub-sections. For both these reasons, the Answer Extraction process does not seem an applicable target for this research.
- Statistical language modeling techniques are most effective when significant amounts of text are available. Answer Validation focuses on validating a set of candidate answers that are returned from the Answer Extraction process with respect to supporting source text from the corpus. At a high level, one might consider that the Answer Validation process takes two small segments of text, the question and a candidate answer, and seeks to determine what level of support this particular question/answer pair has within the document corpus. In this light, the Answer Validation process lends itself to the application of generative techniques;

one may look at Answer Validation as the likelihood of a particular question/answer pair, given a statistical models of the documents which exist in the corpus.

- For the purposes of this research, Response Generation is viewed as an extension of Answer Validation. Specifically, we consider Response Generation to be simply the selection of the highest ranked answer after the Answer Validation process completes. While this is certainly a simplified view of the Response Generation process we consider it sufficient for the scope of this work.

We propose to answer the following research questions during the course of our work:

1. Can we incorporate statistical language modeling techniques into the Candidate Document Selection stage of the Question Answering architecture, and thereby improve QA performance?
2. Can we design a model for Question Context based on statistical language modeling methods that is able to incorporate latent semantic information that exists within the corpus?
3. Can we incorporate statistical language modeling techniques into the Answer Validation stage of the Question Answering architecture, and thereby improve QA performance?
4. Can we design a model for Answer Credibility based on statistical language modeling methods that can be incorporated into the Answer Validation process that can be used to improve QA accuracy?

1.2 Contributions

The primary contributions of this research are as follows:

1. Formal development of the Aspect-Based Relevance Language Model on the basis of two established statistical language models: Probabilistic Latent Semantic Analysis (PLSA) and Relevance-Based Language Models.
2. Development of a Question Context Model (QCM) based on the Aspect-Based Relevance Language Model.
    a. Our QCM uses only the corpus itself as a knowledge source, and relates the query to one or more sense-disambiguated concepts that exist within in the corpus.
3. This Question Context Model (QCM) is then incorporated into Candidate Document Selection process using query expansion and document smoothing methods, showing an improvement in recall over baseline methods.
4. Then, we use the Question Context Model to derive a model for a novel measure called Answer Credibility. At a high level, Answer Credibility is a measure of the reliability of the source document associated with a candidate answer.
5. We then incorporate Answer Credibility into the Answer Validation process, and show improvements in accuracy and Mean Reciprocal Rank (MRR) that are comparable with state-of-the-art systems.

1.3 Dissertation Organization

This dissertation is organized into six chapters. The first chapter, this chapter, introduces Question Answering and statistical language modeling, describes the contributions of our research, and presents background information. Chapter 2 provides a literature review, broken down into two major areas. The first area is statistical language modeling, and it is reviewed in the large (i.e., not domain specific), with a focus on the application to Information Retrieval. The second area is Question Answering, which has a rich history extending back approximately 60 years. In our review, we give an overview of Question Answering research and then provide detailed reviews of research in the areas of Candidate Document Selection and Answer Validation. Chapter 3 describes the rationale and formal derivation of the Aspect-Based Relevance Language Model. Chapter 4 discusses the design of the Question Context Model (QCM), its application to Candidate Document Selection, our evaluation methodology, and the results of our evaluation. Chapter 5 discusses the design of the Answer Credibility Model (ACM), its application to Answer Validation, our evaluation methodology, and the results of our evaluation. Chapter 6 concludes this dissertation and presents some areas for future work.

1.4 Introduction to Question Answering

Hirschman (Hirschman & Gaizauskas, 2002) proposes an architecture for a generic QA system, which we adopt as the basis for our Question Answering architecture. In this conceptual architecture, five core processes must be performed in order for a question to be successfully answered:

1. Question Analysis, during which a question is interpreted and decomposed into whatever constituent parts are required for subsequent processing. A user model and QA context are helpful during this stage for the incorporation of any a priori knowledge that the system may want to take into account.
2. Document Collection Pre-processing, during which the document corpus is decomposed into whatever format is most helpful for subsequent processing. Some common activities that may occur in this process include stemming, part-of-speech tagging, passage identification, entity extraction and indexing.
3. Candidate Document Selection, during which a set of documents which is likely to include the candidate answer are selected from the processed document corpus. This stage is likely to involve some type of IR mechanism.
4. Answer Extraction, during which a set of candidate answers are extracted from the documents and/or passages returned from the Candidate Document Processing step. This step may involve some Information Extraction or Natural Language Processing techniques; a significant amount of QA research to date has focused on Answer Extraction.
5. Answer Validation and Response Generation, during which the set of candidate answers are validated with respect to supporting source text from the corpus and then reduced to a single response that can then be returned by the system.

When considering the remainder of this research, it may be valuable to consider this generic Question Answering architecture, as presented in Figure 1.

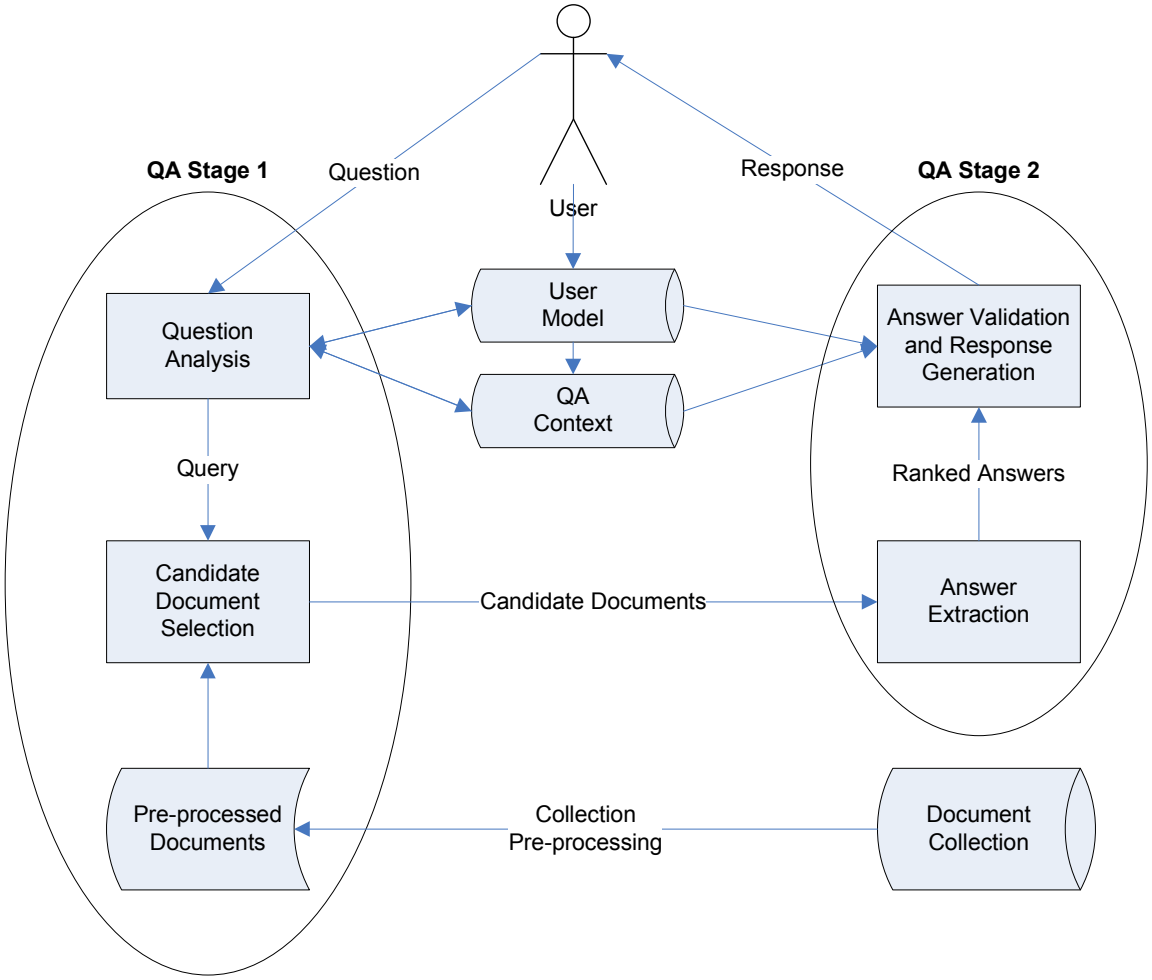

Figure 1: Generic System Architecture for Question Answering

1.5 Introduction to Language Modeling

Language modeling is a formal probabilistic retrieval framework with roots in speech recognition and natural language processing (Jurafsky & Martin, 2000). The underlying assumption of language modeling is that human language generation is a random process; the goal is to represent that process via a statistical model. Using a language model, we can calculate the likelihood of a language sequence, such as a sentence, being generated.

Language models were first successfully applied to information retrieval by Ponte and Croft (Ponte & Croft, 1998). In that work, the authors proposed a query-likelihood model (X. Liu & Croft, 2005) in which a query is considered to be generated from an "ideal" document that satisfies an information need. The retrieval engine estimates the likelihood that each document in the corpus is the ideal document, and then ranks the documents accordingly. The underlying premise to this approach is that each document in the corpus has a different language model (Manning, Raghavan, & Schutze, 2007). This allows the use of statistical techniques to both estimate document models and to score documents against a particular query.

Later works (Lavrenko & Croft, 2001; Song & Croft, 1999; C. Zhai & Lafferty, 2004) expand on this early research using more sophisticated models that include topics, phrases and relevance. All confirm that language modeling techniques are preferred over *tf-idf* (term frequency-inverse document frequency) weights (S. Robertson & Jones, 1997)**,** because of empirical performance as well as the probabilistic meaning that can be formally derived from a language modeling framework. In contrast to the classic vector space model (Salton & McGill, 1986) which produces a geometric document score, a language model produces a likelihood estimate which is intuitively easier to understand. The majority of language modeling approaches to information retrieval can be categorized into one of four groups: a) the generative query likelihood approach, which ranks based on the likelihood of a document language model generating the query b) the generative document likelihood approach, which ranks based on the likelihood of a query language model generating a document c) the comparative approach, which ranks based on the similarity between the query language model and document language model, and

d) translation models, which rank based on the likelihood of the query being viewed as a translation of a document. Two language models that are especially foundational to this work are Probabilistic Latent Semantic Analysis (PLSA) and the Relevance-Based Language Model; these are discussed first within the context of the literature in Chapter 2 and explored in more detail as a part of our theoretical framework in Chapter 3. A full survey of language modeling techniques and their application to Information Retrieval (IR) is presented in the following chapter.

2. LITERATURE REVIEW

The review of literature is divided into five sections. The first two sections are focus on language modeling. Section 2.1 describes statistical language modeling; Section 2.2 traces the application of statistical language modeling to Information Retrieval. The next three sections focus on Question Answering. Section 2.3 provides a general perspective on QA. Sections 2.4 and 2.5 describe current research in Candidate Document Selection and Answer Validation respectively.

2.1 Statistical Language Modeling

The roots of statistical modeling date back to the middle of the twentieth century with the work of Shannon (Shannon, 1951) . Shannon proposed a conceptual framework in which a subject was asked to guess the next letter in a sequence from a stream of printed English, given the preceding letters and words in the passage. The subject is informed if he guessed correctly; if he guessed incorrectly, he is told the correct letter in the sequence and can then proceed to the next letter. Only the letters that are incorrectly guessed, that is, the letters where the system must provide the correct letter to the subject, are written down. The next subject in the experiment would then be asked to re-construct the original sequence from the stream of incorrectly guessed letters, which Shannon calls the "reduced text." The underlying premise of this experiment, and the concept which carries through to statistical language modeling, is that a stripped down or lossy version of a full text can be used to reconstruct the complete document.

The work of Zipf (Zipf, 1949) is also fundamental to statistical language modeling. While studying statistical word occurrences in natural language texts, Zipf

postulated that the frequency of word occurrences in a document is roughly inversely proportional to a word's rank within the document, if the words are sorted by occurrence frequency. Conceptually, Zipf's Law says that "while only a few words are used very often, many or most are used rarely (Zipf, 1949)." Zipf's work evidences the applicability of statistical models to natural language; it should be noted that while his most prominent work dealt with English language texts, he also successfully applied the same model to languages such as Chinese, which have a widely divergent grammar from English.

Fundamentally, a statistical language model is a generative pattern of language; that is, it seeks to estimate the probability of occurrence of a sequence of words. Language modeling is predictive in nature; the goal of the model is to estimate the probability of future words, given the pattern of words that we already know (Manning & Schütze, 1999). Language modeling has been applied to a variety of fields including natural language processing, speech recognition, machine translation, and information retrieval. A full survey of statistical language modeling techniques across these divergent applications is beyond the scope of the current paper, and the reader is encouraged to review Rosenfeld's survey on the topic (Rosenfeld, 2000) for an overview. Seminal papers in the field have included the work of Bahl, et. al. (Bahl, Brown, de Souza, & Mercer, 1989), which discusses statistical language modeling in the context of speech recognition, Jelinek and Mercer (Jelinek & Mercer, 1980), which presents a method for estimating language model parameters in a sparse data environment and Katz's method (Katz, 1987) for CPU and storage efficient language model computation.

The perplexity metric is commonly used to assess the effectiveness of a language model (Jelinek, Mercer, Bahl, & Baker, 1977). Perplexity is a type of information entropy measure; mathematically, perplexity is defined by the formula below, where H is the entropy measure for a particular random variable X having a probability distribution $p(x)$.

$$perplexity = 2^{H(X)} = 2^{\sum p(x)\log_2 p(x)} \quad (1)$$

Conceptually, perplexity can be thought of as a confidence measure of the predictive properties of the model. At a word or passage level, the perplexity measure defines how confidently we can define the next word in a sequence given the context of the previous words. Language models can be evaluated against one another by comparing their perplexity measures. It should be noted that perplexity is domain dependent; a given language model will typically have much lower perplexity (better performance) in a highly specialized domain than in general English.

The simplest form of language model is the "bag of words" or unigram model, which examines each word independently of its context (Manning et al., 2007). There are many other more complex types of language models such as bi-gram models or tri-gram models which condition the existence of the next word based on the previous word or two words; words are considered as sets of two or three, respectively in these approaches (Jurafsky & Martin, 2000). Many modern approaches (Rosenfeld, 2000) have used large corpora to train language models; these approaches have typically used simple models such as unigram or bi-gram approaches to language models, relying on the large quantities of training texts of various types to improve performance (Rosenfeld, 2000).

The fundamental problem of language modeling is that we never have a clear confirmation of the specific model that we are assigning to any given document (Manning et al., 2007). This is because the language model is an estimate of word occurrence probabilities based on the text of a document; we treat each document in our collection as a piece of representative text from its underlying language model. Thus, the creation of a language model for a given document is essentially the problem of deriving a complete word occurrence probability model from an incomplete or lossy sample (the document text). In this situation, there are often cases where words which ought to be included as a part of the language model for a document are, in fact, not present in the actual text, even if the size of the text is huge. For example, a document about healthful eating ought to have a high probability of occurrence for the words "diet" and "nutrition" as a part of its language model even if those words are not in the text of the document. However, a document about healthful eating is less likely to contain the word "alligator" since it is less likely that a document about healthful eating will be related to alligators. From these examples one can see that it is problematic to use estimates such as the Maximum Likelihood Estimate which are based strictly on term counts for language model construction. This is often referred to as the "zero frequency problem" or the "sparse data problem (Witten & Bell, 1991)."

Smoothing is a set of techniques used to address the "sparse data" issue. Smoothing is so called because it attempts to raise low word-occurrence probabilities and lower high word-occurrence probabilities, based on prior knowledge about a passage, document or corpus (C. Zhai & Lafferty, 2004). Smoothing is a fundamental part of the language modeling paradigm; "in the language modeling approach, the accuracy of

smoothing is directly related to retrieval performance(C. Zhai & Lafferty, 2004)." In the sections below the smoothing techniques which are applied by each of the language modeling approaches will be explicitly described.

2.2 Language Modeling Approaches to Information Retrieval

Prior to embarking on a comparative study of any set of Information Retrieval (IR) techniques, it is valuable to first engage in a brief discussion of evaluation metrics. In an IR context, performance is most commonly measured via precision and recall. Precision can be described as the ratio of relevant documents returned by the query to the total number of query results. Recall can be described as the ratio of the relevant documents returned by the query to the total number of relevant documents in the collection. We consider the number of documents which are relevant to a given question in the corpus to be those documents which contain the correct answers when calculating the precision and recall metrics. We use the following formulae to describe precision and recall:

$$precision = \frac{| D_{Ret} \cap D_{Rel} |}{| D_{Ret} |} \quad (2)$$

$$recall = \frac{| D_{Ret} \cap D_{Rel} |}{| D_{Rel} |} \quad (3)$$

Here, D_{Ret} is the set of documents that are returned by a particular query and D_{Rel} is the set of documents in the corpus which are relevant to the query.

A metric called Mean Average Precision (MAP) has been introduced to evaluate the performance of retrieval engines for the TREC experiments (Voorhees & Harman, 1999). The Average Precision measure divides a function of the number of documents retrieved

up to a certain cut off ranking by the total number of relevant documents retrieved. "The average precision measure has a recall component in that it reflects the performance of a retrieval run across all relevant documents and a precision component in that it weights documents retrieved earlier more heavily than documents retrieved later (Voorhees & Harman, 1999)." MAP is then the mean value of these Average Precisions across a set of queries and/or topics. In recent TREC experiments (Voorhees, 2005b) a measure called Geometric MAP or GMAP has been introduced, which uses a geometric mean, rather than arithmetic mean to calculate MAP from Average Precision scores. The GMAP measure weights retrieval topics which have a lower MAP most heavily, and it has been used as an evaluation metric for the TREC Robust Track (http://trec.nist.gov/data/robust.html) which looks for engines that perform consistently across a diversity of topics.

Since the introduction of language modeling to information retrieval (IR) by Ponte and Croft (Ponte & Croft, 1998), there have been a number of IR approaches based on language modeling techniques As stated earlier, this paper categories IR research using language modeling techniques into four broad categories: 1) generative query-likelihood models, 2) generative document-likelihood models, 3) model comparison approaches, 4) statistical translation methods and 5) cluster-based language models. The remainder of this section explores the research that has been performed in each of these categories in detail.

2.2.1 Query Likelihood Models

The premise of the query-likelihood approach is that each document in a collection can be thought of as having an individual document language model, and document language models within the collection can be ranked by their probability of being able to generate a particular query. The earliest work in the query-likelihood family of approaches can be considered to be that of Kalt (Kalt, 1996). Kalt considered that term probabilities for documents related to a single topic can be modeled by a single stochastic process; documents related to different topics would be generated by different stochastic processes. Kalt's model treats each document as a sample from a topic language model. Since the problem he considered was text classification, queries were derived from a training set rather traditional query strings. Kalt's approach was based on the Maximum Likelihood Estimate (MLE) (Manning & Schütze, 1999), and incorporated collection statistics, term frequency and document length as integral parts of the model. Although later query-likelihood approaches are more robust in that they consider that each document (vs. a group of documents) is described by an underlying language model, Kalt's early work is clearly a pre-cursor to language modeling in information retrieval.

The Ponte and Croft query-likelihood model (Ponte & Croft, 1998) assumes a unigram language model and, like Kalt's model, starts from the basis of the MLE. Mathematically, MLE is defined below where $p(t|d)$ is the probability of a term given a document d, tf is the term frequency of the given term t, and dl is the total number of terms within the document, which can also be thought of as the document length. Ponte and Croft equate MLE to the probability of the term given the document's language model, or $p(t|M_d)$.

$$p(t \mid M_d) = \frac{tf}{dl} \qquad (4)$$

To smooth the zero probability terms in the MLE, the authors take the background probability of all terms in the collection into account; that is, they augment the probability of the term appearing in any specific document with the probability of the term across the entire document collection. However, as not all the documents in the collection come from the same language model, a geometric risk function is incorporated into the smoothing methodology. This risk function essentially minimizes the impact of documents which might be "outliers" for a given term; in other words, it reduces the impact of those documents which have term frequencies which diverge from the normalized mean of the collection by reducing their contribution to the language model. The risk function *R(t,d)* for term *t* in document *d* is shown below, where f_t is the mean term frequency of term *t* in those documents in which it occurs and tf is the term frequency for term *t* in document *d*.

$$R(t,d) = (\frac{1}{1+f_t})(\frac{f_t}{1+f_t})^{tf} \qquad (5)$$

Ponte and Croft incorporate the risk function into their approach by using it to modulate the influence of the term frequency within the specific document as well as the average occurrence of the term within the corpus. The final formula, including the risk function, for the probability of occurrence of a term within a given language model $p(t|M_d)$ is shown below.

$$p(t \mid M_d) = (\frac{tf}{dl})^{(1-R(t,d))} p_{avg}(t)^{R(t,d)} \qquad (6)$$

Here, $p_{avg}(t)$ is defined to be the average occurrence of a term *t* within each document in the corpus in which it exists.

To test their approach, the Ponte and Croft implement a research prototype called Labrador (Ponte & Croft, 1998) and perform experiments using queries and data from past TREC collections (http://trec.nist.gov/data.html). The results are evaluated against baseline results from the INQUERY retrieval engine, which uses a *tf-idf* retrieval approach combined with inference networks (Callan, Croft, & Harding, 1992). The results showed that the language model approach outperformed *tf-idf* for both query sets evaluated.

In a later work, Song and Croft (Song & Croft, 1999) apply a more generic view to the language modeling problem based on the same query-likelihood viewpoint; they evaluate a number of approaches to language modeling and propose a set of improvements that can either be used independently or in conjunction with one another. Specifically, they focus on smoothing document language models with the Good-Turing estimate (Manning & Schütze, 1999) and curve fitting, expanding a document model with the corpus document model, modeling a query as a sequence (rather than a set) of terms, and finally combining the unigram language model with a bi-gram language model.

The Good-Turing estimate adjusts the raw term frequency scores *tf* in the following manner:

$$tf = (tf + 1)\frac{E(N_{tf+1})}{E(N_{tf})} \qquad (7)$$

Here, N_{tf} represents the number of terms that have term frequency *tf*, and $E(N_{tf})$ is the expected value of the number of terms that have a term frequency of *tf*. Intuitively, the Good-Turing estimate states that the ratio of two adjacent term frequencies will be equivalent to the ratio of the expected values of the number of terms that have those frequencies. Practically, however, it is difficult to determine the number of terms that

have specified frequencies due to the limited data available within a document; too many terms may have frequencies close to zero to make the Good-Turing estimate useful. Song and Croft adopt a curve fitting approach that uses a geometric distribution with a nested logarithmic exponent to approximate $E(N_{tf})$.

Song and Croft use the Good-Turing estimate to create a smoothed language model for both the individual document as well as the corpus. Then, they apply a weighted sum to combine the document language model with the corpus language model. The weighted sum approach is represented below:

$$p(t \mid d) = wP_{document}(t \mid d) + (1-w)P_{corpus}(t) \qquad (8)$$

The weighting parameter w is a number between 0 and 1, and the weighted sum approach of combining probabilities has the advantage of always producing a normalized result; in other words, the weighted sum approach will always produce a number between 0 and 1 for $p(t|d)$.

Interpolation is also used to combine the unigram document model with a bi-gram document model. This is represented below:

$$p(t_i, t_{i-1} \mid d) = \lambda_1 p(t_i \mid d) + \lambda_2 p(t_i, t_{i-1} \mid d) \qquad (9)$$

Here, the weighting parameters λ_1 and λ_2 should be set so that $\lambda_1 + \lambda_2$ is equal to 1 for every term t. This is done empirically by Song and Croft; however, the Expectation Maximization algorithm (Dempster, Laird, & Rubin, 1977) can be used to set these parameters using a training corpus.

In terms of query processing, Song and Croft treat the query as a sequence of terms, as opposed to the set of terms approach adopted by Ponte and Croft. This can be represented as follows:

$$P_{sequence}(Q \mid d) = \prod_i p(t_i \mid d) \qquad (10)$$

The experiments conducted by Song and Croft on Wall Street Journal and TREC4 data show that their generic approach performs better than both INQUERY and the original Ponte and Croft approach (Ponte & Croft, 1998).

Hiemstra (Hiemstra, 1999) presents a related approach which applies statistical language modeling to information retrieval. Hiemstra's approach emphasizes the importance of the ordering of the terms in the document. That is, "the most important modeling assumption we make is that a document and a query are defined by an ordered sequence of words and terms.(Hiemstra, 1999)." In this framework, text is modeled as an ordered sequence of *n* random variables, one for each unique term which appears in the document. This is represented by the following models for documents and queries:

$$P(T_1, T_2, ...T_N \mid D) = \prod_i P(T_i \mid D) \text{ and } P(T_1, T_2, ...T_N \mid Q) = \prod_i P(T_i \mid Q) \qquad (11)$$

where D and Q are the documents and queries respectively and T_i is the event that term i occurred in document D or query Q. The matching process between a query and a document are represented by the following formula:

$$P(D \mid Q) = \sum_\tau P(\tau \mid Q) P(D \mid \tau), \tau = T_1, T_2, ...T_N \qquad (12)$$

Here τ represents the set of all possible term sequences T_1, T_2,..T_N.

Hiemstra also addresses the sparse data problem. "We believe," he states, "that the sparse data problem is exactly the reason that it is hard for information retrieval systems to obtain high recall values without degrading values for precision." Hiemstra smoothes the probability distribution for each term using a linear interpolation of term frequency and document frequency. This is represented by the equation below:

$$P(T_i = t_i \mid d) = \alpha_1 \frac{df(t_i)}{\sum_t df(t)} + \alpha_2 \frac{tf(t_i, d)}{\sum_t tf(t, d)} \quad (13)$$

Here, $df(t_i)$ is the number of documents in which term t_i appears, which is also known as the document frequency and $tf(t_i,d)$ is the number of times term t_i; appears in document d, which is also known as the term frequency. The document frequency term in the equation above can be thought of as the smoothing contribution coming from the document corpus, while the second term can be thought of as the contribution coming from an individual document. α_1 and α_2, are the weighting coefficients for the document frequency and term frequency contribution to the smoothing model, respectively. Hiemstra does not specifically proscribe a method for setting α_1 and α_2. "In general," he says, "one wants to find the combination of weights that works best, for example, by optimizing them on a test collection consisting of documents, queries, and corresponding relevance judgments (Hiemstra, 1999)." It should be noted that this smoothing approach can be directly related to the well-known *tf-idf* approach.

Hiemstra's smoothing model is similar to Song and Croft's smoothing model in that both approaches are fundamentally a linear combination of a corpus document model (Hiemstra's document frequency component) and an individual document model (Hiemstra's term frequency component). However, there is one fundamental differences between the two approaches: Song and Croft smooth the document and corpus language model (via the Good-Turing estimate) prior to the linear interpolation; Hiemstra combines the two models without smoothing (Song & Croft, 1999). Hiemstra validates his approach by running experiments on the Cranfield collection (Cleverdon, Mills, & Keen, 1966), and comparing the results against the vector space model (Salton & McGill,

1986). "The linguistically motivated approach performed better for a wide range of different α_1 and α_2. (Hiemstra, 1999)."

A two-stage Hidden Markov Model forms the basis for the query-likelihood language model presented by Miller, Leek and Schwartz (D. R. H. Miller, Leek, & Schwartz, 1999). A discrete Hidden Markov Model is defined by "a set of output symbols, a set of states, a set of probabilities for transformations between the states, and a probability distribution on output symbols for each state (D. R. H. Miller et al., 1999)." The Hidden Markov Model is referred to as "hidden" because the state transition process itself is never directly observed; it can only be inferred based on observed events. Hidden Markov Models are widely used in natural language processing and speech recognition (Jurafsky & Martin, 2000).

The first stage in the proposed Hidden Markov Model represents the probability that a given query term will be found within the document; the second stage in the proposed Hidden Markov Model represents the probability that a given query term will be found within General English but is unrelated to the document The proposed model makes the simplifying assumption that a given query term will move to either the first or second stage; thus only two stage changes are considered to be part of the model. Miller et. al. (D. R. H. Miller et al., 1999) make simplifying assumptions similar to Ponte and Croft (Ponte & Croft, 1998) and Hiemstra (Hiemstra, 1999). Rather than using the Expectation Maximization algorithm (Dempster et al., 1977) to compute transition probabilities and output distributions, the simpler MLE method is used. In this model, the "General English" stage of the Hidden Markov Model provides a smoothing function. This stage is approximated by using the entire document corpus as an approximation for

the full English language. A graphical representation of the two-stage Hidden Markov Model is depicted in Figure 2 below.

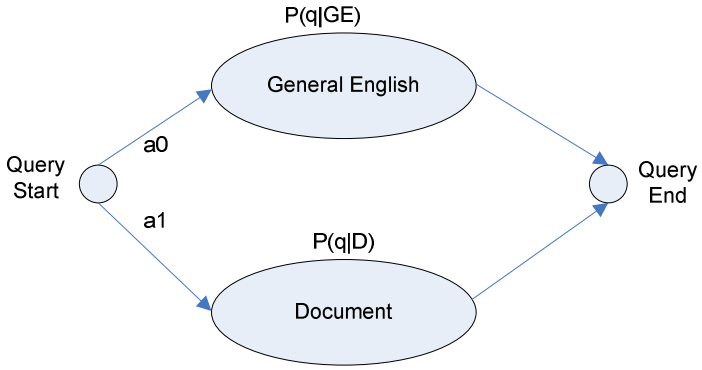

Figure 2: The Two-stage Hidden Markov Model Approach to Modeling Query-Likelihood Language Models

Mathematically, the two-stage Hidden Markov Model is represented by the following equation:

$$P(Q \mid D) = \prod_{q \in Q} a_0 P(q \mid GE) + a_1 P(q \mid D) \qquad (14)$$

where *P(Q|D)* is the likelihood of the query being a representation of the document, *P(q|D)* is the probability that an individual query term *q* will be found in the document stage of the HMM, and *P(q|GE)* is the probability that the query term will be found in the General English stage of the HMM, but not in the document stage. The coefficients a_1 and a_0 are used to weight the respective components of the equation.

Experimentally, ad hoc retrieval experiments were run which compared the performance of the Hidden Markov Model based approach against *tf-idf* on the TREC-6 and TREC-7 collections (http://trec.nist.gov/data.html). The HMM based approach outperformed tf-idf on queries against document titles, document descriptions, document narratives and against the full text of documents. After the results of the initial experimentation, the authors included four refinements to their system to further increase

performance: blind feedback, bi-gram modeling, feature dependent priors and query section weighting. Of these, blind feedback yielded the most significant improvements. The blind feedback approach taken within the Hidden Markov Model framework augments the initial query with words appearing in the top ranked documents, and then adjusting the state transition probabilities to account for increased likelihood of these terms.

2.2.2 Document Likelihood Models

The premise of the document-likelihood approach is that a language model can be generated for the query, and documents within the collection can be ranked by their probability of having been generated by the query's language model. In practice, this approach is most often used to enable expansion-based feedback as the terms in a query are generally too sparse to produce a reliable language model. Zhai and Lafferty (C Zhai & Lafferty, 2001) first introduce the idea of a query language model, and propose two methodologies for query language model construction: 1) a generative model of feedback documents and 2) a model that minimizes divergence over feedback documents. The generative model is a mixture model that generates a feedback document by mixing the query terms with a collection language model, which is taken to be a reasonable model of irrelevant content in a document. A document can then be generated from the resulting language model if it contains either the query terms or the collection language model. Conceptually, the divergence minimization language model estimates the query model by minimizing the average divergence between the query terms and the feedback documents. The estimated resulting query model is close to each feedback document

model; however, in order to minimize the effect general terms that may be common to all the feedback documents, a regularization term is added to prefer documents that have a greater divergence from the collection language model.

Empirically, the results of this feedback methodology were compared to a *tf-idf* approach with tuned Rocchio feedback on three TREC collections. When compared with the tuned Rocchio approach, both model-based feedback methods performed better in terms of precision.

The relevance model proposed by Lavrenko and Croft (Lavrenko & Croft, 2001) can also be thought of as a document-likelihood model. Conceptually, the relevance model is a description of a user's information need which manifests itself in the form of a query. Given a collection of documents and a user's query, there exists a set of documents that are relevant to that query in the user's judgment. The ideal relevance model for a given query run on a specified document collection would be constructed from only the set of relevant documents within the collection; the relevance model in this framework is assumed to be a language model to which word probabilities are assigned. Each document relevant to the user's query then simply becomes a sample from the underlying relevance model.

The problem with this scenario is that in a typical retrieval environment we do not know the full set of relevant documents to a query and furthermore, we may not have any examples of documents which are relevant to the query. Lavrenko and Croft (Lavrenko & Croft, 2001) suggest a methodology that constructs a relevance model from a set of top ranked documents returned from a query. A relevance model is formally defined as the probability of observing a word w in a set of relevant documents R, or $p(w|R)$. The query

q is also treated as a sample from *R*, although the sampling process that produces q is not necessarily the same as the process that generates *w*. Lavrenko and Croft formally derive a process whereby $p(w|R)$ can be estimated via $p(w|M_D)$, where M_D is the document model for a limited set of top-ranked documents returned from the query. They describe $p(w|M_D)$ as follows:

$$p(w \mid M_D) = \lambda \frac{tf(w,D)}{\sum_v tf(v,D)} + (1-\lambda) P(w \mid G) \qquad (15)$$

Here $tf(w.D)$ is the number of occurrences of *w* in *D*, $\Sigma_v tf(v,D)$ is the total number of occurrences of all terms *v* in *D*, and $P(w|G)$ is the collection frequency of *w* divided by the total number of terms in the collection. The smoothing parameter λ is set empirically. This approach is elegant in that it can easily incorporate common information retrieval procedures which would not otherwise fit cleanly into a language modeling framework such as pseudo-relevance feedback or true relevance feedback. A linear interpolation method is used to smooth the MLE document models with the background model of English; smoothing parameters are set experimentally.

Subsequent work in relevance models by Li (X. Li, 2005) treats a query as a short, special document and includes it in the documents that are used to approximate the relevance model to improve the robustness of the relevance modeling approach. In addition, instead of using a uniform prior as in the original relevance model, documents are assigned different priors based on their lengths and the probability of a term in the language model is adjusted by its probability in the language model of the corpus. This variant of the relevance modeling approach was applied to both a pseudo relevance feedback and true relevance feedback environment, and compared with the Ponte and Croft (Ponte & Croft, 1998) query-likelihood approach as well as against the original

Lavrenko and Croft (Lavrenko & Croft, 2001) relevance model. The model proposed by Li outperformed both the other approaches in terms of mean average precision in the area of document retrieval with both pseudo relevance feedback and true relevance feedback.

In later works, Lavrenko et. al. (Lavrenko, Choquette, & Croft, 2002) extend the relevance modeling approach to Cross-Language Information Retrieval (CLIR). The proposed method "constructs an accurate relevance model in the target language, and uses that model to rank the documents in the collection (Lavrenko et al., 2002)." Their approach discusses two estimation strategies: one that assumes a parallel corpus (eg. documents discussing the same topic in both query and target languages), and the other that assumes the existence of bilingual lexicon. In the former case, a joint probability model of word observations across the two languages is constructed; that is, a probability model is constructed that describes co-occurrence of words across languages that exist in documents related to the same topic. In the latter case, it is the lexicon which provides translation probabilities between words across the two languages. For cross-language information retrieval, the authors diverged from their earlier approach of ranking retrieved documents by the Probability Ranking Principle; instead they use Kullback-Leibler divergence (Kullback & Leibler, 1951) as they found that this is "a more stable metric." Experimentally, this model was run on the TREC9 Cross-Language Track dataset against a mono-lingual baseline, and performed at 90-95% of the mono-lingual baseline in terms of average precision.

2.2.3 Model Comparison Approaches

The model comparison approach is first introduced by Lafferty and Zhai (Lafferty & Zhai, 2001). In their risk minimization framework, "queries and documents are modeled using statistical language models, user preferences are modeled through loss functions, and retrieval is cast as a risk minimization problem (Lafferty & Zhai, 2001)." Within this context, a query is viewed as the output of a probabilistic process associated with a user *U* and a document is viewed as the output of a probabilistic process associated with an author or document source *S*. A user first selects an internal model Φ_Q having a probability distribution $p(\Phi_Q|U)$. A particular query *q* is then generated based on the parameters of that internal model with a probability of $p(q|\Phi_Q)$. Similarly, a document source selects an internal model Φ_D according to probability $p(\Phi_D|S)$, and then the probability of a generation of a particular document is given by the probability $p(d|\Phi_D)$. The two Markov chains that represent this risk minimization process are shown in Figure 3 below.

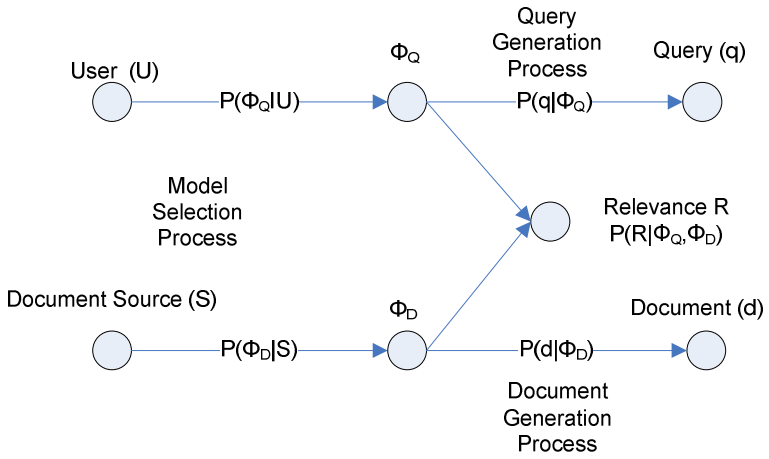

Figure 3: Representation of the Query and Document Generation Processes of the Risk-Minimization Framework Proposed by Lafferty and Zhai

Here, relevance *R* is a binary variable which is 1 if a given document is relevant to a user's query and 0 otherwise. Relevance is described by the distribution $p(R|\Phi_Q,\Phi_D)$.

In the Bayesian decision framework, there is an expected risk associated with every action of a given system. In this context, the particular action that the system performs is returning a document d_i. in response to a query. The risk function for an action a can be modeled by understanding the loss $L(a)$ associated with that function. The function $R(d_i,q)$ which describes the risk associated with returning a particular document d_i, in response to a query is shown in the equation below:

$$R(d_i,q) = \sum_{R\in\{0,1\}} \int_{\phi_Q} \int_{\phi_D} L(\phi_Q,\phi_D,R) \times p(\phi_Q \mid q,U)\, p(\phi_D \mid d_i,S)\, p(R \mid \phi_Q,\phi_D)\, d\phi_D\, d\phi_Q \qquad (16)$$

This is the basic retrieval formula based on risk minimization proposed by Lafferty and Zhai, which is used to calculate the ranking of documents d_i, returned in response to a query q. Lafferty and Zhai show how this risk minimization framework can be used to derive the "special cases" of the classical probabilistic model using a relevance-based loss function, and the query likelihood language modeling approach using a distance-based risk function. The Kullback-Leibler (KL) divergence model which is later elaborated in (C Zhai & Lafferty, 2001), is presented as a special case of the more general risk-minimization framework. In the KL-divergence model, the relevance value of a document with respect to a query is measured by the probabilistic Kullback-Leibler divergence between the query model and document model. The problem of matching a query to a document thus reduces to a similarity or "distance" comparison which is similar to the classic vector-space model.

To address the sparse data problem, the authors explore an approach to query and document language model expansion based on Markov chains which is motivated by statistical translation methods of Berger and Lafferty (Berger & Lafferty, 1999). The intent of the Markov chain is to model the user's browsing process, and the chain

proposes a random walk alternating between queries and documents. Conceptually, the user is "surfing" through the word index for a given document collection, viewing the documents which contain that word, and then refining their information need as they go along. Practically, this approach calculates "the posterior probabilities of words according to the translation model for generating the query and a prior distribution on initial terms selected by the user (Lafferty & Zhai, 2001)." Mathematically, this approach can be represented by the following formula describing the probability of occurrence for a word *w* in a query *q*:

$$p(w \mid \phi_q) \propto \sum_i t(q_i \mid w) p(w \mid U) \qquad (17)$$

Here, $p(w|\Phi_q)$ is an estimate for the language model, $t(q_i|w)$ is the translation model which describes the likelihood of translation between query term q_i and word *w*, and $p(w|U)$ is the likelihood that the user will start their "surfing" from the initial word *w*. The analogous formula for document expansion is shown below:

$$p(w \mid \phi_d) \propto t(d \mid w) p(w \mid U) \qquad (18)$$

Experimentally, the risk minimization framework and query and document expansion methods were evaluated against three different TREC collections, and the results showed improved performance over baseline *tf-idf* methods. The approach showed the greatest performance improvements over the baseline approach when using short queries against TREC web data.

2.2.4 Statistical Translation Models

In their statistical translation model, Berger and Lafferty (Berger & Lafferty, 1999) propose that the formulation of a query is really the distillation of a user's

information need into a succinct form. That distillation from a "fat" document to a "skeletal" query, the authors propose, "is a form of translation from one language to another (Berger & Lafferty, 1999)." Berger and Lafferty represent this document to query translation process using a statistical model. They characterize the translation process as having two stages. In the first stage, the translation analyst chooses a word *w* from the document according to a distribution *l(w|d)* which is called the document language model. In the second stage, that word *w* is translated into a word or phrase q using a translation model *t(q|w)*. Thus, the statistical translation model for a single query term *q* can be described by the following equation:

$$p(q \mid d) = \sum_{w \in d} l(w \mid d) t(q \mid w) \qquad (19)$$

This model must be applied *n* times to account for a query containing *n* terms; the number n in this model is chosen according to a sample size model *Φ(n|d)*. Berger and Lafferty propose that a Poisson distribution with mean *λ(d)* can be used to calculate the sample size model.

$$\phi(n \mid d) = e^{-\lambda(d)} \frac{\lambda(d)^n}{n!} \qquad (20)$$

Applying this assumption leads to the complete statistical translation model, which Berger and Lafferty call Model 1:

$$p(q = q_1, q_2, ..., q_n \mid d) = e^{-\lambda(d)} \prod_i e^{\lambda(d) p(qi|d}) - 1 \qquad (21)$$

It should be noted that if each word can be translated only to itself (eg. p(q|t) = 1 only when q=t) this model decomposes into the query-likelihood model proposed by Ponte and Croft (Ponte & Croft, 1998). Berger and Lafferty call this simplest version of the translation model Model 0.

The parameters for the translation model were set empirically using the Expectation Maximization (EM) algorithm (Dempster et al., 1977). A simple linear mixture model is used to combine the background unigram model for the corpus and the EM-trained translation model. The smoothing parameters were derived empirically by optimizing the algorithm on the TREC Spoken Document Retrieval data. While significant improvements over a baseline *tf-idf* approach were reported by this method, the experiments required a large corpus of training data which may not be practical in all cases. An interesting corollary to this approach is that it is naturally extensible to the problem of multi-lingual information retrieval.

Murdock and Croft (Murdock & Croft, 2004) extend the translation model approach to sentence retrieval. The motivation for this application of translation models stems from a Question Answering (QA) application; in general, most QA systems use a passage retrieval system for the first stage of processing. The more accurately a passage retrieval system can retrieve succinct segments of text, the better the QA system is likely to perform overall. A high-quality sentence retrieval system would be a strong candidate for application to the QA domain. However, sentences, which are much smaller, than documents are too short to accurately estimate a language model. Murdock and Croft (Murdock & Croft, 2004) approach this problem by using a translation model to judge similarity between a words in query and words in a candidate sentence. The translation model allows for a looser matching strategy which identifies the relationships between corresponding terms which mean the same thing or are related to one another, but which are not the same term. Murdock and Croft make use of the IBM translation Model 1 (Brown et al., 1990) to rank documents according to their translation probability, given

the query. IBM Model 1 assumes that all alignments are possible between the source sentence and target sentence, and constructs synthetic training data in the absence of the availability of a training corpus of queries and relevant documents. Empirically, the results of this model were compared against query-likelihood with and without smoothing, and showed significant improvements over the baseline.

In related research, Jin and Hauptmann (Jin & Hauptmann, 2001) use language models for document title generation. Rather than using the document directly as the knowledge source for the document title generation, they introduce the idea of a "distilled information source" which is a sample of important content words from the original document. The optimal title for the document can then be generated from this distillation. The underlying premise of this method is that title generation is a "reverse" information retrieval task – in other words, the perfect title for a document would be the same as the query for which that document is an "ideal" response. In an information retrieval paradigm, the generated titles for each document can be evaluated against the queries that are presented to the system and the documents ranked accordingly. This approach is novel in that it provides a bridge from language modeling and information retrieval to the related tasks of text summarization and categorization.

2.2.5 Summarization of Early LM Approaches to IR

Prior to moving on to more recent work in cluster-based language modeling, we feel it is valuable to consider a summary of the early work in language modeling for information retrieval, as almost all later works build upon a variation of one of these approaches. We present this summary in Table 1 below.

Table 1: Chronological Summary of Early Language Modeling Approaches to Information Retrieval

Method	Language Model	Basis	Smoothing Approach	Training Data Requirements	Significance
Ponte-Croft Query Likelihood Model	Uni-gram	Maximum Likelihood Estimate (MLE)	Risk function which incorporates background word probabilities from the collection	Not required	Initial usage of LM in IR
Hiemstra Model	Uni-gram	Document is modeled as an ordered sequence of random variables	Linear interpolation term frequency and document frequency	Required to set smoothing coefficients	Modeling of documents and queries as an ordered sequence of terms; relation of smoothing approach to tf-idf
Song and Croft, Query Likelihood Model	Linear interpolation of Uni-gram and Bi-gram	Query-likelihood model	Good-Turing estimate and curve fitting, incorporating corpus document model	Required to set interpolation parameters required to combine unigram and bi-gram models	Creation of an extensible LM approach to IR
Miller – Hidden Markov Model (HMM) Approach	Uni-gram, Bi-gram	Two-stage HMM	Second stage ("General English") of the HMM provides smoothing	Required to train HMM and to construct General English stage of the model	Usage of HMM to LM for IR, incorporation of techniques such as blind feedback, feature dependent priors, query section weighting
Berger and Lafferty – Statistical Translation	Uni-gram	Translation models	Parameter smoothing based on document and background word probabilities	Large amounts of training data required to set translation model parameters	Application of translation models to LM for IR; view of query as a "distillation" or translation of a document
Jin and Hauptmann – Title Generation	Uni-gram	Translation model	N/A	Required to set translation model parameters	LM for IR in reverse; "distilling' the ideal title for a document from

					the full text using a translation model
Lafferty and Zhai – Risk Minimization Frameworks	Uni-gram	Bayesian decision framework	Markov model based query and document expansion using translation models	Required to set translation model parameters	Application of Bayesian decision theory to LM framework;
Lavrenko and Croft – Relevance Models	Uni-gram	Relevance model	Linear interpolation with parameter smoothing	Not required	Explicit modeling of relevance in an LM for IR

2.2.6 Cluster-Based Language Models

In the general sense, cluster-based language modeling approaches use document clustering to organize collections around topics. Each cluster is then assumed to be representative of a topic, and a language model can be created for the cluster (X Liu & Croft, 2004). Cluster-based language models are most commonly used in the context of Topic Detection and Tracking (Kraaij & Spitters, 2003), but have been incorporated into other IR frameworks (X Liu & Croft, 2004) as well. Although we choose to break out cluster-based language models as a separate section in this paper, one could just as easily group individual cluster-based modeling approaches with query-likelihood, document-likelihood or model comparison approaches, depending on the specific nature of the language model employed to describe the clusters. (We are not aware of any cluster-based language modeling approaches to date which have employed statistical translation methods.) In some sense, one might even say that cluster-based language modeling can trace its roots to Kalt (Kalt, 1996), which focused on statistical models for topics as opposed to individual documents.

One of the earliest cluster-based language modeling approaches was employed in the context of distributed information retrieval. Xu and Croft (Xu & Croft, 1999) propose that "the task of a distributed retrieval system is first to determine which topics are best for a query and then to direct the searching process to those collections containing the topics." They consider that the problem of determining the most suitable topics for a given query can be addressed using a generative model; that is, the best topics for a given query are those which have a language model that is most likely to generate the query. Collection selection for distributed IR is then performed by selecting the collections which contain the best topics.

Later work by Liu and Croft (X Liu & Croft, 2004) proposes the use of cluster-based language models in the context of an ad-hoc retrieval framework. Liu and Croft propose two language models for cluster-based retrieval: the first is used in the ranking and retrieving clusters and the other uses cluster language models to smooth individual document language models within the cluster. These cluster-based models are then integrated into both a query-likelihood and relevance model framework. Empirical results show that cluster-based retrieval can potentially be more effective than document based retrieval. It is specifically interesting to note the success of the cluster-based smoothing methods, and that "clusters generated by static clustering tend to produce better-quality cluster models for smoothing purposes than those generated by query-specific clustering (X Liu & Croft, 2004)." The authors believe that the static clustering methods outperform query-based clustering methods as query-based clusters may contain an inherent bias to a specific interpretation of a query term. Static clustering, performed

without query text, looks at all documents in the collection and generates clusters that provide better coverage for all aspects of a given topic.

In a more recent approach, Kurland (Kurland, Lee, & Domshlak, 2005) proposes that a document retrieved as a part of pseudo-relevance feedback may be considered as a "rendition" of the original query. Documents which are good renditions of the query may be considered to be pseudo-queries and considered to be a wholesale replacement for the original query itself. Kurland's approach proposes that once we have created an initial set of pseudo-queries, the process can be repeated so that in the next iteration the algorithm is searching for the documents that are the best rendition of the pseudo-queries. After the process is complete, the result should be a set of distilled pseudo-queries which are optimally informative of the user's information need. Kurland proposes three algorithms (the Viterbi Doc-Audition algorithm, the Doc-Audition algorithm, and the Cluster-Audition algorithm) to score documents as candidate renditions of the pseudo-query.

The motivation behind this approach is to increase the "aspect recall" of the system. The problem of "aspect recall" is described in (Buckley, 2004; Harman & Buckley, 2004) and can be categorized in one of four ways:

- The IR system emphasizes one aspect of a query, and misses other required terms in the query
- The IR system emphasizes one aspect of a query, but misses other aspects
- The IR system fails to properly combine aspects in a query when returning query results
- The IR system emphasizes an irrelevant aspect

Here, the term "aspect" is defined to mean a term feature that occurs in the query. For example, in the query "What incidents have there been of stolen or forged art?" we might consider "stolen art" and "forget art" to be two aspects of the query. The cluster-based language modeling approach suggested by Kurland considers language models for clusters of documents within the corpus, rather than single documents, and is based on the hypothesis that clusters are richer sources of information regarding corpus aspects than single documents alone.

Query drift is an issue that this particular algorithm is particularly prone to. "Query drift" (Mitra, Singhal, & Buckley, 1998) is a problem encountered by many automatic query expansion approaches; when expanding a query if non-relevant terms are included as a part of the expansion the resulting query may "drift" from the original information need. Kurland addresses this problem by using re-scoring techniques to periodically re-align the pseudo-queries with the original queries. The empirical results obtained from this methodology are promising when compared against a baseline relevance modeling approach.

Other cluster-based language modeling approaches include Zhang (Zhang, Ghahramani, & Yang, 2005), who models the generation of clusters using a Dirichlet process mixture model, where the base distribution can be treated as the prior of general English model and the precision parameter which controls the random generation process for creating new clusters. Like other cluster-based modeling approaches, this was effectively applied to the TDT task.

2.2.7 Smoothing Studies

Smoothing is an integral part of the language modeling paradigm and the performance of the smoothing component of a language model is essential to the overall performance of the model. Several classes of smoothing strategies have been proposed; the most common has been described as parameter smoothing (X. Liu & Croft, 2005), and uses linear interpolation to influence the roles that multiple knowledge sources play in smoothing out the probability distributions of a language model. Zhai and Lafferty (C. Zhai & Lafferty, 2004) studied three approaches to smoothing: Jelinek-Mercer smoothing, Dirichlet priors and absolute discounting, as well as the backoff versions of these methods. Five test collections were used to examine the effects of each of these smoothing mechanisms. For title queries (short queries), there was a clear ordering among the methods in terms of precision results; Dirichlet priors performed better than absolute discounting, which performed better than Jelinek-Mercer. Overall, Dirichlet priors had an average precision performance that was significantly better than the other two methods. For longer queries, both Jelinek-Mercer and Dirichlet priors have a better performance than absolute discounting. These results highlight the somewhat counter-intuitive result that there is more correlation between query length and query

A later study by Smucker and Allan (Smucker & Allan, 2007) investigate the causes behind the improved performance yielded by Dirichlet prior smoothing performs over Jelinek-Mercer smoothing for short queries, which are assumed to be most characteristic of a query that would be posed by a user. Both Dirichlet prior and Jelinek-Mercer linearly combine the maximum likelihood estimated (MLE) document model with the MLE model of the collection. Both are discounting smoothing methods that reduce the probability of the words seen in the document and reallocate the probability

mass to words not seen in the document. The only difference between the two smoothing methods that can be observed is that Dirichlet prior smoothes longer documents less and Jelinek-Mercer smoothes all documents to the same degree. Intuitively this makes sense since it stands to reason that the MLE model of a short document contains less observed information than the MLE model of a long document. Smucker and Allan found that Dirichlet prior's performance advantage comes more from its penalization of shorter documents rather than from its estimation process for long documents; in other words, Dirichlet prior is able to correctly reduce the scores of short documents. This research points to the importance of factors such as document length in the language modeling process.

2.2.8 Semantics and the Language Modeling Framework

Semantic smoothing (X. Liu & Croft, 2005) is one area in which there are significant opportunities for the integration of context-sensitive semantic knowledge into the language modeling framework. Translation models may be thought of as one form of semantic smoothing, since a translation model provides a mechanism for mapping based on document-query training sets. However, translation models do not provide any mechanism for sense disambiguation; as such, terms that appear with high likelihoods in two different senses will be mixed together. For example, the term "apple" may appear in conjunction with the word "computer" and the word "pie" both with high likelihoods.

Zhou et. al. (Zhou, Hu, Zhang, Lin, & Song, 2006) proposes a context-sensitive semantic smoothing as a part of the language modeling framework, which introduces the concept of a topic signature for a document or query and then uses these topic signatures

to train the translation model. Zhou defines a topic signature to be an order-free relationship between two concepts, where a concept represents a set of synonymous terms within a domain. For the biomedical domain, the Unified Medical Language System (UMLS) Metathesaurus (http://www.nlm.nih.gov/research/umls) is one source for concept definitions. Incorporation of semantic information by way of topic signatures alleviates the sense disambiguation (synonymy and polysemy) problems that are incurred with previous translation model approaches. This approach presents a different view of a document representation within the language modeling framework; a document is presented as a weighted set of topic signatures and concepts. An Expectation-Maximization (EM) (Dempster et al., 1977) based training method is used to train the context-sensitive model. The empirical results of this semantic approach showed significant improvements over the baseline on TREC Genomics 2004 (Hersch & al., 2004) and 2005 (Hersch & al., 2005) data. One drawback to this ontology-based approach, however, is that it requires the existence of domain-specific ontological resources which are not often available. This may make such an approach difficult to apply to a broad range of fields.

Another approach that may lend itself to incorporation within a semantic smoothing framework is Probabilistic Latent Semantic Analysis (PLSA) (Hofmann, 1999). The foundation of the PLSA approach is an underlying Aspect Model which proposes that we can define words and documents in terms of "aspects" which are associated with a latent class variable.

The Aspect Model is based on two assumptions:

- Words and documents are independent of one another (bag of words assumption)

- Documents and words are both conditioned on a latent aspect (z) which may be thought of as a concept

The Aspect Model has several intuitively appealing features. First, by conditioning words and documents on a latent variable, the zero-frequency problem is addressed. Secondly, a priori knowledge is not required about the concepts within the corpus for the algorithm to work effectively. And finally, the usage of probabilistic methods defines a generative model of the data which is better able to address common text processing issues such as synonymy and polysemy. PLSA uses the following approach which is depicted graphically in Figure 4:

- Select a document d with probability P(d)
- Pick a latent class z with probability P(z|d)
- Generate a word with probability P(w|z)
- Formulate the probability of observing a pair P(d,w) while the latent variable z is discarded

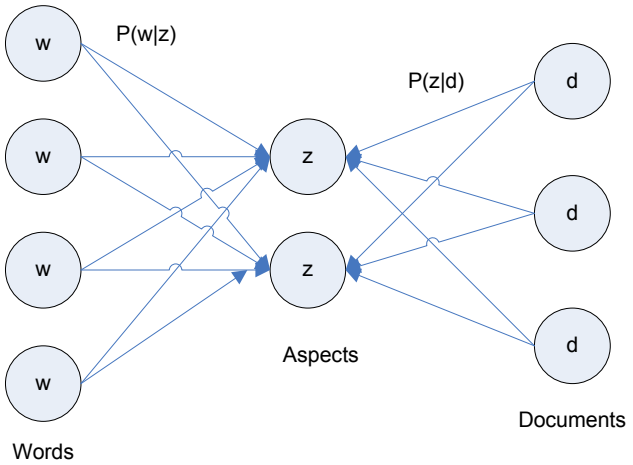

Figure 4: A Conceptual Model of Probabilistic Latent Semantic Analysis and the Aspect Model

PLSA is described mathematically by the following equation:

$$P(d,w) = \sum_{z} P(z)P(w \mid z)P(z \mid d) \qquad (22)$$

Intuitively, PLSA is based on maximizing the log-likelihood of the divergence between the empirical distribution of observed word-document pairs and the probabilistic model P(w,d). PLSA uses the EM (Dempster et al., 1977) algorithm to estimate the probabilities P(z), P(w|z) and P(d|z) for latent variable models. The E-step estimates the posterior probabilities of the latent variables, while the M-step is used for parameter estimation based on expected statistics. The resulting set of PLSA latent concepts (z values) are appealing because they are sense disambiguated without requiring availability of any external knowledge or ontological resources.

2.2.9 Language Modeling and XML Retrieval

Ogilvie and Callan (Ogilvie & Callan, 2003) believe that the information contained in the structure of an XML document can be used to improve document retrieval for structured knowledge sources. In order to leverage this information, they model document structure as an integral part of a document language model. Specifically, XML documents are modeled as trees where each node in the tree correspond directly with tags present in the document. For each document node in the tree, a language model can then be estimated. Language models for leaf nodes with no children are estimated directly from the text of the node. The language models for other nodes in the tree are estimated by taking a linear interpolation the language model of the text contained in that node alone, and the language models of all child nodes in the document's tree. The language models for each node are then smoothed via a interpolation with a collection language model. The collection model used for the

interpolation may be specific to the node type, which provides some measure of context sensitive smoothing, or the collection model may be one large model estimated from everything in the corpus, which gives a larger sample size.

In a subsequent work, Ogilvie and Callan (Ogilvie & Callan, 2006) focus on the problem of parameter estimation for the hierarchical language model for XML documents. Specifically, the parameters that are considered are the interpolation parameters which are used to combine the language models at a given node with the language models of the node's children. Unlike their previous work (Ogilvie & Callan, 2003), Ogilvie and Callan make a simplifying assumption; they do not recursively smooth the language models when they traverse the structure of the XML document. Instead, they linearly interpolate the parent's unsmoothed language model, each child's unsmoothed language model, the document's unsmoothed language model, and the collection language model. This simplification allows for a cleaner formulation of the parameter estimation problem which enables application of a modified version of the EM algorithm which the authors call Generalized Expectation Maximization (GEM). The proposed approach first trains the EM algorithm, but observes that using positive examples alone places the most weight on the document language model which results in very poor retrieval performance. To counter this effect, negative examples are included in the model; negative examples are non-relevant components that come from documents that contain relevant components. Conceptually, the GEM algorithm maximizes the probability that the language models of the positive samples will generate the query term while minimizing the likelihood that the language models of the negative samples will not generate the query term. The approaches presented in (Ogilvie & Callan, 2003, 2006)

are validated empirically on data from the INEX CO tasks, with performance that is mixed but shows promise for further development.

2.2.10 Language Modeling and Web Retrieval

The Divergence from Randomness (DFR) framework (Amati & Van Rijsbergen, 2002) has applied the language modeling paradigm to Web retrieval. The DFR framework has its origins in the early work on automatic indexing that observed that the distribution of informative content words in a document collection deviates from the distribution on non-specialty words. "Specialty words, like words belonging to a technical vocabulary, being informative, tend to appear more densely in 'elite' documents whereas non-specialty words, such as the words that are usually included a stop list, are randomly distributed over the collection (Amati & Van Rijsbergen, 2002)." Early work in automatic indexing by Harter (Harter, 1975) proposes that non-specialty words can be modeled by a Poisson distribution with some mean λ. In Harter's model, a "specialty word" can be systematically detected by measuring the extent to which its distribution differs from a Poisson distribution for the entire document collection. A secondary hypothesis of Harter's model is that a "specialty word" follows the Poisson distribution with mean μ when only a small set of 'elite' documents is considered, where μ is much larger than λ. In Harter' model, the set of 'elite' documents was defined as a hidden variable and the estimation of the parameter μ was problematic. In the DFR model, the set of elite documents is assumed to be all documents which contain the specialty term.

The DFR framework develops this 2-Poission model by proposing that the weight of a term in a document is a function of two probabilities, Prob1 and Prob2, which are related by the following equation:

$$w = -\log_2 \Pr ob_1^{(1-\Pr ob_2)} \qquad (23)$$

The distribution $Prob_1$ is derived using a methodology similar to that used by Harter, but supposes that words which bring little information are randomly distributed on the whole set of documents, rather than being distributed by the Poisson model. The DFR framework provides different basic probabilistic models ($Prob_1$) for randomness; among these are the binomial distribution, the Poisson distribution, Bose–Einstein statistics, the inverse document frequency model, and a mixed model using Poisson and inverse document frequency. The second probability, $Prob_2$, is the probability of occurrence of the term within a document that is a part of the 'elite' set of documents for the term. 1 – $Prob_2$ can be thought as "the risk of accepting a term as a good descriptor of the document when the document is compared with the elite set of the term (Amati & Van Rijsbergen, 2002)." The formula in equation 22 can be re-written as:

$$w = Inf_1 * Inf_2 \text{ where} \qquad (24)$$

$$Inf_1 = -\log_2 \Pr ob_1 \text{ and } \quad Inf_s = 1 - \Pr ob_2$$

Here, the terms Inf_1 and Inf_2 can be thought of as two informative content functions – the first related to the informative content of the term with respect to the collection and the second with respect to its set of elite documents.

In the context of the DFR framework, term frequency normalization can be thought of as analogous to smoothing in a pure language modeling framework. Dirichlet prior term frequency normalization is studied extensively in (B He & I Ounis, 2005; B.

He & I Ounis, 2005), and an automatic theoretically-driven tuning methodology for the Dirichlet Priors normalization is presented which models term frequency normalization as a function of the correlation between the normalized term frequency and the document length.

Empirically, the DFR framework was one of the best-performing runs of the TREC 2002 WEB Track (Craswell & Hawking, 2002), and showed promising results when a Bose-Einstein model of randomness was incorporated into the base framework described above. Later work has focused on the application of the DFR framework to large collections, and specifically to web-document finding tasks (He & Ounis, 2007). The Terrier Information Retrieval Platform (Ounis et al., 2006) is built on the foundations of the DFR framework and is an open-source retrieval engine that has been applied for both English and non-English (Macdonald, Lioma, & Ounis, 2007) Web retrieval tasks.

2.2.11 Challenges and Issues

Allan et. al. [59] presents a comprehensive discussion of near and longer term challenges in Information Retrieval to which language modeling may be applied. These challenge areas include "retrieval models, cross-lingual retrieval, Web-search, user modeling, filtering, Topic Detection and Tracking (TDT), classification, summarization, question answering, meta-search and distributed retrieval, and multimedia retrieval, information extraction and testbeds." Although this paper was written several years ago, the majority of the challenges and issues presented are still pertinent to state-of-the-art Information Retrieval systems today. It is beyond the scope of this paper to discuss each of the challenge areas in detail; instead we will focus on three areas that are of particular

interest to us: retrieval models, cross-language information retrieval and question answering.

Empirically, retrieval models that incorporate relatively simple language modeling techniques have produced promising results over the past decade. The majority of these approaches have used a unigram language model, although a few have explored approaches based on bi-gram and tri-gram models. This relatively simplistic view of the document as a "bag of words" is an opportunity for future research. In his survey of statistical language modeling techniques, Rosenfeld [13] says, "ironically, the most successful statistical language modeling techniques use very little knowledge of what language really is … only a handful of attempts have been made to date to incorporate linguistic structure, theories or knowledge." The incorporation of these types of formalized structures into a language modeling framework may help to increase the effectiveness of language modeling approaches. In addition, further improvements in the language modeling paradigm "are likely to require a broad range of techniques in addition to language modeling [59]." This is especially true if one considers the changing landscape of information resources; grass-roots innovations such as wikis, blogs, social bookmarking and even social networking applications such as MySpace and Facebook present unique challenges and opportunities for modeling language. Hierarchical models such as those proposed by Ogilvie [49] are promising when one considers retrieval applications that merge unstructured and semi-structured data; it may be likely that future approaches build on this foundation when considering language data from diverse sources.

A second area challenge area is cross-language information retrieval research. Allan et. al. [59] states that "though initially the Web was dominated by English speakers, now less than half of existing web pages are in English." If this statement was true in 2002, today the importance of cross-language information retrieval is even more evident. From a humanitarian standpoint, one may consider that we have a social responsibility to ensure that cross-language information search and retrieval techniques are not limited to those languages for which large amounts of data are available – such as English, Spanish, French, Arabic, and Chinese. In order to effectively extend language modeling techniques to languages for which less data is available, existing methods should look to develop techniques for which little or no training data is required. If such research is not undertaken, the distinction between those who have access to information and those who do not will be made along language boundaries, and the information gap will widen as time goes on.

A final challenge area for language modeling that is of particular interest for this work is Question Answering (QA). Recent developments in Question Answering research have started to address many of the topics that were discussed in Allan [59]. Recent TREC conferences [60-62] have considered reliable factoid QA, interactive QA, development of user and session models which require that a given question be cognizant of the QA dialogue that preceded it and novelty detection within a QA setting (eg. "Tell me something interesting about the topic that I have not seen yet.") In addition, recent QA approaches [63] have considered merging of structured and semi-structured information (such as information from knowledge sources such as Wikipedia and WordNet) to improve the QA reliability. The TREC Genomics Track [64] can also be

thought of as a QA exercise; in those experiments, researchers must return the answers to broad questions with some source document context. Despite the recent focus in this area, however, there are still many aspects of QA that require development or improvement. Within the QA application area, language modeling has largely been used to improve the first stage of processing which collects candidate documents and passages that are then passed on to an Information Extraction (IE) engine which determines the specific response words or phrases. A future goal for language modeling may be to provide for extensions that allow for integration with the second stage of QA – either by the incorporation of knowledge patterns or via statistical mechanisms that can be integrated into the IE or Natural Language Processing (NLP) techniques commonly used in downstream QA processing.

2.3 Question Answering Research and Trends

Research in natural language question answering has a history dating back to 1959 (Simmons, 1965). At that time a new paradigm was introduced into computer science spurred by Chomsky's research in linguistic analysis (Chomsky, 1956) which categorized language into syntactic, semantic and phonological components. For the first time, language could be categorized by a defined structure, and translation between languages was viewed as a set of transformational rules to morph the meaning of one language into another. The first generation of QA systems that came out of this new paradigm included social conversation systems, machine translation systems and programs that attempted to answer questions from English text (Simmons, 1965).

However, these QA systems were uniformly criticized for their limited semantic analysis capability (Simmons, 1965).

The next generation of QA systems included conversation machines, fact retrieval systems, mathematical word processors and natural language text processors (Simmons, 1970). The most famous of these early systems is BASEBALL (Green, A., Chomsky, & Laughery, 1961) which was a fact retrieval system that processed and returned data related to the game of baseball. BASEBALL used a pseudo-natural language interface for question input; a similar interface was used to insert facts (knowledge) into the system. Questions were processed by first breaking down sentences into candidate elements, and then evaluating the list of semantic features in a sentence against the internal knowledge source.

Winograd (Winograd, 1972) and Lehnert (Lehnert, 1977) present formal studies of the question answering paradigm, including knowledge representation models and knowledge extraction models. Lehnert presents the first comprehensive question taxonomy; she divides the full spectrum of possible questions into one of 13 categories and uses this taxonomy to develop the first conceptual theory of question answering. Because both Winograd and Lehnert propose solutions to a "generic" QA problem; their models are applicable to both open domain and domain-specific systems

The reinvigoration of research in QA in the mid-1990's may be attributable to two events: the creation of WordNet (G. A. Miller, 1995) and the establishment of a Question Answering Track at the 8th Text Retrieval Conference (TREC) (Voorhees, 1999). WordNet, (G. A. Miller, 1995) created at Princeton University, is the first instance of a comprehensive on-line, universally accessible lexical database of the English language.

"English nouns, verbs, adjectives, and adverbs are organized into sets of synonyms, each representing a lexicalized concept. Semantic relations link the synonym sets (G. A. Miller, 1995)." WordNet defines the vocabulary of the English language as a set of form-sense pairs, where the form is the string that uniquely identifies any given word and the sense is an element from a set of meanings. This structure allows WordNet to aid in the resolution of word synonymy (different words that share the same sense) and word polysemy (a single word that has more than one sense) – two classical problems that have faced QA systems from their early history (Simmons, 1970).

In 1999, the first TREC Question Answering Track was run (Voorhees, 1999). This event established comprehensive evaluation criteria for QA systems; the first TREC QA Track was run against data from the Ad-hoc Retrieval Track over a set of 198 questions. (An example of TREC QA questions and example documents from the AQUAINT corpus are presented in Appendix A).Human judges determined the correctness of each run submitted, and a final set of question-answer pairs were established which could then be used to assess the performance of subsequent systems. Twenty different organizations participated in the Track, and most followed a similar baseline strategy: (1) Questions were first classified into categories that would enable the prediction of the type of returned entity using heuristic methods. For example, "who" questions returned entities of type "Person"; (2) The systems then performed a "shallow parse" of the documents returned to detect entities of the same type as the expected answer; (3) In the event that no appropriate entity was found, the systems defaulted to passage retrieval based on established IR methods (Voorhees, 1999).

Over the past decade, a number of groups have tackled the formidable problem of TREC open domain QA, each with their own unique perspective and focus. Moldovan and Harabagiu (Moldovan et al., 2002) emphasize the importance of lexical chains in the question answering process. "Lexical chains improve the performance of question answering systems in two ways: (1) increase document retrieval recall and (2) improve the answer extraction by providing the much needed world knowledge axioms that link question keywords with answer key concepts (Moldovan et al., 2002)." Conceptually, a lexical chain is a sequence of semantically related words that tie two concepts together. Lexical chains can be used to re-write the original query so that the terms present in the query are more easily relatable to the text fragment that contains the answer. For example (S. Harabagiu, Moldovan, Rus, & Morarescu, 2001), given the question "How many dogs pull a sled in the Iditarod?" the word sled can be replaced with the word "harness" by mining WordNet. This allows the question to match the text fragment "Race rules require mushers to arrive in Nome with at least five dogs in harness."

Moldovan and Harabagiu have developed several QA systems that have performed well in the TREC QA experiments. They submitted the LASSO system in TREC 9 (Moldovan & Harabagiu, 2000) and the Falcon system in TREC 10 (S. Harabagiu et al., 2001). Subsequently, they commenced research on the PowerAnswer and PowerAnswer-2 systems which have had exceptional results in recent TRECs (S. Harabagiu et al., 2005). At TREC 2005, PowerAnswer-2 had the best performance for both factoid and list questions, and ranked third in performance for the "Other" component (Voorhees, 2005a), which is designed to evaluate how well the QA engine can locate information nuggets not previously known. The PowerAnswer-2 system is

comprised of a Question Processing Module, an Answer Processing Module, and World Knowledge Axioms that are fed into a Logic Prover module called COGEX (S. Harabagiu et al., 2005), which uses semantic information and abductive logic to score answers. Details of the operation of COGEX beyond the scope of this paper, but can be found in (Moldovan, Clark, & Harabagiu, 2003).

Although less overtly successful (Humphreys, Gaizauskas, Hepple, & Sanderson, 1999; Scott & Gaizauskas, 2000), the QA systems developed by the University of Sheffield during recent TREC experiments are valuable for their exploratory nature. For example, the fundamental coupling of IR and QA is questioned by the group at Sheffield. "Experiments suggest that when using the top 20 relevant passages, answer bearing passages are found only 66% of the time (Gaizauskas et al., 2003)." Thus, they argue, for those systems working with the top 20 retrieved passages, an upper bound of 66% was placed on answer extraction even before any processing took place. They experiment with using a Boolean IR system called MadCow (Gaizauskas et al., 2003) to increase the effectiveness of the IR component of the overall QA system. Unlike probabilistic IR systems such as Okapi (S. E. Robertson, Walker, & Beaulieu, 2000), a Boolean IR system allows the user to input the query as a sequence of terms linked together with Boolean operators (Gaizauskas et al., 2003). The IR system returns documents (or sentences or passages) which exactly match the Boolean query specified. A Boolean IR system gives the user more control over the documents returned, but requires more effort during the query formulation process. Additional experiments performed by the University of Sheffield group explored various methods of corpus pre-processing, answering "definition" questions, and improving part-of-speech tagging. All of these

efforts were started in 2003 and continued through 2004, 2005 and 2006 (Gaizauskas et al., 2005; Gaizauskas, Greenwood, Hepple, Roberts, & Saggion, 2004; Gaizauskas et al., 2003; M. A. Greenwood, Stevenson, & Gaizauskas, 2006), though the current set of results have not yet proved conclusive.

The group from the Tilburg University (Buchholz, 2001; Buchholz & Daelemans, 2002) propose a Question Answering System based on the usage of the World Wide Web as an alternate knowledge source. Their Shallow Parsing for Question Answering (Shapaqa) component uses a complex set of grammatical rules to increase the effectiveness of their part-of-speech tagger. The results of the system against the TREC ACQUAINT corpus are compared against results derived using the World Wide Web as a knowledge base. While initial results of the system were mixed, the experiments did support the assertion that facts which occur in the corpus more frequently are more reliable than those occurring less frequently.

Other approaches to QA have made heavy use of IE techniques. A full discussion of IE techniques and approaches is beyond the scope of this thesis, and the reader is encouraged to review (Gaizauskas, 1998) for a more comprehensive view. Molla-Aliod (Molla-Aliod, Berri, & Hess, 1998) proposes a method of Answer Extraction which is rooted in IE, and extracts answers to common Unix question from 30 Unix manual pages. The proposed solution is not as ambitious as a technique that would address the full-blown Question Answering problem since the question domain and the corpus from which the answers will be mined are both part of a narrow domain. However, the system successfully demonstrates the application of IE to QA.

Srihari (Srihari & Li, 2000) develops a more sophisticated IE-based QA system called Textract centered around "a hierarchical, three-level architecture which is domain-independent throughout." The three levels of the system address different levels of entity extraction. The first level is called Named Entity and deals with the identification of simple, individual entities. The second level, Correlated Entity, "is concerned with extracting pre-defined multiple relationships between entities." And the final level is to extract General Events which deal with subject-verb-object structures that can be used to identify relationships between entities and actions. Using all three levels of IE, Textract proposes the formulation of a framework that can be extended to handle a breadth of domain independent QA scenarios.

Other interesting approaches QA have included designing a system patterned after the Model-View-Controller design (Dalmas & Webber, 2004), usage of information fusion techniques (Dalmas & Webber, 2005), and the application of the Mutual Information statistic to estimate question categories (Mann, 2001). In addition, researchers at MITRE have explored areas of the QA space adjacent to the problems presented by the TREC experiments: specifically, they have experimented with reading comprehension systems (Hirschman, Light, Breck, & Burger, 1999) and human-computer dialogue systems (Doran, Aberdeen, Damianos, & Hirschman, 2001); this work can be seen as a direct descendant of the conversation and word problem solving systems of the 1960's.

To date, the efforts in domain-specific question answering have been limited. Zweigenbaum (Zweigenbaum, 2003) presents survey of question answering experiments that had been run prior to 2003 in the biomedical domain. He concludes that "no specific

evaluation of medical QA has yet been performed (Zweigenbaum, 2003)." In 2006, the TREC Genomics Track introduced a QA component (Hersh, Cohen, Roberts, & Rekapalli, 2006) to extend the ad-hoc retrieval tasks that had been run in the preceding TREC Genomics Tracks (Hersh, Yang, Bhupatiraju, Cohen, & Roberts, 2005). The main component of this track required the retrieval of passages that directly answered questions. Up to 1000 passages could be submitted for every question, and the MAP (Mean Average Precision) measure was used to evaluate results. The responses to this Track were enhanced by the use of several domain-specific knowledge sources such as Unified Medical Language System (http://www.nlm.nih.gov/research/umls/) and the Gene Ontology (http://www.geneontology.org/). A key architectural element in several competing systems was a concept recognition component (Caporaso et al., 2006; Trieschnigg, Kraaij, & Schuemie, 2006), which matched known concepts with terms found in questions and documents against a knowledge source such as a thesaurus. "The use of a thesaurus allows the identification of multi-word terms and the mapping of synonyms to one concept (Trieschnigg et al., 2006)."

Despite variation in evaluation measurements, the results of the 2006 TREC Genomics Track (Hersh et al., 2006) on the whole surpass open-domain TREC QA results (E. M. Voorhees & D. K. Harman, 2005) in terms of accuracy and precision. However, the Question Answering component of the TREC Genomics Track was ended in 2007 (Hersh et al., 2006) which may indicate a decrease in interest in QA for the biomedical domain. While the TREC Legal Track (http://trec-legal.umiacs.umd.edu/) also focuses on domain-specific retrieval tasks, no specific QA tasks are planned for that area.

2.4 Candidate Document Selection

Two recent forums that highlight research in Candidate Document Selection are the Information Retrieval for Question Answering (IR4QA) Workshops (Gaizauskas, Hepple, & Greenwood, 2004) that occurred in 2004 as a part of the ACM Special Interest Group in Information Retrieval (SIGIR) and in 2008 as a part of the International Conference in Computational Linguistics (COLING) (M. Greenwood, 2008). Both workshops consider a high-level Question Answering architecture that is similar to the two-stage approach that we use as the foundation for this work. They propose that limited attention has been paid to the first stage (the IR stage) of the QA architecture, and seek to draw attention to work explicitly being done in IR that specifically addresses the challenges raised by QA. Some of the broader issues that are discussed in the workshop, which might be considered fundamental challenges of Candidate Document selection, include:

- How can a question be pre-processed to generate a query that is then sent on to an IR system?
- How can the document collection be most effectively indexed to include significant information that is pertinent to QA?
- How should the similarity metrics that are commonly used to determine relevancy be modified for QA?
- Can passages or even sentences be returned in response to a query?

In the remainder of this section, we describe recent approaches to Candidate Document Selection in the context of these questions. We propose that there are three broad

categories of approaches to Candidate Document Selection: 1) question to query pre-processing approaches, 2) IR4QA similarity modeling approaches, and 3) passage and sentence retrieval approaches. Each of these approaches is discussed in detail in the following sub-sections.

2.4.1 Question to Query Pre-processing Approaches

Bilotti (Bilotti, Katz, & Lin, 2004) examines different approaches for handling morphological variation in question terms, specifically in a Boolean IR setting. In particular, they compare the use of Porter stemming at indexation time to an approach using morphological expansion at query time. In the latter, a term is replaced with an OR-ed set of its morphological variants, where the morphological variants may be given an equal or lesser weight than the original term. Their results indicate that stemming leads to lower recall, while expansion leads to higher recall. The approach that gives different terms differing weights performs above that where all morphological variants are weighted equally. It should be noted here that in this paper they make the important assertion that recall is of primary importance in IR4QA systems. We make a similar assertion when we propose our own approach for Candidate Document Selection later in this text.

Another query expansion approach is presented by Negri (Negri, 2004). Negri proposes an approach that uses relevance feedback to help sense-disambiguate words in the query. Instead of solving the difficult problem of word sense disambiguation on short questions, this approach shifts the problem o the easier one of performing word sense disambiguation within the context of feedback documents. So, for example, the top N

documents returned from an initial query are used as context which is then used to sense disambiguate the query terms. Terms which are semantically related (using WordNet (G. A. Miller, 1995) as an external source of semantic relationships) to the appropriate sense of each disambiguated word are then used to expand the query. Presented results for this technique show some improvement in recall for the top 50 documents.

Dercynski (Derczynski, Wang, Gaizauskas, & Greenwood, 2008) proposes a novel approach in which answer texts from previous TREC QA evaluations are paired with queries and analyzed in an attempt to identify performance-enhancing words. These words are then used to evaluate the performance of a query expansion method. Data driven extension words were found to help in over 70% of difficult questions, specifically by improving recall in the expanded query set. Simple blind relevance feedback (RF) was correctly predicted as unlikely to help overall performance, and an possible explanation is provided for its low value in IR for QA. Dercynski considers that relationships exist between query words and performance-enhancing words might be mined and used to enhance simple blind relevance-feedback approaches.

Saggion (Saggion, Gaizauskas, Hepple, Roberts, & Greenwood, 2004) takes a different route to query formulation by focusing on a Boolean IR environment. They systematically consider a number of strategies for the construction of boolean queries from natural language question and the ranking of results returned by these queries. These include expanding the query by disjoining (OR-ing) WordNet synonyms or morphological variants of query terms, and "relaxing" the query by progressively discarding query terms with low inverse document frequencies. Saggion also experiments with different methods of handling proper names and quoted strings, and

experimenting with different sized windows within which terms are constrained to match. The results of these experiments show that while the Boolean IR strategies considered do not match the results for state-of-the-art retrieval systems at 200 documents retrieved. However, for results at lower ranks they do offer reasonable coverage for small passages, which may be helpful to some QA systems depending on the QA system's overall ability to extract answers from retrieved passages.

A lexico-semantic approach to query expansion for passage retrieval is proposed by van der Plas (van der Plas & Tiedemann, 2008). They use primarily corpus-based methods to construct the their lexical resources and evaluate their techniques on the Dutch Cross-Language Evaluation Forum (CLEF) QA Track (Peters, 2006, 2007, 2008). This corpus-based approach, however, does not show significant improvements over the baseline.

2.4.2 IR4QA Similarity Modeling

Stoyanchev (Stoyanchev, Song, & Lahti, 2008) proposes an approach to matching queries with documents that considers phrases which are automatically identified from previous questions as exact match constituents to search queries. This approach uses past questions in the QA series as a knowledge base that can be used to mine similarity relationships for future queries. The results show an improvement over baseline on several document and sentence retrieval measures on the TREC Web dataset (E. M. Voorhees & D. Harman, 2005). A 20% relative improvement is achieved in MRR for sentence extraction on the TREC Web dataset when using automatically generated phrases and a further 9.5% relative improvement when using manually annotated phrases. These results, however, should be tempered by the fact that queries run on the

ACQUAINT dataset (the testbed for TREC QA) showed no effect on IR performance when using exact phrase-based matching.

In a different vein, Monz (Monz, 2004) proposes a novel proximity-based approach to document retrieval for QA called minimal span weighting. The approach uses a parameterized combination of three factors in computing query-document similarity: 1) a conventional global query-document similarity score, 2) the minimum size of the document text window that covers all terms common to the query and the document, and 3) the ratio of matching terms in a query-document pair to the overall query length. Experimental results show that approach leads to significant improvement when compared with state-of-the-art document retrieval approaches.

2.4.3 Passage and Sentence Retrieval Approaches

Cui (Cui, Kan, Chua, & Xiao, 2004) describes a set of detailed experiments on ranking retrieved sentences for use in responding to definitional questions. The utility of ranking for various types of information is considered, including words co-occurring with the definition target in external resources such as WordNet or Wikipedia. Hard (proper noun) and soft (descriptive) definitional patterns are acquired using both supervised and unsupervised techniques. Results show that external resources are helpful, and that machine-learned definitional patterns outperform manually (heuristically) constructed ones. In general, the results show that soft patterns outperform hard patterns, and that the highest performance gains are achieved when supervised techniques are used to supplement unsupervised ones.

Usunier (Usunier, Amini, & Gallinari, 2004) also applies a machine learning approach to the task of passage retrieval. Specifically, they focus on the problem of re-

ranking the passages returned by a conventional retrieval engine. A series of baseline ranking functions are applied to generate a number of scores for each passage, and then the final ranking of the passages is based on a weighted combination of these scores. The weights applied to create the final combination are learned using a boosting algorithm, and the QA set is split to provide the training and test data. The results show significant improvement over the initial ranking produced by the IR engine.

Tiedemann (Tiedemann & Mur, 2008) considers the question: what is a passage? Past approaches to passage retrieval have typically focused on fixed-size windows. Tiedemann investigates several ways of dividing documents into passages. In particular, the study considers semantically motivated approaches using co-reference chains and discourse clues compared with simple fixed-size window-based techniques. The results of the experiments show the somewhat surprising result that the simple techniques using fixed sized windows outperform the semantically motivated approaches. This indicates that uniformity in size seems to be more important for passage level retrieval than semantic coherence. Related research performed by (Khalid & Verberne, 2008) shows the effectiveness of sliding fixed-size windows compared with disjoint windows.

White and Sutcliffe (White & Sutcliffe, 2004) present the results of a manual investigation in which 50 TREC factoid questions are compared with the sentences in the ACQUAINT corpus which contain their answers. The goal of the study was to identify the range of morphological and semantic transformations linking the query terms to related sentence terms, and the relative importance of each term. The results of the study are important to consider when designing a sentence retrieval approach; specifically, their results show the importance of hypernyms and word co-occurrence mappings. In terms

of the research presented in this paper, these qualitative results support our use of statistical methods that are based on word co-ocurrence relationships (PLSA). Furthermore, the study also suggests that a high proportion of the TREC questions may have a supported answer contained within a single sentence. (This was true for 90% of the 50 questions considered.)

We have previously mentioned the work of Murdock in the area of sentence retrieval (Murdock & Croft, 2004) when discussing statistical translation models in Section 2.2 of this text. Although we do not re-state that discussion here, it should be noted that Murdock's work falls into this category of approaches.

2.5 Answer Validation

In recent years, Answer Validation has become a topic of significant interest within the Question Answering community. It is the goal of open domain QA systems to search for answers to a natural language question either on the Web or in a local document collection. The majority of the QA techniques discussed in Section 2.3 propose methods for extracting the text fragments containing candidate answers from the document collection; in other words, they focus on the Answer Extraction process discussed in our QA architecture. The output of the Answer Extraction process can be considered to be a set of text fragments and their associated scores or rankings; in this text we refer to these text fragments as candidate answers.

In this context, automatic techniques for Answer Validation are of great interest for the development of open domain QA systems. The availability of a completely automatic evaluation procedure for evaluating a candidate answer makes it feasible for QA systems

to generate and test approaches. For example, "until a given answer is automatically proved to be correct for a question, the system will carry out different refinements of its searching criteria checking the relevance of new candidate answers (B Magnini, M Negri, R Prevete, & H Tanev, 2002a)." In addition, given that most of the QA systems rely on complex architectures and the evaluation of their performances requires a huge amount of work, the automatic assessment of the relevance of an answer with respect to a given question has the potential to speed up both algorithm refinement and testing.

Recent approaches to are showcased in the Answer Validation Exercise (AVE) which is conducted as a part of the Cross-Language Evaluation Forum (CLEF) Question Answering Track (Peters, 2006, 2007, 2008). The CLEF AVE task considers a set of triples in each of the languages in which CLEF is run: Bulgarian, German, English, Spanish, Finnish, French, Italian, Dutch and Portuguese. These triples consist of a question, a candidate answer and a passage of supporting text. The goal of the exercise is to determine whether the question is correctly answered on the basis of the supporting text. In effect, systems must provide a "YES" or "NO" Answer Validation judgment decision for each triple. The results of the decisions are then judged by human assessors in each language. The baseline is a system that always answers "YES" to every question-answer-text triple. Typically, Answer Validation systems that enter the CLEF experiments run in multiple languages; systems are compared based on how well they perform for each language individually and overall across all the languages for which they choose to participate.

A survey of the literature, including the CLEF Working Notes papers, shows that automated approaches to Answer Validation can be broadly separated into two

categories. The first category can be broadly generalized as machine-learning and lexico-semantic approaches to the problem of textual entailment. Textual entailment can be thought of as the problem of hypothesis proving. In other words, textual entailment seeks to understand how well a hypothesis is entailed by a segment of supporting text. For example, given the question "What is the capital of the USA?" the problem of validating the answer "Washington" is equivalent in the textual entailment paradigm to estimating the truthfulness of the hypothesis "The capital of the USA is Washington DC" given a segment of supporting text. Textual entailment methods first generate a hypothesis (often using lexical or semantic methods), and then seek to understand the validity of that statement given a segment of supporting text. The majority of the CLEF entries use this type of approach. The second broad category of Answer Validation strategies are statistical methods. To date, only a few groups have effectively applied statistical methods to the problem of Answer Validation, although statistical methods have proven to be effective and efficient for this task.

The remainder of this section describes the work that has been done in each of these categories of Answer Validation strategies in detail.

2.5.1 Entailment Methods for Answer Validation

A comprehensive look at the application of textual entailment to Question Answering is presented by Harabagiu (S Harabagiu & Hickl, 2006). Specifically, Harabagiu explores three methods by which textual entailment may be incorporated into QA. In the first method, each of a the candidate answers in a ranked list that do not meet the minimum conditions for textual entailment are removed from consideration. Then, the

remaining candidate answers are then re-ranked based on the entailment confidence assigned by the textual entailment system. Finally, the system outputs a new set of ranked answers which do not contain any answers that are not entailed by the user's question and the associated supporting text contained within the document collection.

The second method focuses on using textual entailment to limit the number of passages considered during the Answer Extraction process. (Harabagiu calls Answer Processing what we have called Answer Extraction.) In effect, Harabagiu uses entailment as a secondary Candidate Document Selection stage, and proposes that since Answer Extraction is often a resource intensive process, textual entailment can be used to limit the number of passages that are considered for Answer Extraction. Lists of passages retrieved by a the Candidate Document Selection module are re-ranked based on textual entailment information; in some cases, the contents of passages may be altered to more specifically represent entailment information. Once ranking is complete, Answer Extraction takes place only on the set of entailed passages that the system considers likely to contain a correct answer to the user's question.

In the third method, Harabagiu builds on previous work (S. Harabagiu et al., 2005) in exploring techniques that can be used to automatically generate well-formed natural language questions from the text of paragraphs retrieved by a Candidate Document Selection module. In the third method of applying textual entailment to QA, sets of automatically-generated questions are created using a stand-alone AutoQUAB textual entailment module, which assembles question-answer pairs from the top-ranked passages returned in response to a question. Harabagiu expects that if a question q_0 logically entails another question q_1, then some subset of the answers entailed by q_1 should also be

interpreted as valid answers to q_0. By establishing textual entailment relationships between a user's question and the automatically-generated questions derived from passages identified by the QA system for that question, Harabagiu proposes that a set of answer passages that contain correct answers to the original question can be identified. If no automatically-generated questions were found to be entailed by the original question, passages are ranked according to their entailment confidence (following method 2) and sent to Answer Extraction built for further processing and validation.

All of these approaches to entailment is reflected in the COGEX entry to the CLEF AVE in 2006 (Tatu, Iles, & Moldovan, 2006), which was one of the most successful Answer Validation approaches across languages that year, uniformly outperforming the baseline.

Also in the 2006 CLEF AVE, Rodrigo (Rodrigo, Peñas, & Verdejo, 2006) proposes a textual entailment system that only uses information about entities (numeric expressions, temporal expressions and named entities) in order to study the importance of entities in answer validation. The motivation for this experiment came from the study of the QA task in CLEF 2005, where 75% of the questions in Spanish were factoids that had a strong focus on entity relationships. For example, the responses to factoid questions contain typically contain entities (eg. person names, locations, numbers, dates). For this reason, Rodrigo considers that the study of entities is important for the textual entailment task, as applied to Answer Validation. In their experiments, Rodrigo uses the hypothesis that in the recognition of textual entailment all the elements in the hypothesis must be entailed by elements of the text. The proposed system obtained better results than the baseline system (which always responds "YES"), achieving an increase of 0.8 in the F-

measure. We note the importance of this result, as the methods which are described for creating an automatically-generated Question Context Model (QCM) later in this text focus on entities as well.

A number of the CLEF AVE entries use machine learning techniques for the purpose of generating textual entailment relationships and scores. In particular, Kozareva (Kozareva, Vázquez, & Montoyo, 2006) uses the already developed machine learning based textual entailment system called MLEnt for Answer Validation. The MLEnt system consists of word overlap and semantic similarity modules. For the AVE competition, only the word overlap module was utilized as the similarity, and it was impossible to adapt it to the different language pairs. MLEnt was combined with a simple voting technique to a newly developed module for acquiring semantic similarity information through Latent Semantic Indexing (LSI) (Deerwester, Dumais, Furnas, Landauer, & Harshman, 1990).

In order to apply LSI to the AVE task, a corpus was required which serves as a basis for the construction of the conceptual matrix. The Answer Validation Text-Hypothesis data provided by the AVE organizers was used for this corpus. The conceptual matrix is constructed only with the text-phrases of the AVE corpus. A study was conducted which found that the most effect LSI matrices were created when using the Text parts of the Text-Hypothesis data, as opposed to the full corpus provided. For each one of the languages – English, Spanish, Italian, German, Dutch, Portuguese and French, different conceptual matrices were constructed using the sentences of the text from the AVE corpora.

From the generated conceptual matrix one can establish the similarities between terms, phrases or documents. Used in the manner described, LSI extracts (in effect) the

similarity relations between the Text-Hypothesis phrases and the results are a list of Text-Hypothesis phrases ordered by their similarity score to the question-answer pair being evaluated. Under the concept of LSI, an AVE pair is correct when the similarity value is close to 1, and less similar when the similarity value is close to 0. Performance of this system uniformly outperformed the baseline across all the languages for which it submitted results. The success of LSI is an important result for our research since we employ a Probabilistic Latent Semantic Analysis (PLSA) indexing to the Answer Validation problem.

Another machine learning approach to Answer Validation is proposed by Wang and Neumann (Wang & Neumann, 2007, 2008). Their system contains a main approach with two backup strategies. The main approach extracts parts of the dependency structures to form a new representation, named Tree Skeleton, as the feature space and then applies Subsequence Kernels to perform Machine Learning. The backup strategies deal with the Text-Hypothesis pairs which cannot be solved by the main approach. One backup strategy is called Triple Matcher, as it calculates the overlapping ratio on top of the dependency structures in a triple representation; the other is simply a bag-of-words method, which calculates the overlapping ratio of words in the Text and the Hypothesis.

Another machine-learning approach to Answer Validation is presented by Glockner (Glöckner, 2006, 2007, 2008). His system, called RAVE (Real-time Answer Validation Engine) is a logic-based answer validation and selection engine which is designed for application in real-time question answering. RAVE uses a deep linguistic analysis processing chain and a simplified validation model where the logic prover only checks to determine if the support passage contains a correct answer at all. Across the three years

of CLEF participation, RAVE moves from a full-up logic-based answer validation system to logical validation of supporting snippets. This permits RAVE to avoid any parsing of answers (the system only needs a parse of the question and pre-computed snippet analyses) so that very fast validation and selection times can be achieved. Machine learning is used for assigning local validation scores using both logic-based and shallow features. One of the key features of RAVE is its innovative aggregation model, which is robust against duplicated information in the support passages. In this model, the effect of aggregation is controlled by the lexical diversity of the support passages for a given answer. If the support passages have no terms in common, then the aggregation has maximal effect and the passages are treated as providing independent evidence. Repetition of a support passage, by contrast, has no effect on the results of aggregation at all. RAVE achieved an F-score of 0.39 and a selection rate of 0.61 across all languages for which it competed, which outperformed the baseline.

A final method of applying machine learning to the Answer Validation task is explored over two years of the CLEF AVE by Tellez-Vallero (Téllez-Valero, Juárez-González, Montes-y-Gómez, & Villaseñor-Pineda, 2008; Téllez-Valero, Montes-y-Gómez, & Villaseñor-Pineda, 2007). Their method is based on a supervised learning approach for recognizing the textual entailment. It mainly uses a set of attributes that capture some simple relations among the question, the answer and the given supported text. In particular, it considers some novel attributes that characterize the compatibility between question and answer types; the redundancy of answers across streams of candidate answer providers (eg. if multiple methods of Answer Extraction are employed), and the overlap between the question-answer pair and the core fragment of the support

text. The objective of this Answer Validation method was to discriminate correct from incorrect answers as well as to provide a capacity to combine the answers from several Question Answering systems. The evaluation results of the proposed method achieved a 0.39 F-measure in the detection of correct answers across all the languages in which it participated, which outperformed the baseline result of the AVE 2008.

2.5.2 Statistical Methods for Answer Validation

Early Answer Validation techniques had the goal of filtering out improper candidates by checking how adequate a candidate answer is with respect to a given question. These approaches relied on discovering semantic relations between the question and the answer. For example, Harabagiu (Moldovan & Harabagiu, 2000) considers answer validation to be an inference process, where an answer is valid with respect to a question if an explanation for it, based on background knowledge from an external semantic resource, can be found.

However, the usage of semantic techniques on open domain QA tasks is problematic. Acquisition and assimilation of authoritative semantic resources is expensive both in terms of the linguistic resources that are required and in terms of computational complexity. These issues motivate the seminal work in modern automated Answer Validation using statistical methods; Magnini (B Magnini, M. Negri, R. Prevete, & H. Tanev, 2002b) presents an approach that uses redundant information sources on the Web for the purpose of Answer Validation. Magnini proposes that the number of Web documents in which the question and the answer co-occurred can serve as an indicator of answer validity. The approach is based on the intuition that the semantic connections

between an answer and a question can be estimated by exploiting the redundancy of Web information. “The hypothesis is that the number of documents that can be retrieved from the Web in which the question and the answer co-occur can be considered a significant clue of the validity of the answer (Magnini, Negri et al., 2002a).” Documents are searched in the Web by means of validation patterns, which are derived from a linguistic processing of the question and the answer. The advantages of this approach are its simplicity and its efficiency.

In (Magnini, Negri, Pervete, & Tanev, 2002), Magnini furthers his original work by the nature of validation patterns. The intuition underlying this approach is that, given a question-answer pair, it is possible to formulate a set of validation statements whose truthfulness is equivalent to the degree of relevance of the answer with respect to the question. Magnini reformulates the answer validation task as a problem of statement reliability. However, there are two issues that Magnini addresses in order to make the approach effective. First, a validation statement needs to be converted to a richer representation format to capture the knowledge that may connect a question to an answer; this representation is what Magnini terms a validation pattern. Second, Magnini proposes that checking the reliability of a validation pattern can be accomplished by relying on statistical counts of terms in redundant information sources.

Magnini (Magnini, Negri, Pervete et al., 2002) proposes the following four-step algorithm:

1. Compute the set of representative keywords from the question and the answer. This step is carried out using linguistic techniques, such as answer type

identification (from the question) and named entities recognition (from the answer);

2. From the extracted keywords compute the validation pattern for the question-answer pair. This step is also performed using linguistic techniques.
3. Submit the validation pattern to a publicly available search engine (Magnini used Google)
4. Estimate an Answer Relevance Score considering the results returned by the search engine.

The results of this algorithm yield considerable increases in precision, recall and success rate (accuracy) over the baseline results. The success of Magnini's work (Magnini, Negri, Pervete et al., 2002; Magnini, Negri et al., 2002a) shows the viability of the application of statistical methods to Answer Validation.

A related approach to Answer Validation is proposed by Li (F. Li, Zhang, & Zhu, 2008). Li also proposes a novel approach based on information retrieval on the Web. However, Li reformulates the Answer Validation problem as a pure distance calculation from an answer candidate to a question. The hypothesis is that, among all candidates, the correct answer has the smallest distance from question. Li employs the conditional normalized minimum distance metric, which is based on Kolmogorov Complexity theory (M. Li & Vitanyi, 1997), for this task. This distance measures the relevance between question and answer candidates conditioned on a surface pattern set. For a distance calculation, Li first extracts keywords from the question (called a set of question focus terms), and then constructs a hierarchical pattern set of related terms based on documents returned from a publicly available search engine. Li makes used of both the "Google"

and “Altavista” search engines. The information distance calculation is simplified as Kolmogrov complexity can be approximated through term frequency counts. The results of their experiments show that the results of this method are stable and robust, and do not depend on the specific question, corpus or search engine.

It should be noted here that we see an opportunity for the application of statistical language modeling techniques to Answer Validation by noting the general success of statistical methods applied to this problem. Furthermore, the lack of work done specifically in applying statistical methods to Answer Validation makes this an excellent candidate area for further research.

3. THEORY AND MODELS

We present the theoretical foundation of our work in four sections. First, we describe our preliminary work in Question Answering in the context of our entry to the 2007 TREC Question Answering Track. We first discuss the nature of the TREC QA task, with specific examples of questions, documents and correctness. Then, we describe our TREC entry. The issues we encountered specifically relating to the incorporation of semantic information into the design of a QA system serve as the motivation for our theoretical approach. Next, we describe our early work that compares the effectiveness of language modeling with other IR approaches for QA using a simple evaluation methodology. Finally, we describe our theoretical model, the Aspect-Based Relevance Language Model, and the Question Context Model that can be derived from it.

3.1. TREC Question Answering Track

As discussed in Chapter 2, the goal of the TREC QA track is to foster research on systems that retrieve short, concise answers to a question rather than lists of documents. The TREC QA task focuses on open-domain data rather than on a particular specialty area. This ensures that domain-dependent resources are not utilized to enhance the performance of the QA systems being evaluated, and ensures transferability of QA designs and models to the widest variety of application areas. In this section we describe the TREC Question Answering testbed in detail. This explanation will not only be useful as background for our preliminary efforts in Question Answering, but will also serve as background material for our later experiments in Candidate Document Selection and

Answer Validation. The TREC QA testbed provides an objective platform on which we can evaluate our algorithms against state of the art approaches. Without the existence of such an evaluation platform, it would be impossible to quantify the merits of our algorithms against other approaches.

3.1.1 TREC Question Answering Test Questions

In 2007, the TREC QA challenge required competing systems to correctly answer a set of 75 question series. Each question series asked for information about a particular target or topic. As in TREC 2006, the topics (or targets) included people, organizations, events and entities. At a high level, a question series may be considered to be an abstraction of a user session with a QA system. Some questions depend on knowing the correct answers to previous questions in the same series. There are, however, no cross-references between question series. For example, the correct answer to a question in a later question series cannot depend on the answer provided in an earlier question series. A QA system that participates in the challenge can consider each question series independently. The TREC QA Track assumes that the user of the system is an "average" adult reader of American newspapers.

Each question series consists of some factoid and some list questions, and ended with exactly one "Other" question. A factoid question requires the QA system to produce a single response, along with a reference to a document in the dataset that supports the response. A list question requires the QA system to produce one or more responses in an unordered list, each with a reference to a document in the dataset that supports the response. “Other” questions are somewhat less intuitive; a correct answer in response to

an "Other" question is any piece of “interesting” (E. M. Voorhees & D. K. Harman, 2005) information about the target that is not covered by the preceding questions in the series. One or more human assessors determine correct answers for factoid and list questions. The final set of judgments used in the TREC evaluations are a composite of the judgments of individual assessors. The official scores for the "Other" questions are computed using multiple assessors' judgments of the importance of information nuggets.

It should be noted here that if no answer is found in the document collections, as will be the case for some factoid questions, then the correct answer will be "NIL.” In other words, it is important for the QA system to identify those cases for which it cannot find an answer. A question is assumed to have no correct answer in the collection if the TREC assessors do not find an answer during the answer verification phase and no participating QA system returns a correct response that is supported by a document in the corpus. If a participating QA system returns a right answer that is unsupported in the document collection, NIL will still be the correct response for that question.

Time-dependent questions are those for which the correct answer can vary depending on the timeframe that is assumed. When a question is phrased in the present tense, the implicit timeframe will be the date of the last document in the document collection; thus, QA systems are required to return the most up-to-date answer supported by the document collection. When the question is phrased in the past tense then either the question will explicitly specify the time frame (e.g., "What cruise line attempted to take over NCL in December 1999?") or else the time frame will be implicit in the question series. For example, if the target is the event "France wins World Cup in soccer" and the question is "Who was the coach of the French team?" then the correct answer should be

the name of the coach of the French team in 1998 when France won the World Cup. In this case, the name of any past or current coach of the French team is an incorrect response.

The question set is given in an XML format. The format explicitly tags the question target as “target”, and each question in the series is specifically marked as “FACTOID,” “LIST” or “OTHER.” Each question will have an identification number of the form X.Y where X is the identifier for the target and Y is the sequence number of the question in the series. Factoid questions are questions that seek short, fact-based answers. List questions are requests for a set of instances of a specified type. Factoid and list questions require exact answers to be returned. Responses to the "other" question need not be exact, though excessive length will be penalized.

An example of a TREC question series is shown in Figure 5 below.

```
<target id="153" text="Alfred Hitchcock">
<qa>
  <q id="153.1" type="FACTOID">When was Hitchcock born?</q>
  </qa>
<qa>
  <q id="153.2" type="FACTOID">Where was Hitchcock born?</q>
  </qa>
<qa>
  <q id="153.3" type="FACTOID">What was Hitchcock's first
    movie?</q>
  </qa>
 <qa>
  <q id="153.4" type="LIST">List his movie nominations for best
    director.</q>
  </qa>
<qa>
  <q id="153.5" type="FACTOID">How many Oscar awards did
    Hitchcock win?</q>
  </qa>
<qa>
  <q id="153.6" type="FACTOID">When did Hitchcock die?</q>
  </qa>
<qa>
  <q id="153.7" type="FACTOID">Where did Hitchcock die?</q>
  </qa>
<qa>
  <q id="153.8" type="OTHER">Other</q>
  </qa>
</target>
```

Figure 5: Example TREC Question Series for Target = "Alfred Hitchcock"

3.1.2 TREC Question Answering Corpus

In past TREC challenges, all questions had been posed against the Advanced Question Answering for Intelligence Project's AQUAINT and AQUAINT II corpuses. AQUAINT and AQUAINT II are both collections of English newswire articles from a variety of sources including: Agence France Presse, Central News Agency (Taiwan), Xinhua News Agency (China), the Los Angeles Times-Washington Post News Service, the New York Times, and The Associated Press. DTD files distributed along with the

datasets describe the specifics of the document formats from each source agency. Each document in the corpus has a unique document identifier that specifies both the source newswire service, the date when the article was posted to the newswire service, and a unique counter that distinguishes that newswire article from other articles posted on the same source newswire service on the same day. The AQUAINT corpus contains roughly 1 million documents spanning the years from 1996 – 2000. The AQUAINT II collection comprises approximately 2.5 GB of text (about 900,000 documents) spanning the time period of October 2004 - March 2006. An example of a document from the TREC 2006 corpus is presented in Appendix A.

In 2007, a new wrinkle was introduced into the Question Answering task when a corpus of blog documents (Blog06) was added to the document corpus. The 2007 TREC QA task differed from previous years in that questions were asked over both blog documents and as well as newswire articles, rather than just over newswire alone. A blog document is defined to be a blog post and its follow-up comments. While newswire is typically composed of well-formed sentences that have been through an editorial process, documents in the blog collection may contain well-formed or badly-formed English. Furthermore, as blogs are typically based on the opinion of an individual, there is no guarantee that the information contained within a blog post is valid. In fact, there is a high likelihood of a significant amount of blog postings being either exaggerations of the truth or outright spam. Mining blogs for answers introduces significant new challenges in at least two aspects that are very important for functional QA systems:

- Being able to handle language that is not well-formed and which does not adhere to any fixed set of grammatical or linguistic rules

- Dealing with discourse structures that are less reliable than newswire.

The Blog06 corpus is distributed by the University of Glasgow and is the same collection as was used in the TREC 2006 Blog Track. Each blog document is a collection of permalinks in which the raw HTML content from the Web is wrapped between a <DOC> </DOC> pair of XML tags. There is also some informational metadata tags associated with the information contained in each permalink, including a document identification number. The Blog06 collection spans roughly the same dates as the AQUAINT II collection, which provides a means for a QA system to perform checks of blogged information against newswire.

It should be noted here that we propose that newswire contains facts and blogs contain opinion. In other words, we largely ignore any biased information that may be distributed via traditional news outlets. The emerging field of sentiment analysis (Godbole, Srinivasaiah, & Skiena, 2007) deals specifically with identification of bias within documents, but a comprehensive discussion of sentiment analysis within the context of Question Answering is beyond the scope of this thesis.

Unlike the AQUAINT and AQUAINT II document collections, usage of the Blog06 collection required TREC participants to pay an additional fee to the University of Glasgow for usage. While this was not a deterrent for some groups of TREC participants, several other groups (including our team from Drexel) were unable to acquire the Blog06 data. As such, some of the aims of the TREC QA organizers may not have been fully realized. Groups who did not have access to Blog06 data competed in TREC 2007, but at a disadvantage. And their algorithms were unable to implement the

features that would distinguish a fact-based well-formed source document from an opinion-based un-edited source document.

3.1.3 TREC Question Answering Judgments

A TREC QA submission consists of exactly one response for each question. The definition of a response varies for each of the different question types. In this section we describe what constitutes a correct answer for each of the TREC QA question types.

For factoid questions, a response is a single answer string and document identifier pair or the string "NIL." The "NIL" string will be judged correct if there is no answer known to exist in the document collection; otherwise it will be judged as incorrect. If an answer string and document id pair is given as a response, the answer string must contain nothing other than the answer, and the document identifier must correctly reference a document in the collection that supports the answer string as an answer. The identifier of a Blog06 document is the <DOCNO> element from the XLM file that contains the set of permalinks associated with a blog document. The identifier of an AQAUINT and AQUAINT II document is the identifier attribute of a <DOC> element, as specified by their respective DTD files. The answer string does not have to appear literally in a document in order for the document to support it as being the correct answer. Some additional axiomatic knowledge and temporal inference processing can be used to correctly identify the answer including:

- Temporal ordering of months and days of the week
- Calculation of specific dates from relative temporal information provided in a document. For example, "July 2007" is a supported correct answer to "When was

the final Harry Potter book published?" if it is supported by a document dated June 2007, containing the phrase "the final Harry Potter book will be published next month..."

Sometimes, different documents may support contradictory answers as being correct. A response is said to be *locally correct* when the supporting document supports the answer string as being correct, and the answer string contains exactly the correct answer. Locally correct answers are assumed to be *globally correct* unless there is a better, contradictory answer supported in the document collection. TREC assessors may use a number of criteria in determining that one locally correct answer is better than another, including the date of the supporting document (more recent is better), the amount of support provided by each supporting document, the number of distinct sources that support the answer as being correct, and the authoritativeness of the source. The assessor will mark as *globally correct* one or more of the most reliable of the known locally correct answers. It should be noted here that "global" correctness is defined with respect to the document collection, and not necessarily with respect to the real world.

An answer string must contain a complete, exact answer and nothing else. Responses will be judged by human assessors who will assign one of five possible judgments to a response:

- Incorrect: the answer-string does not contain a correct answer or the answer is not responsive
- Unsupported: the answer-string contains a correct answer but the document returned does not support that answer

- Non-exact: the answer-string contains a correct answer and the document supports that answer, but the string contains more than just the answer or it is missing parts of the answer
- Locally correct: the answer string consists of exactly a correct answer and that answer is supported by the document, but the document collection contains a contradictory answer that the TREC assessor believes is better.
- Globally correct: the answer-string consists of exactly a correct answer, that answer is supported by the document, and the document collection does not contain a contradictory answer that the assessor believes is better.

In this context, "not responsive" can be taken to mean imprecise. For example, if the question is asked “How much does a gallon of gas cost?” an answer of “3.00” would be taken to be imprecise since units are not specified inside the answer string. Another example of a non responsive answer deals with the difference between famous entities and their replicas. If the question is asked, “Where is the Taj Mahal?” a responsive answer would return “Agra, India” while a non-responsive answer would refer to the Taj Mahal casino in Atlantic City, New Jersey. A complete description of responsiveness is provided in the TREC-8 Question Answering Track Report (Voorhees, 1999). It should be noted here that a factoid question may have more than one globally correct response in the corpus.

For “LIST” questions, a response is an unordered, non-empty set of answer string and document identifier pairs, where each pair is called an instance. The interpretation of the pair is the same as for factoid questions, and each pair will be judged in the same way as for a factoid question. It should be noted here that this means that if an answer string

actually contains multiple answers for the question it will be marked inexact and will thus hurt the question's score. In addition to judging the individual instances in a response, the assessors will also mark a subset of the instances judged globally correct as being distinct. If multiple globally correct instances contain answer strings that are conceptually identical, the assessor will mark exactly one arbitrary instance as distinct; in other words, a correct answer should only show up once in the list response. Scores will be computed using the number of globally correct, distinct instances in the set. It should be noted here that “NIL” is never a correct answer for a “LIST” question.

Like “LIST” questions, the response for the "OTHER" question that ends each target's question series is also an unordered, non-empty set of answer string and document identifier pairs. However, the interpretation of the pairs differs somewhat from a “LIST” or “FACTOID” response. A pair in an "OTHER" response is assumed to represent a new interesting fact about the target that has not been divulged by the previous questions in the series. Individual answer string and document identifier pairs are not be judged for these questions. Instead, the TREC assessors construct a list of desirable information nuggets about the target, and count the number of distinct, desirable nuggets that occur in the response as a whole. There is no expectation of an exact answer to "OTHER" questions, but responses will be penalized for excessive length.

Once the TREC assessors have finished the judgment process, the final set of judgments for all questions is released to the TREC community. Individual answers are marked with scores that indicate whether they are non-exact, locally correct, or globally correct, allowing for some degree of flexibility when the TREC testbed is used in experimentation.

3.2. TREC 2007 Question Answering Track Entry

This section describes the first time participation of Drexel University's Intelligent Information Processing group in the TREC Question Answering Track. Our primary goal for TREC 2007 was the development of a Question Answering system framework, and to use this framework as a vehicle through which we could understand QA challenges and issues. As a result, our results this year were not significant; our primary accomplishment was the establishment of a baseline system, on top of which later enhancements can be applied.

Next, we will describe the components of our preliminary system architecture. We will explain each component in a generic sense, as well as their specific implementations in our system this year. Our system architecture is shown in Figure 6 below. As previously stated, the first stage of our architecture is primarily an Information Retrieval (IR) stage. In this stage, the AQUAINT II documents were pre-processed using stop-word removal, stemming, and indexing using tools provided as a part of the LEMUR toolkit (http://www.lemurproject.org). Once the corpus pre-processing was completed, we used the Indri (Strohman, Metzler, Turtle, & Croft, 2005) search engine to perform document retrieval. Indri queries were created from the TREC questions and targets using Indri's query-likelihood retrieval method which is based on the Ponte and Croft language modeling approach using Dirichlet priors for smoothing (Ponte & Croft, 1998). The top documents returned from Indri were then considered to be the set of Candidate (answer-bearing) Documents. These documents were then sent on to the

second Question Answering stage for further processing. We considered Candidate Documents sets in sizes of 5, 10 and 25 documents for the three runs submitted this year.

We reiterate here that due to time and budgetary constraints we utilized only the AQUAINT II corpus for our experiments. No Blog data was included as a part of our submission this year, which further contributed to our lack of significant results.

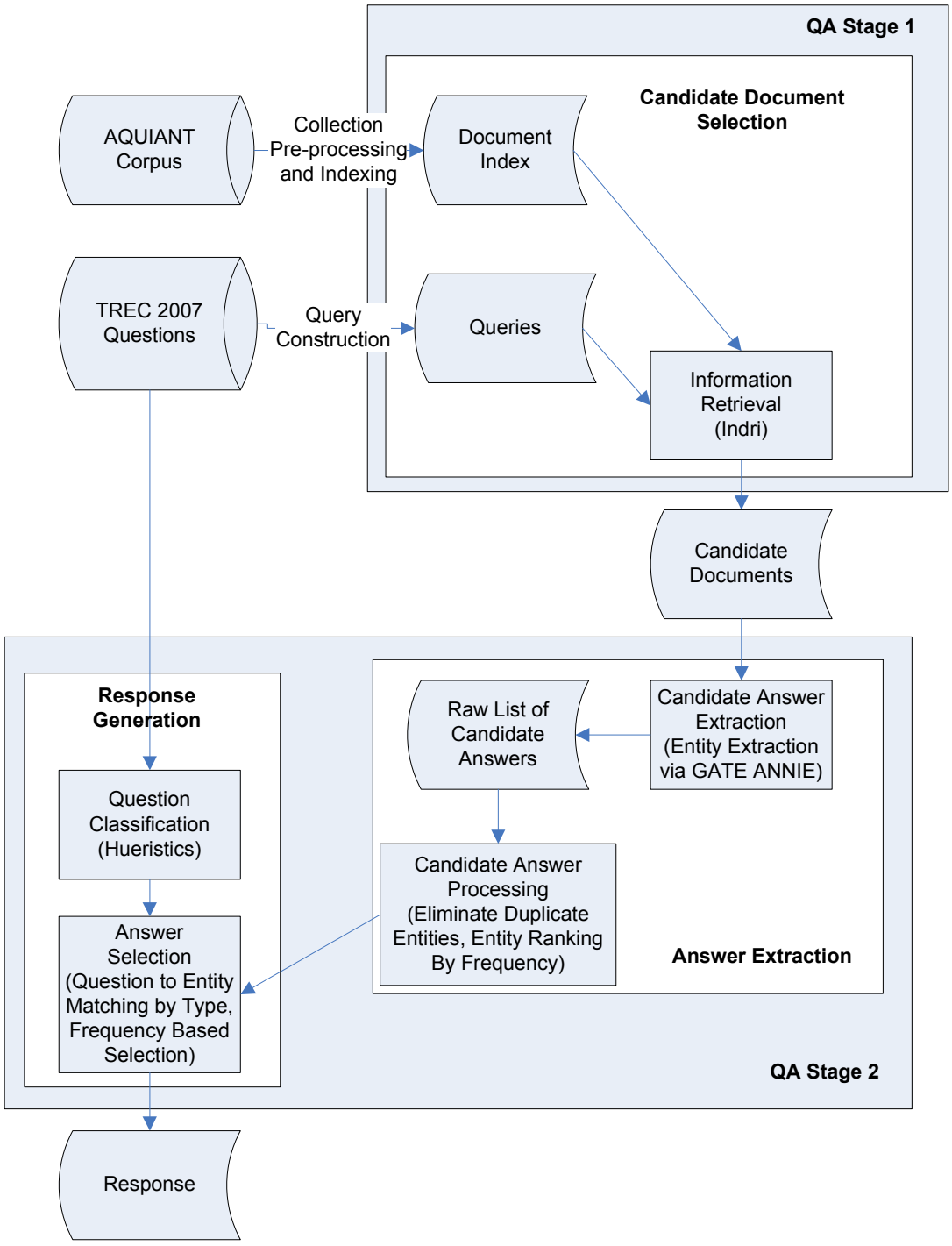

Figure 6: 2007 TREC Question Answering Entry System Architecture

The set of Candidate Documents were sent to a Candidate Answer Extraction module. Our current implementation of this module made use of the Information Extraction (IE) resources provided by the GATE ANNIE Toolkit (Bontcheva, Tablan, Maynard, & Cunningham, 2004) (http://gate.ac.uk/ie/annie.html). ANNIE was used to extract all entities of type "Person," "Location," "Organization," "Date," "Currency," and "Percentage" from the Candidate Documents. These entities, which we considered in a generic sense to be the Raw List of Candidate Answers, were stored in an Oracle database for further processing.

We then performed two types of basic processing on the Raw List of Candidate Answers. First, we eliminated duplicates using simple string matching functions provided by Oracle. Then, we grouped the answers by entity type (eg. "Person," "Location," "Organization") and ranked the entities according to their frequency of occurrence in the set of Candidate Documents. At this point, we returned to the original query to try to understand what type of entity would best answer the question that was posed. We created a simple set of heuristics based on the existence of keywords within the query such as "who," "what," "when," "where," "why," "how many" and "how much." Having reviewed the query classification scheme proposed by Lehnert (Lehnert, 1977), we recognize that this is an area where our approach can use significant improvement. The output of our simple classification scheme was an Answer Type, which is represents the entity type that will correctly answer the question.

Knowing the Answer Type and the set of Candidate Answers allowed us to implement a simple matching algorithm. Our final response to the question was the top ranked Candidate Answer of the proper Answer Type. We then went back to our Raw

Answer List to determine the document from which the top-ranked Candidate Answer came. Our final response to the question consisted of the top-ranked Candidate Answer and its associated Document Number.

The above approach was designed primarily with factoid questions in mind. We were able to re-use the same approach for list questions. In the case of list questions, however, we selected a set Candidate Answers to be returned as the query response. We selected this set by considering answers which ranked above a fixed frequency threshold. Our approach does not currently incorporate a mechanism to respond to "OTHER" questions.

As might be expected, our results this year were not significant. We submitted three runs, using Candidate Document sets of 5, 10 and 25 documents, and achieved accuracy results of 0.016, 0.019, and 0.022 for these runs respectively. The accuracy results of our system were thus directly influenced by the quantity of data returned by the IR stage of our system. However, our results are far below the accuracy results for state-of-the-art systems. We recognize this, and focus our efforts on enhancements to specific components going forward.

In conclusion, our primary goal this year was to establish a framework for our future work in Question Answering. We started with a holistic approach to the Question Answering process, using the generic Question Answering architecture proposed by Hirschman (Hirschman & Gaizauskas, 2002) as a foundation. From these preliminary efforts, we choose to target our work in Question Answering to focus on several key areas: development and management of a Question Answering Context using statistical language modeling methods, development of robust algorithms for Answer Validation,

and the incorporation of statistically generated semantic information into the Question Answering process.

The TREC Question Answering exercises were unfortunately discontinued after 2007. As such, we did not have a forum to further present our experiments.

3.2. Preliminary Studies - Language Modeling Applied to QA

This section describes our preliminary work in applying language modeling to Question Answering. Our primary goal for these activities was to gage the effectiveness of language modeling techniques applied to QA, as compared to other (more traditional) IR techniques. Our preliminary results show that language modeling techniques outperform other state-of-the-art IR approaches for the QA paradigm.

Roberts and Gaizauskas (Roberts & Gaizauskas, 2004) introduce the metrics of coverage and redundancy to describe the effectiveness of an IR system with the context of the QA task. Coverage is defined to be "the proportion of the question set for which a correct answer can be found within the top *n* passages retrieved for each question (Roberts & Gaizauskas, 2004)." Mathematically, for a question set Q, we can define the coverage to be:

$$\text{coverage}(Q, D, n) = \frac{|\{q \in Q\} | R_{D,q,n} \cap A_{D,q} \neq 0|}{|Q|} \qquad (25)$$

where *R* is the set of *n* top ranked documents contained in a document collection *D* and *A* is the subset of those documents that contain the correct answers for each question *q* within the question set. Put another way, coverage is the maximum likelihood that the second stage of the QA system has of answering the question correctly, given the documents that are returned by the IR stage. Redundancy "gives the average number, per

question, of passages within the top ranks retrieved which contain a correct answer." (Roberts & Gaizauskas, 2004) Mathematically, for a question set Q, we can define the redundancy to be:

$$redundancy(Q, D, n) = \frac{\sum |R_{D,q,n} \cap A_{D,q}|}{|Q|} \qquad (26)$$

where *R*, *D* and *A* have the same definitions as above. Intuitively, redundancy quantifies the number of times the correct answer appears in the documents returned by the IR engine. IR systems with a higher redundancy score affords a QA system more opportunities of capturing the correct answer in the second stage of processing.

Our preliminary experiments in applying language modeling techniques to QA were conducted using a subset of the TREC 2005 and TREC 2006 QA Track questions. We manually reviewed each question in the TREC 2005 and TREC 2006 question sets to extract only those questions that returned a person's name as the correct answer; we call this category of questions "who"-type for obvious reasons. This culling was performed to streamline the evaluation process. The answer for a question that returns a person's proper name is relatively straight-forward; answers to other types of questions such as those which require a date or numeric type response are more difficult to evaluate given our limited subject matter expertise and resources. Once the culling process was concluded, a final set of 101 "who"-type questions was developed for this experiment, spanning a variety of topic types. These 101 questions include all "who"-type questions present in the TREC 2005 and TREC 2006 question sets.

Our question set was processed against the AQUAINT corpus, which was used in QA tracks of both TREC 2005 and TREC 2006. The AQUAINT documents were pre-processed using stop-word removal, stemming, and indexing. Once the corpus pre-

processing was complete, queries were constructed for the various IR strategies used in this experiment and an IR engine was invoked to produce the set of candidate documents. We considered candidate document sets ranging in size from 1, 5, 10, 25, 50 and 100 documents. The various IR strategies employed will be discussed in greater detail later on in this section.

The set of documents produced by the IR engine were sent to the GATE ANNIE (Bontcheva et al., 2004) for Information Extraction (IE) (http://gate.ac.uk/ie/annie.html). ANNIE was used to extract all entities of "Person" type from the candidate documents. These entities were stored within an Oracle database for evaluation. Judgments in Questions Answering (QA) tracks of TREC 2005 and TREC 2006 were used to evaluate the extracted entities for correctness. All names that were judged correct by the TREC evaluators were considered to be correct; it should be noted that we did not consider portions of names to be correct if they were not explicitly judged correct. So, for example, the correct responses to the question "Who was Carolyn Bessette Kennedy married to?" were limited to only those variants of John F. Kennedy Jr. judged correct by the evaluators.

Figure 7 below depicts the overall process of our experimentation methodology.

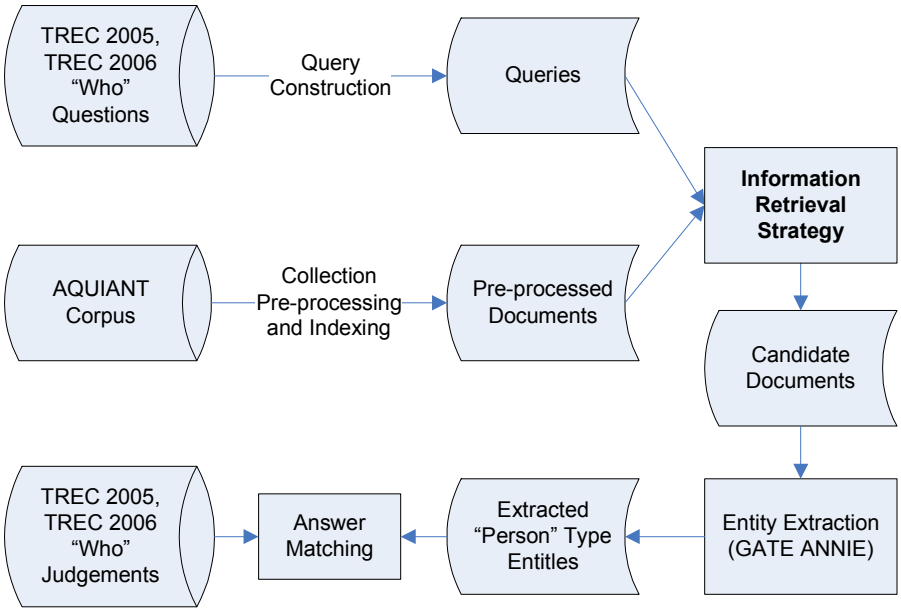

Figure 7: Experimentation Methodology for Preliminary Experiments Applying Query-Likelihood Models to Question Answering

Our experiments were conducted using three well-established IR approaches: (1) the query likelihood language model-based approach, (2) Okapi BM25 and (3) the traditional Tf-Idf. We used the Indri "#combine" operator to invoke query likelihood retrieval; the default Indri settings were used for our experiments including the Dirichlet prior smoothing methodology (C. Zhai & Lafferty, 2004).

We utilized the LEMUR implementations of Okapi and Tf-Idf. For both these approaches, we created a custom stop-word list including words such as "who", "which" and "name" that frequently appeared within question text. The ParseToFile utility was used to construct queries based on the question text and the stop-word list; we also employed Porter stemming during query construction. Both Okapi BM25 and Tf-Idf were run using the default LEMUR parameters. Indri (Strohman, Metzler, Turtle, & Croft, 2004) was used for the query likelihood language model-based approach in our experiment.

The coverage and redundancy formulae presented earlier in this section consider passage level retrieval. That is, these metrics describe the top *n* passages retrieved. In our experiment, we consider a single document in its entirety as a passage, and we did not break down the document into its constituent passages. This was done for several reasons: (1) there are numerous methods of breaking down a document into its constituent passages, (2) different IR approaches favor one approach over another, and (3) the interpretation of a "passage" may be different for the various IR algorithms employed in this study.

The coverage and redundancy results for Indri query-likelihood document level retrieval, Okapi BM25 and Tf-Idf are presented in Figure 8 and Figure 9 below. The Okapi results are relatively consistent with those presented by Roberts and Gaizauskas (Roberts & Gaizauskas, 2004), although we used different question sets and evaluation techniques. For example, using Okapi we show a coverage percentage of 60% after retrieving 10 documents, 73% after retrieving 50, and 81% after retrieving 100 documents. By way of comparison, Roberts and Gaizauskas report coverage percentages of 60%, 74% and 79% after retrieving 10, 50 and 100 documents respectively. This close correlation in results indicates a close correlation between our study and theirs. There is, however, a variation in redundancy between our results and those reported by Roberts and Gaizauskas; this is likely due to differences in questions and document collections.

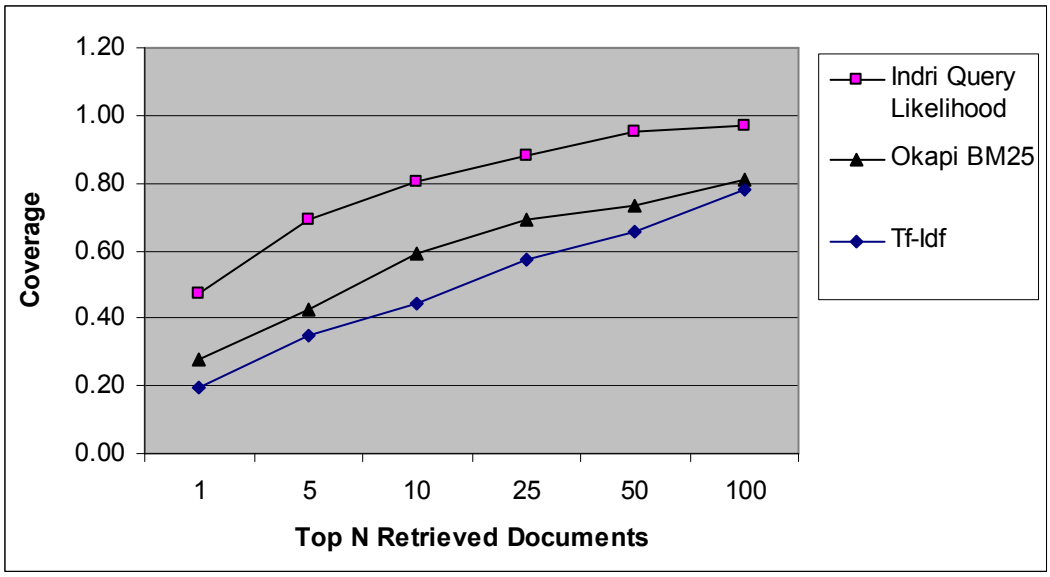

Figure 8: Candidate Document Coverage Results for Initial Experiments with the Query Likelihood Language Model and Question Answering. Baseline approaches include a) Okapi BM25 and b) tf-dif.

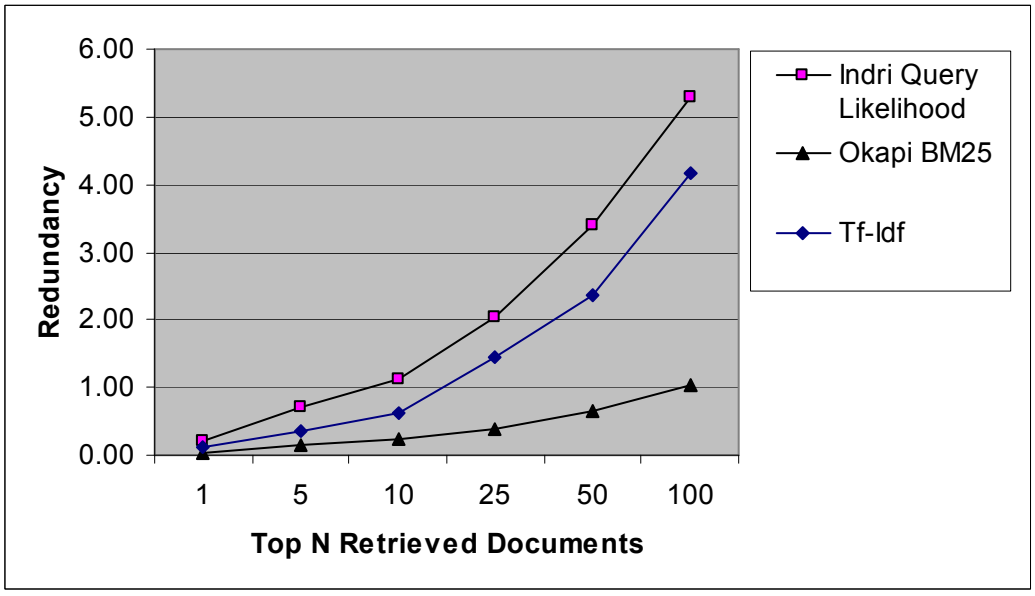

Figure 9: Candidate Document Redundancy Results for Initial Experiments with the Query Likelihood Language Model and Question Answering. Baseline approaches include a) Okapi BM25 and b) tf-dif.

As can be seen, the query likelihood language model outperforms both the Okapi BM25 and Tf-Idf approaches in terms of both coverage and redundancy. Using the query likelihood language model measure we achieve coverage of 97% after retrieving 100 documents. This is a significant result as it shows that IR engines utilizing a query likelihood language model approach are likely to return documents containing a correct response 97% of the time for "who"-type questions. We can improve the performance of the second stage of our QA system by using this result to set the number of candidate documents processed, as well as to determine the method by which those documents are retrieved.

The redundancy measurement of the query likelihood language model approach is also significant. From the data above, we can see that (on average) the query-likelihood model is two times as likely to have redundant information as Tf-Idf, and five times as likely to have redundant information as Okapi BM25. That is, utilizing a query likelihood language model approach in the first stage will give the second stage of our QA system a higher number of opportunities (on average) to find the correct answer within the set of retrieved documents.

It is interesting to note that while Okapi BM25 had higher coverage than Tf-Idf, Tf-Idf shows higher redundancy. This is true because the Tf-Idf algorithm favors longer documents; in other words, although it is more likely that the right answers will appear in the document set returned by Okapi, Tf-idf is more likely to return multiple occurrences for those correct answers that it does report. QA systems may find this characteristic helpful in certain situations.

In conclusion, our preliminary results show that the utilization of a language model-based IR strategy in the first stage of a QA system has significant advantages. The coverage and redundancy measures we report indicate that the top 100 documents returned using the query likelihood language model approach are likely to contain the correct answer 97% of the time, and will have higher occurrences of that correct answer. These results have impact not only on the design of the first stage of our QA architecture, but will also have design implications on the answer extraction processes in the second stage.

In (Spärck Jones, 2004) Karen Sparck-Jones is skeptical about the application of the language modeling to the question answering paradigm. She questions the appropriateness of a generative model when considering the problem of question answering; in the language modeling framework, the question and the answer would need to be composed from the same underlying model – which does not necessarily make intuitive sense. The goal of this section has been to explore the application of language modeling to the IR stage of QA, and for the remainder of this thesis we consider language modeling as a broader QA strategy. In this section, we reveal that the language model approach outperforms other approaches in the first stage of a QA system. In the remainder of this thesis, we continue the application of language modeling to other parts of the QA architecture.

3.3 The Aspect-Based Relevance Language Model

This section describes our core theoretical model, the Aspect-Based Relevance Language Model. We first return to the Aspect Model from Probabilistic Latent

Semantic Analysis (PLSA) and Relevance-Based Language Models as background. Although both of these language modeling approaches were discussed in our literature review, we feel that a more in-depth discussion is appropriate here. We next use formal methods to derive the Aspect-Based Relevance Language Model, and describe qualitatively how this model can be used to create a Question Context Model.

3.3.1 The Aspect Model

The Aspect Model forms the foundation of the Probabilistic Latent Semantic Analysis (PLSA) method proposed by (Hofmann, 1999). The underlying premise of the Aspect Model is that we can define words and documents in terms of "aspects" which are associated with a latent class variable. The Aspect Model is based on two underlying assumptions:

- Words and documents are independent of one another (bag of words assumption)
- Documents and words are both conditioned on a latent aspect (z) which may be thought of as a concept

The Aspect Model has several features that make it intuitively appealing for our research. First, by conditioning words and documents on a latent variable, the zero-frequency problem is addressed. Secondly, a priori knowledge is not required about the concepts within the corpus for the algorithm to work effectively. And finally, the usage of probabilistic methods defines a generative model of the data which is better able to address common text processing issues such as synonymy and polysemy. At a high level, the Aspect Model uses the following approach:

- Select a document d with probability $P(d)$
- Pick a latent class z with probability $P(z|d)$
- Generate a word with probability $P(w|z)$
- Formulate the probability of observing a pair (d,w), while the latent variable z is discarded.

PLSA is described mathematically by Equation (22) and depicted graphically in Figure 4. PLSA is based on maximizing the log-likelihood of the divergence between the empirical distribution of observed word-document pairs and the probabilistic model P(w,d). PLSA uses the Expectation-Maximization (Dempster et al., 1977) algorithm to estimate the probabilities P(z), P(w|z) and P(d|z) for latent variable models. In the remainder of this section, we explore the steps of the PLSA algorithm in detail to get a better understanding of how the z categories are specifically determined:

1. PLSA Initialization
    a. The number of documents N is initialized to the total number of documents in the corpus
    b. The number of words n is initialized to the total number of unique terms in the corpus
    c. The best test LogLikelihood value is initialized to negative infinity
    d. Three p(z) arrays are each initialized to 0 for all z values. One array represents the p(z) calculated by current algorithm iteration, one array represents the p(z) calculated by the previous algorithm iteration, and the last are the p(z) values from the best (highest LogLikelihood) iteration.

e. Three p(w|z) matrixes are initialized to 0 for all w and all z. One matrix represents the p(w|z) calculated by current algorithm iteration, one matrix represents the p(w|z) calculated by the previous algorithm iteration, and the last are the p(w|z) values from the best (highest LogLikelihood) iteration.

f. Three p(d|z) matrixes are initialized to 0 for all d and all z. One matrix represents the p(d|z) calculated by current algorithm iteration, one matrix represents the p(d|z) calculated by the previous algorithm iteration, and the last are the p(d|z) values from the best (highest LogLikelihood) iteration.

2. Next, initialize the inverted matrix that maps words (unique terms) to the documents in which they occur
3. The document corpus is randomly divided into a training and test set. The test set is randomly set to be a percentage of the documents (as specified by input parameters – default is 10%). The remainder of the documents is considered to be the training set.
4. For each iteration of the algorithm:
    a. One of the inputs to the algorithm is the maximum number of Expectation-Maximization iterations that should be performed.
    b. This number is set as the upper bound on the number of times the algorithm iterates. If the algorithm converges before it reaches this maximum number, it stops.
    c. The algorithm computes LogLikelihood measures for p(z), p(d|z) and p(w|z)

d. The first iteration of the algorithm is highly influenced by the term frequency weights

    i. So, tf(w) and tf(w,d) heavily influence p(w|z) and p(d|z) respectively for all z

e. As the algorithm progresses, the following E-step and M-step calculations are performed alternately until the algorithm converges :

$$P(z \mid d, w) = \frac{P(z)P(d \mid z)P(w \mid z)}{\sum_{z'} P(z)P(d \mid z')P(w \mid z')} \quad \text{E-step} \quad (27)$$

$$P(w \mid z) = \frac{\sum_{d} n(d, w)P(z \mid d, w)}{\sum_{d, w'} n(d, w')P(z \mid d, w')} \quad \text{M-step} \, (28)$$

$$P(d \mid z) = \frac{\sum_{w} n(d, w)P(z \mid d, w)}{\sum_{d', w} n(d', w)P(z \mid d', w)} \quad \text{M-step} \, (29)$$

$$P(z) = \frac{1}{\sum_{d, w} n(d, w)} \sum_{d, w} n(d, w)P(z \mid d, w) \quad \text{M-step} \, (30)$$

Here, n(d,w) denotes the term frequency or the number of times that the word n occurred in the document d. The values P(z), P(d|z) and P(w|z) which are calculated in the M-steps and then used to calculate the next variation of the E-step. Then, the value of P(z|d,w) calculated in the E-step is then used in the M-step calculations. The variables w', d' and z' are summation variables that are used for the purposes of distinguishing the contents of the summation from the conditioned variables w, d and z.

We present this in-depth description of the PLSA algorithm so that the reader can appreciate the mechanics of how conditional probabilities are assigned to latent aspects,

words and documents. In practice, a variation of the PLSA algorithm, which incorporates a Tempered Annealing technique called Tempered Expectation Maximization (TEM) (Hofmann, 1999) is used to modify the M-step to make it more generally applicable across corpus types. TEM introduces a control parameter called β, which modifies only the M-step of the algorithm.

3.3.2 Relevance-Based Language Models

One of the main obstacles to effective performance of the classical probabilistic IR models has been precisely the challenge of modeling relevance. However, estimating relevance in a typical retrieval environment is difficult because of the lack of training data. In a typical retrieval environment, we are given a query and a large collection of documents without any indication of which documents might be relevant. These conditions are further exacerbated by a Web-search environment where the number of documents and their contents are changing in real-time. Faced with the absence of training data, researchers have used heuristic estimates for relevance, leading to models that are difficult to interpret. Lavrenko (Lavrenko & Croft, 2001) notes that estimating the likelihood of non-relevance is easier that estimating the likelihood of relevance, since we have plenty of training data. For a typical query, almost every document in the collection is non-relevant if the collection is assumed to be large.

The famous probability ranking principle, advocated by Robertson in (S. E. Robertson, 1997), asserts that optimal performance is achieved by an IR system only if the documents are ranked by the posterior probability that they belong to the relevant set of documents. Robertson (S. E. Robertson, 1997) also shows that it is equivalent to rank

the documents by the likelihood of being observed in the relevant class divided by the likelihood of being observed in a non-relevant class. Lavrenko (Lavrenko & Croft, 2001) makes the common assumption that a document is composed of a "bag-of-words" each of which is independent of each other to arrive at the following equation:

$$\mathrm{Re}\,levance \sim \frac{P(D \mid R)}{P(D \mid N)} \cong \prod_{w \subset D} \frac{P(w \mid R)}{P(w \mid N)} \qquad (31)$$

Here, P(D|R) is the likelihood that a document D is relevant to an information need and P(D|N) is the likelihood that a document D is not relevant to an information need. Similarly, P(w|R) is the likelihood that a word w contained in D is relevant to an information need and P(w|N) is the likelihood that a word w contained in D is not relevant to an information need.

If we had available training data in the form of relevance judgments, estimating P(w|R) could be as simple as counting the number of occurrences of w in the relevant documents and then appropriately smoothing the term counts. However, in a typical retrieval setting we are given only a large collection of documents and a user query, and we do not know which documents comprise the relevant set. However, Lavrenko proposes that we do know that all the relevant documents are somehow related to the information need. Lavrenko assumes that for every information need (or user query) Q, there exists an underlying relevance model R, which assigns the probabilities P(w|R) to the word occurrence in the relevant documents. The relevance model also assigns probabilities P(Q|R) to the various queries that might be issued by the user for that specific information need. A conceptual representation of the Relevance-Based

Language Model is shown in Figure 10.

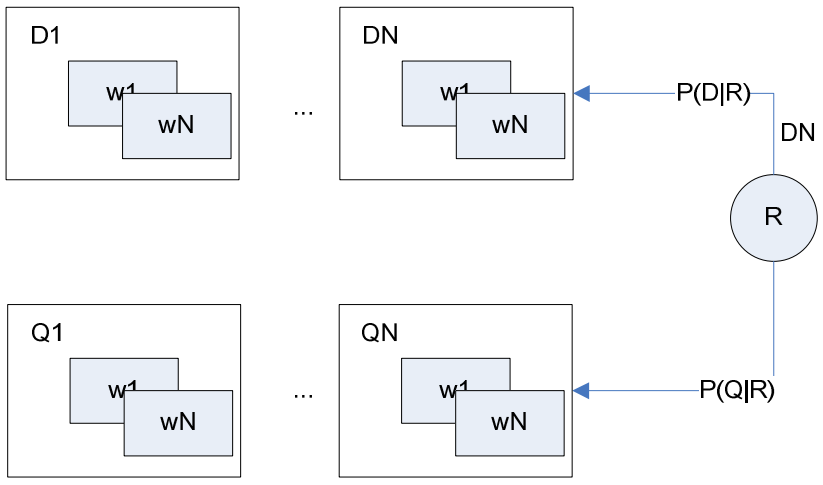

Figure 10: Conceptual Representation of the Relevance-Based Language Model

Lavrenko proposes that the query Q is also a sample from R; however, the sampling process that generates the terms in Q does not have to be the same as the process that generates the words in the relevant documents. In other words, the probability of a one-word query "w" given a relevance model R need not be the same as the probability of observing the same word w in a random relevant document, given the same relevance model R. Lavrenko does not assume that the query is a sample from any specific document model, as was the case with past language modeling approaches based on a query-likelihood assumption. Instead the Relevance-Based Language Model assumes that both the query and the documents are samples from an unknown relevance model; known information about the queries and pseudo-relevance feedback information from documents can then be used to approximate the relevance model.

To this end, Lavrenko proposes the following scenario which we leverage for the design of the Aspect-Based Relevance Language Model. Suppose that we have a black box from which we can repeatedly sample words. After sampling m times, we observe the words $q_1, \ldots, q_m$ coming out of the black box. We can then relate the probability that w will be the next word coming out of the black box to the conditional probabilities of the words $q_1, \ldots, q_m$ that we have already observed. In other words:

$$P(w \mid R) \approx P(w \mid q_1, \ldots q_m) \qquad (32)$$

Two methods of estimating the probability $P(w \mid q_1, \ldots, q_m)$ are described in (Lavrenko & Croft, 2001). Both methods assume the existence of a set U of underlying source distributions from which w and $q_1, \ldots, q_m$ could have been sampled. They differ in their independence assumptions. Method 1 assumes that all query words and the words in relevant documents are sampled from the same distribution. Method 2 assumes that the query words $q_1, \ldots, q_m$ are independent of each other but are dependent on w.

3.3.3 Motivation for the Aspect-Based Relevance Language Model

One of the fundamental problems associated with natural language Question Answering is the general sparseness of the queries presented by a user. For example, the question "What position does Warren Moon play?" from the TREC 2006 Question Answering Track (Voorhees, 2005a) requires a fair amount of background information before it can be correctly "understood" by either a person or a system. To correctly answer the question, one must realize that Warren Moon is a person, and that he plays football, and that in football players are assigned regular positions. In other words, there

is a context associated with this query that plays a critical role in our understanding of the question and our understanding of what entails a reasonable answer.

We propose that the Aspect Model can be used to supplement the sparse information of a query with information from concepts related to query terms within the corpus. So, in the example above, the terms "position," "play," and "Warren Moon" would be the terms in the query that should be relatable to concepts within the corpus. We can review the latent concepts for which "position," "play" and "Warren Moon" rank most highly, and then choose to incorporate words associated with those concepts into our corpus.

The problem, however, with this approach is that we have no way of understanding what concepts are most relevant to this particular query. In our example above, the word "position" may occur in multiple concepts, each with a meaning that may be fundamentally un-related to our particular situation. The same also applies to the term "Warren Moon," which might occur in the context of accepting an award, engaging in disputes over salary, etc. In other words, we need to be able to determine which concepts are most relevant to this particular issuance of the query.

3.3.4 The Aspect-Based Relevance Language Model

We base our approach on the same set of assumptions that guide the Relevance-based Language Modeling approach. We assume that for every information need (or question) there exists an underlying relevance model R. In the Relevance-based Language Modeling paradigm, R assigns probabilities $P(w|R)$, which are the word occurrences in the relevant documents for the information need. In practice, since we do

not know which documents are relevant to an information need R is unknown and $P(w|R)$ is unknown.

In our model, we assume that R is assigned probabilities $P(z|R)$ where z is a latent aspect of an information need, as defined by the Aspect Model. Thus, the relevance to an information need is described not in terms of words, but in terms of the latent aspects or concepts associated with the information need. Conceptually, this alternative representation of the information need includes context information via the inclusion of aspects that extends beyond knowledge related to the specific words themselves.

However, in this case, as with the Relevance-based Language Model, R and $P(z|R)$ are unknown. However, the query, which is composed of terms $q_1,\ldots,q_N$ is known. (Lavrenko & Croft, 2001) assumes that we can approximate a relevance model R, by viewing the query terms $q_1,\ldots,q_N$ are a random sampling of words from R. Using the same assumption, we can approximate:

$$P(z \mid R) \approx P(z \mid q_1,...,q_N) \qquad (33)$$

Thus, we can view the probability of a latent concept existing within R as being approximated by the probability of the existence of that concept given that the words that we have sampled from the relevance model so far.

Using Bayes Rule, we arrive at the following equation:

$$P(z \mid q_1,...,q_N) = \frac{P(z)P(q_1,...,q_N \mid z)}{P(q_1,...,q_N)} \qquad (34)$$

Using the conditional independence assumption stipulated by the Aspect Model:

$$P(z \mid q_1,...,q_N) = \frac{P(z)\prod_i P(q_i \mid z)}{P(q_1,...,q_N)} \qquad (35)$$

Since $P(q_1,\ldots,q_N)$ will be the same for all z we can effectively disregard this for ranking purposes. This gives us the following equation:

$$P(z \mid R) \approx P(z \mid q_1,...,q_N) \cong P(z)\prod_i P(q_i \mid z) \qquad (36)$$

The PLSA algorithm provides the methodology for the calculation of $P(z)$ and $P(q_i|z)$. Once we complete the Expectation-Maximization computation, we will have arrived at a distribution of concepts which are relevant to the user's information need.

If we use P(z|R) and P(w|z) as ranking mechanisms, we can arrive at a Question Context Model that includes only the most relevant aspects for an information need, and only the words that are most closely related to those aspects. We thus propose that the Aspect-Based Relevance Language Model can serve as the basis for a Question Context model. We discuss in detail the Question Contextualization process in the following chapter, including our efforts to understand the parameters that govern Question Contextualization.

3.3.5 Early Qualitative Results

At this stage, we feel that some of the interesting qualitative results that we have observed in early experiments with the Question Context Model merit some discussion. From inspection, we found that when we ranked words by their P(z|R) and P(w|z) values, the words added to the base questions did, in fact, enhance our understanding of topics and the specific aspect of the topic to which the question related. For example, when the question "What does LPGA stand for?" is processed, the first two words included in the question context are "golf tour." If one is unfamiliar with the LPGA, these words added to the context of the query assist in providing some descriptive text that might help a

human or system "understand" the topic of the question. It is important to note that these semantic associations were made using only statistical information contained within the corpus, without the explicit usage of ontological information, making our methodology transferrable across languages and domains.

Another interesting qualitative observation comes from reviewing the words that are added to the Question Context as we include aspects with lower relevance values. In response to the question "Where was Wolfgang Amadeus Mozart born?" the words "music," "symphony," "opera," "Austria" and "Vienna" are first added to the question context. However, as we increase the window of aspects incorporated into the Question Context, the words "children," "brain," "effect," and "research" are also included. There are clearly two different aspects of Mozart's music that are being represented when these two sets of words are considered. The first set of terms is related to Mozart's music and as his work as a composer; the second set describes the relationship that researchers believe exists between Mozart's music and early brain development in children. While both are valid aspects to associate with Mozart, only one is relevant to the particular question being asked.

A final observation is that in some cases, the answer to the question itself sometimes appeared as a part of the Question Context. For example, in response to the question "Who was the leader deposed by the overthrow of the Pakistani government in 1999?" the word "Sharif" was added to the Question Context. In the general case answers to questions are not included in the Question Context.

Some example Question Contexts from this early work in Question Contextualization is shown in Figure 11.

```
topic: lpga
question: what does lpga stand for?
query context: golf tour two open round year first
last us

topic: pakistani government overthrown in 1999
question: who was the leader deposed by the overthrow?
query context: afghanistan taliban two indian foreign
minister sharif united military government bhutto

topic: wolfgang amadeus mozart
question: where was mozart born?
query context: works piano hall first opera austria
vienna music symphony children brain mozart effect
parents research music orchestra symphony minister eu
foreign Austrian salzburg festival european
```

Figure 11: Example Question Contexts from Early Question Contextualization Experiments with the Aspect-Based Relevance Language Model

In conclusion, we have presented a novel theoretical model called the Aspect-Based Relevance Language Model that can be incorporated into the design of a Question Context model for QA. Our approach is motivated by the fact that questions are typically very sparse, and that the addition of semantic contextual information is required to better "understand" the question being asked. We base our theoretical approach on the PLSA Aspect Model and Relevance-Based Language Models. Our use of statistical techniques in the absence of any domain-specific resources makes our model portable across domains and languages, which is critical for its broad applicability.

4. CANDIDATE DOCUMENT SELECTION

This chapter describes our experiments and results as we apply the Aspect-Based Relevance Language Model to Candidate Document Selection. To this end, we first describe our experiments with the Question Contextualization process, which is an integral part of our experimentation methodology. The output of the Question Contextualization process is our Question Context Model. We then describe two separate approaches to incorporating this Question Context Model into Candidate Document Selection. For each approach, we discuss the foundational algorithms, the experiment methodology and the empirical results.

4.1. Question Contextualization

The primary contribution of the Aspect-Based Relevance Model is that it provides us a mechanism to rank aspects according to their likelihood of probability to a specific information need. PLSA provides us a mechanism to quantify the likelihood of a particular word, given an aspect. Considering these two probabilities in tandem provides us a powerful mechanism for generating a Question Context Model; that is, to generate a set of sense-disambiguated terms that can be considered to provide a context for the user's information need, given the background information provided by the corpus.

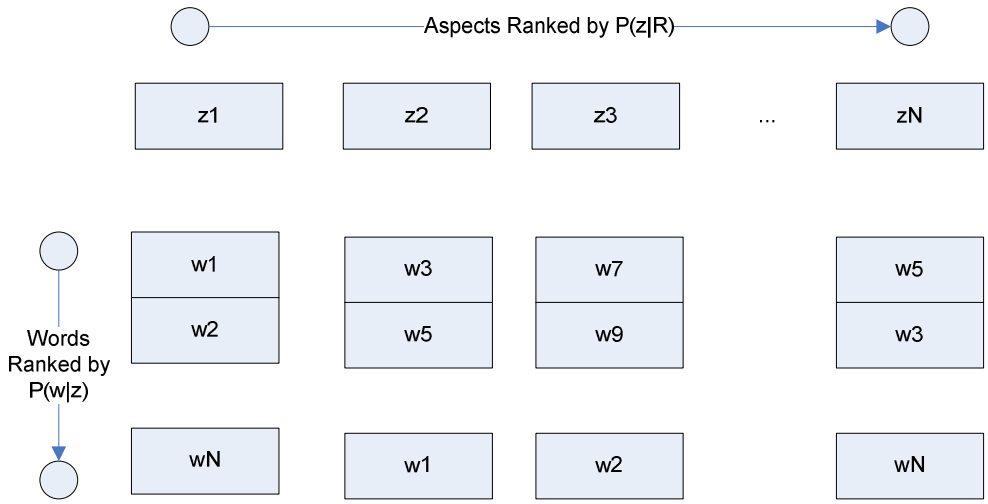

Figure 12: Graphical Representation of the Outcome of the Aspect-Based Relevance Language Model

Recall from Section 2.1 that the effectiveness of a language model can be assessed by the perplexity metric which is defined in Equation (1). Perplexity is an entropy measure; conceptually, perplexity may be thought of as a measure of how "confused" (Jurafsky & Martin, 2000) a language model is. For example, a perplexity of 100 indicates that the language model has to choose uniformly between 100 choices for the next word, given a sequence of proceeding words. As such, the lower the perplexity, the better the language model.

In our experiments with Question Contextualization, our goal was to construct a language model which included the top-N aspects with the highest likelihood of relevance and the top-K most likely words associated with those aspects. In other words, our goal was to understand how we can put thresholds on both P(w|z) and P(z|R) so that the resulting language model had the lowest perplexities. Our first step in this process is to understand how to construct the PLSA Model so that the resulting aspects had minimal perplexities.

The PLSA algorithm is sensitive to both the size and nature of the corpus, as well as the number of aspects (categories) that should be included in the resulting model. The corpus and the number of z-categories are both inputs to the PLSA algorithm. In our experiments we discovered that, like most matrix-based algorithms, PLSA is effectively only able to process relatively small corpuses, given the amount of computing hardware we had available. For example, the Lemur (Ogilvie & Callan, 2001) implementation of PLSA which we used in our experiments was able to handle a maximum corpus of 3000 documents, which is much, much smaller than the full AQUAINT corpus. Furthermore, breaking the full corpus into thousands of aspects may not be useful; since so many concepts and words co-exist with each other, some of the co-occurrence relationships may get muddied if too diverse a document set is used by the algorithm.

To verify this, we conducted an experiment that constructed corpuses of the top-N documents associated with each Question Series target in the TREC 2006 question set, using the Indri (Strohman et al., 2004) search engine for the IR functionality required. We then ran PLSA using a varying number of z-categories, and calculated perplexities for each run. Our results are presented in Table 2.

Table 2: Perplexity of PLSA Language Models Constructed with Varying Corpus Sizes and Z-Categories

	Number of Z-Categories				
Corpus Size	**5**	**10**	**20**	**50**	**100**
5	800	732	623	598	601
10	724	693	642	476	583
30	643	598	512	396	421
50	655	552	401	343	498
100	721	677	543	521	502
500	843	827	815	795	683
1000	948	930	901	896	728

	Number of Z-Categories				
Corpus Size	**5**	**10**	**20**	**50**	**100**
2000	1076	986	923	912	878
3000	1120	1003	956	942	931

These results indicate that the PLSA Model performs best when the number of documents in the corpus is in small, and when the number of z-categories is roughly similar to the number of documents in the corpus. Specifically, we found empirically that corpus sizes of 30 – 50 documents, with 50 or 100 Z-Categories had the best perplexity measures.

However, the resulting perplexities are still very high. Experiments in Answer Validation performed by Rodrigo (Rodrigo et al., 2006) indicate that for factoid Question Answering applications, it is most important to consider those words that are associated with entities. Following that example, we examine the effects of limiting the PLSA language model to include entities alone. We used the LingPipe (http://alias-i.com/lingpipe/) Part of Speech tagger to tag the common and proper nouns in each corpus associated with a TREC Question Series target. We then constructed a pseudo-corpus that was composed of documents containing only common and proper nouns. The perplexities associated with running PLSA on the resulting documents are presented in **Error! Not a valid bookmark self-reference.**.

Table 3: Perplexity of PLSA Language Models Constructed with Varying Corpus Sizes and Z-Categories After Part-of-Speech Tagging

	Number of Z-Categories				
Corpus Size	**5**	**10**	**20**	**50**	**100**
5	772	705	598	574	576
10	697	667	617	454	559

	Number of Z-Categories				
Corpus Size	**5**	**10**	**20**	**50**	**100**
30	618	574	489	376	400
50	629	529	381	324	476
100	694	651	520	498	480
500	814	798	786	767	657
1000	917	900	871	866	701
2000	1043	955	893	882	848
3000	1087	971	925	911	900

These results indicate that the PLSA Model performs better after part-of-speech tagging is performed, and only nouns (common and proper) are included in the documents that are indexed. Again, we see empirically that corpus sizes of 30 – 50 documents, with 50 or 100 Z-Categories had the best perplexity measures.

Now that we have established a set of optimal parameters for the PLSA Aspect Model which is required to calculate the Aspect-Based Relevance Language Model, we turn to the task of determining how to threshold the P(z|R) values so that only the aspects with the highest likelihood of relevance are included in the Question Context. To this end, we use an approach that is sensitive to the query drift (Mitra et al., 1998). Query drift is likely to occur if too many aspects and too many context terms from those aspects are included in the Question Context Model. Based on our earlier experiments, we first fix the corpus size to 50 documents per TREC Question Series target, and the number of Z-Categories is also fixed to 50. We then examine the effects on precision and recall as the top-N aspects (based on the ranking provided by the Aspect-Based Relevance Language Model) are added to the query and the expanded query is sent to the IR engine. The purpose of this experiment was to determine the range of aspects should be

considered in future experimentation. The results of these experiments are shown in Table 4 and Table 5.

Table 4: Precision of TREC 2006 Factoid Questions After Top-N Aspects from the Aspect-Based Relevance Language Model are added to the Raw Query

Number of Documents Returned	Raw Query	Raw Query + Top Aspect	Raw Query + Top 5 Aspects	Raw Query + Top 10 Aspects	Raw Query + Top 20 Aspects
1	0.56	0.61	0.60	0.53	0.50
5	0.07	0.12	0.11	0.04	0.01
10	0.01	0.06	0.05	0.02	0.05
20	0.00	0.00	0.00	0.00	0.00
50	0.00	0.00	0.00	0.00	0.00
100	0.00	0.00	0.00	0.00	0.00

Table 5: Recall of TREC 2006 Factoid Questions After Top-N Aspects from the Aspect-Based Relevance Language Model are added to the Raw Query

Number of Documents Returned	Raw Query	Raw Query + Top Aspect	Raw Query + Top 5 Aspects	Raw Query + Top 10 Aspects	Raw Query + Top 20 Aspects
1	0.00	0.00	0.00	0.00	0.00
5	0.11	0.13	0.12	0.10	0.09
10	0.25	0.29	0.26	0.25	0.24
20	0.32	0.44	0.35	0.31	0.31
50	0.49	0.63	0.50	0.41	0.44
100	0.56	0.72	0.57	0.54	0.54

These initial experiments indicate that both precision and recall benefit when the Top-1 and Top-5 aspects are added to the raw query. After that point, however, query performance degrades as additional aspects are added. As such, for the remainder of our experiments we will consider including only the Top-1 and Top-5 aspects to the query. As a heuristic, we will consider only the top 20 words from each aspect.

4.2. Context-Based Mixture Model

Our first approach to incorporating the Question Context Model into the Candidate Document Selection process is the Context-Based Mixture Model. In the following sections, we first discuss our theory and approach, then introduce our experiment methodology and finally discuss our experimental results.

4.2.1 Theory and Approach

To incorporate the Question Context Model into an IR framework, we view the query as mixture of two parts: a "seen" part which consists of words directly written by the user and an "unseen" part which consists of Context Terms (CTs) which are the words that are part of the query's context. Thus, we can represent the query-likelihood model by the following equation:

$$P(q \mid d) = P(q_{seen} \mid d) + \gamma P(q_{unseen} \mid d) \qquad (37)$$

If γ was set to 1, we would have the case of simple query expansion, in which case the CTs would be directly added to the query without any weighting. q_{unseen} includes $\alpha(t)$ which represents a probabilistic weighting scheme for the Context Terms t that are being added to the query. Using the Aspect-Based Relevance Language Model, we can set:

$$\alpha(t) = \sum_{z=1}^{z=topN} P(z \mid R) * p(t \mid z) \qquad (38)$$

Here, t represents a Context Term (CT) and $topN$ represents the top-ranked aspects that are included in the query. In other words, our confidence that a Context Term t is relevant to a query is determined by how related that term is to a given concept, and how relevant that concept is to a given query. If we use a mixture model approach to generate

the query-likelihood models for $p(w|d)$, we arrive at the following formulae for the seen and unseen parts of the query:

$$P(q_{seen} \mid d) = \sum_{w \in qseen} \lambda p_{mle}(w \mid d) + (1-\lambda) p_{coll}(w) \qquad (39)$$

$$P(q_{unseen} \mid d) = \sum_{t \in qCT} \alpha(t)[\lambda p_{mle}(t \mid d) + (1-\lambda) p_{coll}(t)] \qquad (40)$$

Here, p_{mle} and p_{coll} are the observed word frequency values in the document and collection respectively, and λ is the mixture model parameter. *w* represents a word that appears in the given question (or query), *t* represents a Context Term, and *qCT* represents the set of all CT's included as a part of the query context. Intuitively, using this approach, the final query likelihood model is broken up into four parts:

1. The likelihood that the document generated the query based on the terms which are seen in the query and seen in the document: [$\lambda\ p_{mle}(w|d)$]
2. The likelihood that the document generated the query based on smoothing terms from the collection which are seen in the query: [$(1-\lambda)\ p_{coll}(w)$]
3. The likelihood that the document generated the query based on terms which are seen in the document and assumed to be a part of the query context: [$\alpha(t)\ \lambda\ p_{mle}(t|d)$]
4. The likelihood that the document generated the query based on smoothing terms from the collection which are assumed to be a part of the query context: [$\alpha(t)(1-\lambda)p_{coll}(t)$]

While (1) and (2) are part of the well-established query likelihood approach, terms (3) and (4) are the contribution of this approach and introduce the Question Context Model.

4.2.2 Experimental Methodology

The experimental methodology we used is shown as a block diagram in Figure 13. To validate our approach, we have used factoid questions which were a part of the TREC 2006 Question Answering Track question set. The TREC AQUAINT dataset (E. M. Voorhees & D. K. Harman, 2005) was first pre-processed and indexed. Once the corpus pre-processing was completed, our next task was to generate an Aspect Model for each topic in the question set using PLSA. We used the Lemur implementation (Ogilvie & Callan, 2001) of PLSA in our experiments. As discussed in the previous section, the PLSA models were trained using a corpus of the top 50 documents retrieved for each topic using Indri (Strohman et al., 2005), and we trained our models using 50 z-categories. In addition, we used part-of-speech tagging to limit the words which were provided to PLSA to nouns only.

We use the following strategy to perform Question Contextualization:

- For each topic in the TREC 2006 Question Answering Track question set, obtain a set of candidate documents that represent the knowledge contained in a corpus by using a well-established IR engine.
- For each topic in the question set, determine a set of corpus-specific concepts (i.e., aspects) by running PLSA against the candidate documents collected

- Once we have the sets of words that define corpus-specific concepts for a particular topic from PLSA, we can calculate a distribution that approximates $P(z|R)$ for each question related to that topic by using the model described in Section 3.2.
- For each question, rank the concepts by $P(z|R)$ and the words related to those concepts by $P(w|z)$. Consider those concepts which have the highest values for $P(z|R)$ as potential candidate concepts for inclusion into the Query Context Model
- Once we have obtained a set of top-ranked concepts, we can consider the top-ranked Context Terms within those concepts as Candidate Context Terms (CCTs) to be included as a part of the Question Context.
- Determine the value of $\alpha(t)$ for each CCT t. Use these values to construct the language model for the updated query.

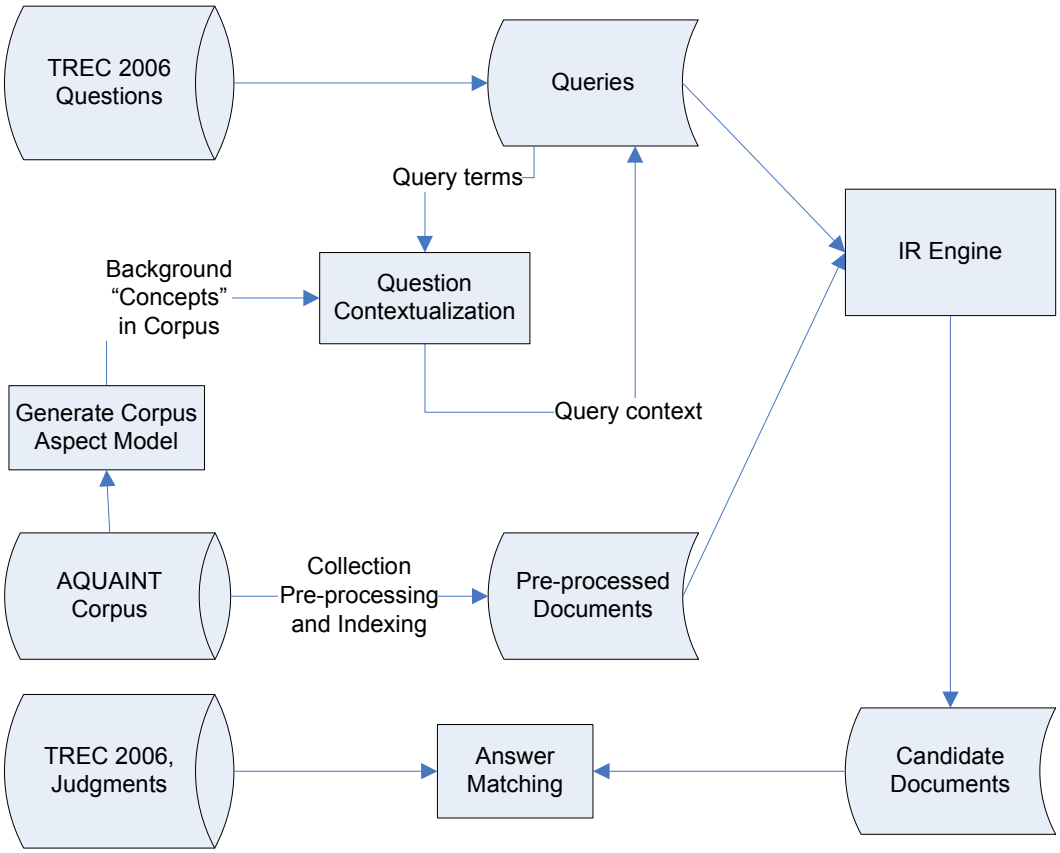

Figure 13: Experiment Methodology for the Context-Based Mixture Model Approach to Incorporate the Question Context Model into Candidate Document Selection

For the purposes of evaluating the effectiveness of our theoretical model, we use the well-known metrics of precision and recall (Manning et al., 2007). It should be noted here that while our initial experiments used the coverage and redundancy metrics, here we revert to precision and recall, which are more standard in the literature. Furthermore, we note here a key distinction between coverage and recall – recall measures the extent to which all correct answers to a question are retrieved, coverage measures the extent to which any answer to a question is retrieved. Since, there were on average between 3-5 globally correct answers for each TREC question, it is easier to achieve high coverage than higher recall. We consider the documents which are relevant to a given question in

the corpus to be only those documents which contain the globally correct answers as given in the TREC 2006 judgments.

4.2.3 Results and Discussion

Our results are presented in Figure 13 and Figure 14. As a baseline for comparison purposes, we used the Kullback-Leibler divergence language modeling strategy (Lafferty & Zhai, 2001). As our methodology uses a feedback approach, we compare our results to both the base KL-divergence with blind feedback using a mixture model approach. This is appropriate as this approach is typically used as the baseline for smoothing approaches (Murdock & Croft, 2004). We used the Lemur (Ogilvie & Callan, 2001) implementation of KL-divergence for our baseline metrics.

Figure 13 and Figure 14 show the precision and recall results of our approach against the baseline when the top 1, 5, 10, and 20 aspects are included as a part of the Query Context Model. As can be seen, our results are promising: when the top 1 and top 5 aspects are included as a part of the Query Context Model, both precision and recall results compare favorably against the baseline language modeling retrieval methods. However, as the number of aspects included in the Question Context Model increased, the precision and recall of our queries decreased dramatically. Intuitively, this makes sense because as we broadened the amount of information contained in the Query Context model, we increased the likelihood that information potentially non-relevant to the query be included. These results also validate our model at a very basic level; in other words, our ranking of aspects by $P(z|R)$ does accurately reflect the ordering of aspects in order of relevance to a query

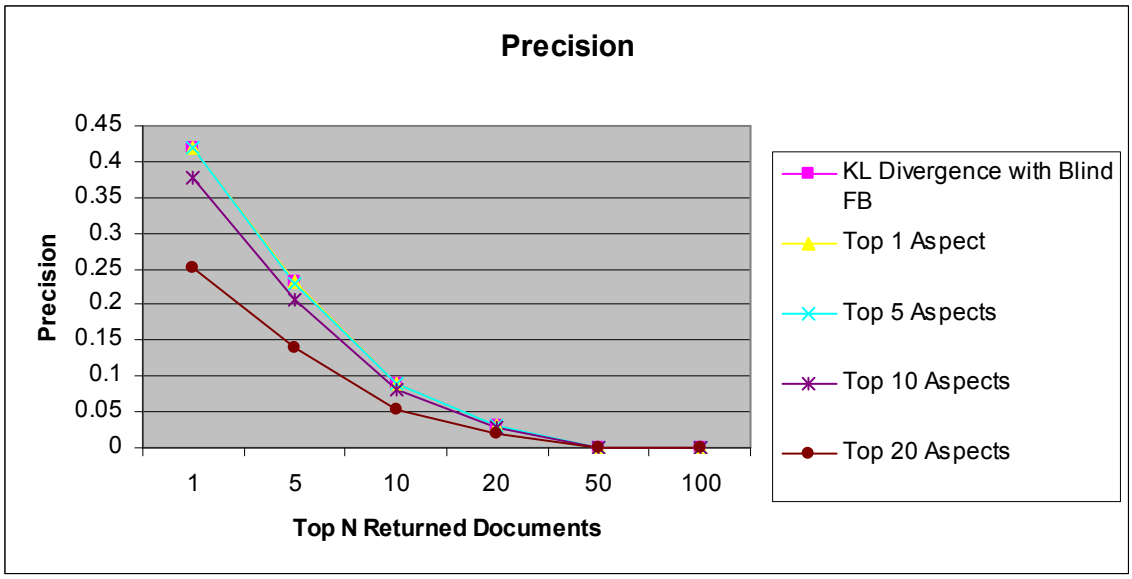

Figure 14: Precision Results for the Context-Based Mixture Model Approach

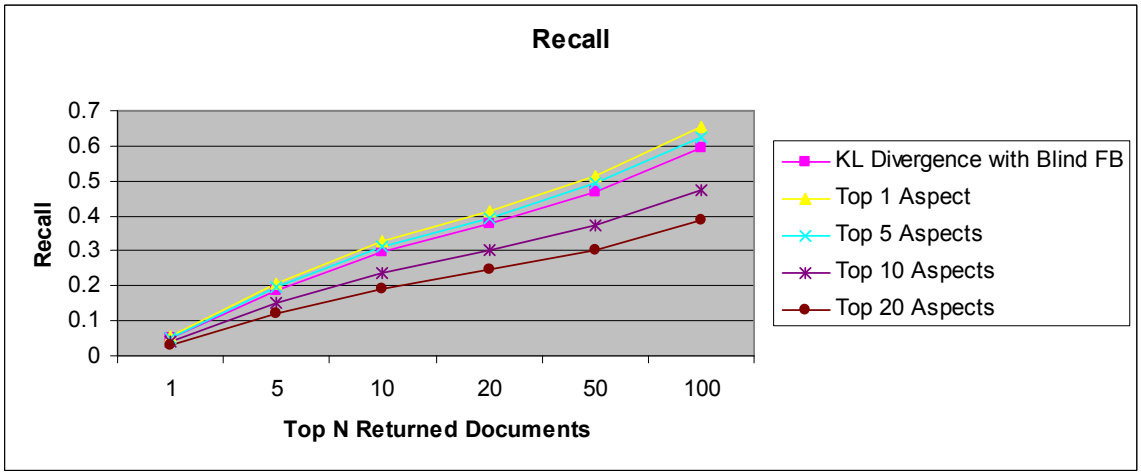

Figure 15: Recall Results for the Context-Based Mixture Model Approach

4.3. Context-Based Query Expansion and Context-Based Smoothing

Our second approach to incorporating the Question Context Model into the Candidate Document Selection process is the Context-Based Query Expansion and Context-Based Smoothing. In the following sections, we first discuss our theory and

approach, then introduce our experiment methodology and finally discuss our experimental results.

4.3.1 Theory and Approach

To incorporate the Question Context Model (QCM) into an IR framework, we consider that there are essentially two sides to the "zero-frequency" (Witten & Bell, 1991) problem : (1) A query input to a QA system rarely contains all of the words that would be relevant to the information need that the query represents; and (2) A document rarely contains all of the words that are related to the information content of the query. To examine the effectiveness of the proposed Query Context Model (QCM), we need to understand its effectiveness when applied to both facets of the "zero-frequency" problem: context-based query expansion and context-based document smoothing.

First, we consider using a query expansion approach. In this case, the Aspect-Based Relevance Language Model, *P(z|R)*, provides a way to quantify the relevance of an aspect to a query. PLSA also provides a posterior probability *P(w|z)*, which quantifies the relationship between an aspect and an individual Context Term associated with the aspect. We define *Candidate Context Terms* (CCTs) by first ranking CTs based on a posterior probability *P(w|z)*, and then selecting the top-ranked CTs. (This is the same definition as that provided in the previous section when discussing the Question Contextualization process.) We can define a weight, $\alpha(w)$, that defines the relevance of an individual Candidate Context Term (*CCT*) to a query as follows:

$$\alpha(CCT) = P(z \mid R)P(CCT \mid z) \qquad (41)$$

Here, we approach query expansion by weighting each CCT that is added to the query by this weight *α(CCT)*. The final query submitted to the IR system is a composite of the original query with a term weight of 1 with the CCTs weighted with *α(CCT)*.

When considering the flip side of the "zero frequency" problem, we propose a linear interpolation approach to document smoothing. The intuition behind our approach is that a document which contains CCTs should have a higher likelihood of generating the query. We propose to interpolate a document model $P_M(w|d)$ with information from the Question Context Model (QCM). Mathematically, we represent the interpolation approach as follows:

$$P(w \mid d) = (1 - P_{avg}(z \mid R)) * P_M(w \mid d) + P_{avg}(z \mid R) * P_{norm}(w \mid QCM) \qquad (42)$$

Here, $P_{avg}(z|R)$ denotes the average relevance of the aspects that are included as a part of the Question Context Model. In the simple case where only the top-ranked aspect is included as a part of the Question Context, this reduces to the highest $P(z|R)$ value for the query.

The term $P_{norm}(w|QCM)$ denotes a normalized $P(w|z)$ measure for each word in the Question Context Model:

$$P_{norm}(w \mid QCM) = \frac{\sum_{\substack{w \in CCT, \\ z \in QCM}} P(w \mid z)}{\sum_{\substack{w' \in CT, \\ z \in QCM}} P(w' \mid z)} \qquad (43)$$

Since the Question Context Model contains only CCTs in the top-*k* ranked aspects, it does not include every aspect that is part of the original PLSA model, nor every word that was associated with those aspects. Therefore, we must normalize the $P(w|z)$ values that

were extracted from PLSA if our model is to remain a probability distribution. We normalize the $P(w|z)$ values by summing $P(w|z)$ for all aspects in the QCM which include the Context Term (CT) w, and then dividing this by the sum $P(w'|z)$ of all words w' which are included as Candidate Context Terms (CCT) for aspects that are included as a part of the QCM.

4.3.2 Experimental Methodology

The experimental methodology we used is shown as a block diagram in Figure 16. To validate our approach, we have used a random set of 100 factoid questions from the Text Retrieval Conference (TREC) 2006 Question Answering Track question set. Each TREC factoid question is associated with a question topic. In the TREC 2006 question set, there are 75 topics, each with approximately 5 factoid questions associated with it, for a total of 403 factoid questions. These questions were processed against the TREC AQUAINT dataset. The first step of our experimentation was to index and pre-process the AQUAINT dataset using some standard techniques such as stemming and stopword removal.

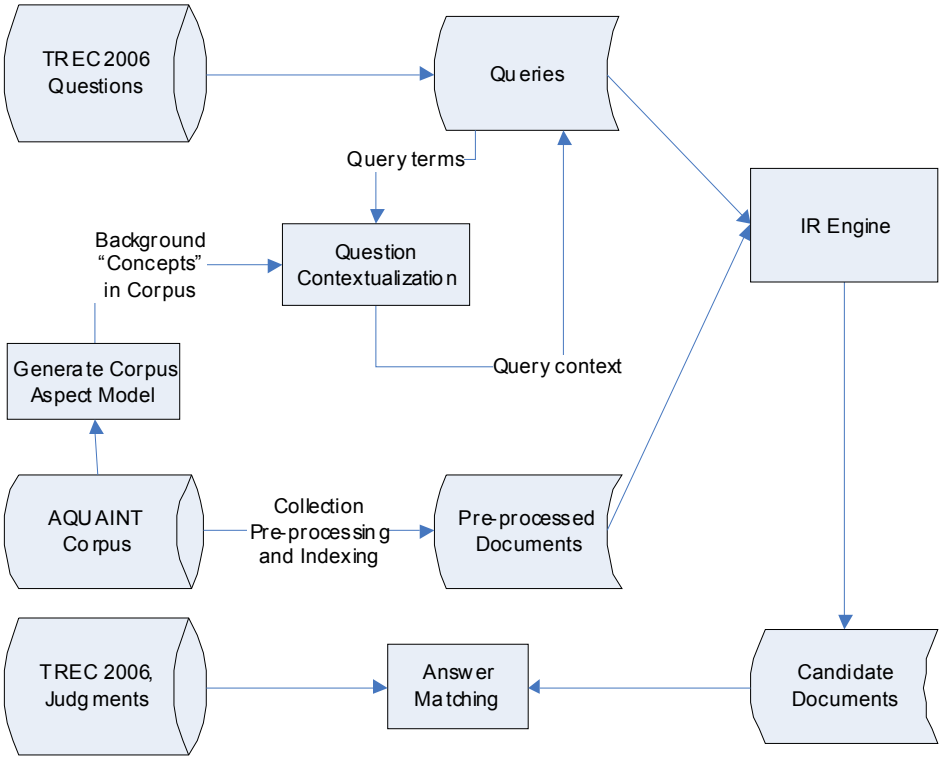

Figure 16: Experiment Methodology for the Context-Based Smoothing and Context-Based Query Expansion Approaches to Candidate Document Selection

We then used the following strategy to perform Question Contextualization:

- For each topic in the TREC 2006 Question Answering Track question set, we obtained a set of 50 top-ranked documents using Indri (Strohman, 2005) to serve as the training documents for PLSA. We determined empirically that 50 documents yielded the most effective training model for our purposes.

- For each topic in the question set, we determined a set of corpus-specific concepts (i.e., aspects) by running PLSA against the candidate documents collected. Our PLSA model was trained using 50 z-categories. In addition, we used part-of-speech tagging to limit the words to include only nouns (including proper nouns). We used the Lemur implementation (Ogilvie, 2001) of PLSA in our experiments.

- We then calculated the distribution that approximates *P(z|R)* for each question related to that topic by using the Aspect-Based Relevance Model approach proposed in this paper.
- For each question, we ranked the aspects (or concepts) by *P(z|R)* and their associated CTs by *P(w|z)*. We considered only those aspects which had the highest values for *P(z|R)* as potential candidate concepts for inclusion into the QCM.

Once we have obtained a set of top-ranked concepts (or aspects), we can consider the words within those concepts with the highest posterior probabilities as CCTs to be included as a part of the Question Context.

Once created, the Question Context Model is then incorporated into the QA framework using both the Approach 1 and Approach 2 methodologies described earlier. We consider the documents which are relevant to a given question to be only those documents which contain the correct answers as given by the TREC 2006 judgments (eg. we only consider a documents which contain globally correct answers).

4.3.3 Results and Discussion

We used the Ponte query expansion approach which is based on a language modeling approach for our query expansion methodology. This method is appropriate to use as a baseline as it employs a query expansion approach based on the top-*N* retrieved documents. Language modeling including Ponte expansion has been used as a baseline by other query modeling approaches (Balog, Weerkamp, & de Rijke, 2008).

Figure 17 shows the results of our query expansion methodology using a Question Context that includes the top 20 words from the top-5 aspects. Our results show the following:

- A 9.4% improvement in recall performance when the Question Context includes the top-1 aspect, at 50 documents retrieved
- A 9.4% improvement in recall performance when the Question Context includes the top-1 aspect, at 100 documents retrieved
- A 9.3% improvement in recall performance when the Question Context includes the top-5 aspects, at 100 documents retrieved

Using the Wilcoxon signed rank test, we determined that these results are statistically significant ($p < 0.05$) from the baseline approach.

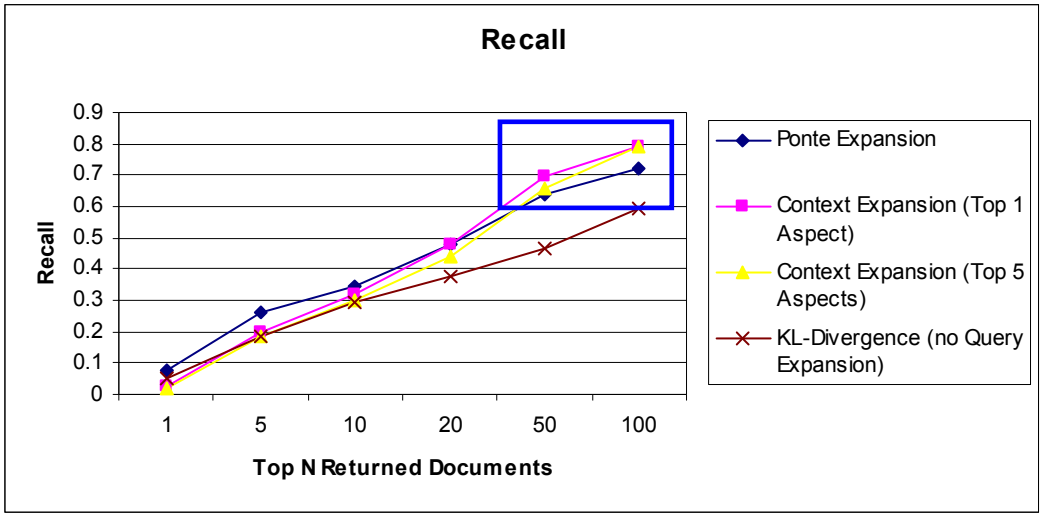

Figure 17: Recall of Context Based Query Expansion vs. Ponte Query Expansion

These results show that our methodology shows significant improvement in recall. Improvements in recall are critically important for Information Retrieval

applications that feed a Question Answering system. In effect, by increasing the recall of relevant (answer-bearing) documents we are providing better opportunities for the second stage of QA system to find the correct answer. It should be noted that using our system, we have a recall of 80% at 100 documents returned when the Question Context contains the top-1 aspect. This means that 80% of the answer-bearing documents are returned in the top-100 results (i.e., documents) that are submitted to the second stage of QA for processing.

For the Query Context-based document smoothing approach, we compared our smoothing approach against the baseline Kullback-Leibler (KL) divergence approach with blind feedback. This is appropriate as this approach is typically used as the baseline for smoothing approaches (Murdock & Croft, 2004). The results of this methodology are shown in Figure 18. The results of this approach were less conclusive. Using the Wilcoxon signed rank test, no significant difference in behavior ($p < 0.05$) was observed between Context-based smoothing and the baseline approach.

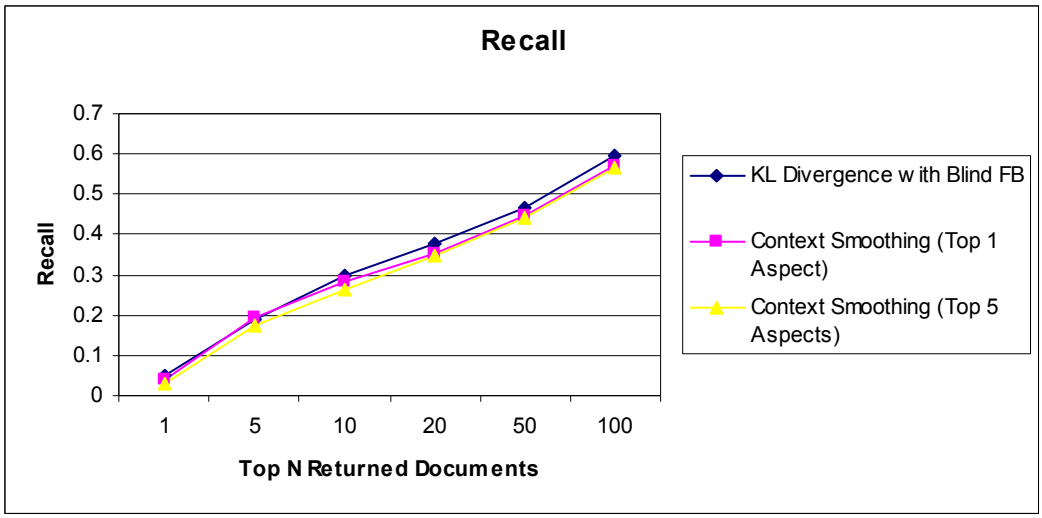

Figure 18: Recall of Context Based Smoothing vs. KL-Divergence with Blind Feedback

We do not include precision results for either of the two methods as these results do not show any significant differences (either positive or negative) from the baseline approaches.

To better understand the results of our methodology for both query expansion and document smoothing, it is useful to consider some examples. When the question "What does LPGA stand for?" is processed, the first two words included in the Question Context are "golf tour." Similarly, when the question "What tobacco company sponsors the Winston Cup Series?" is processed, the first three words to be added to the Question Context are "car," "NASCAR," and "racing." Intuitively, these words enhance the description of the query's information need. Thus, it seems reasonable that a query expansion technique which includes Candidate Context Terms (CCTs) will have a higher likelihood of finding relevant documents than by the use of the terms in the query alone. When document smoothing is considered, on the other hand, we find that many documents that include the term "LPGA" also include the term "golf" and "tour." Our model adds minimal value because the likelihood that a document that includes the term

"LPGA" would generate the query is similar to the likelihood that a document that includes the terms "LPGA," "golf" and "tour" would generate the query. We believe that this is a characteristic of newswire data, which is typically contains articles focused around a tight topic area. Our future efforts should include investigation into domains which do not share this characteristic.

In conclusion, we have formally presented a novel approach that uses statistical language modeling methods to create a Question Context Model (QCM). We then incorporated the QCM into the IR stage of QA using two approaches, one of which shows significant improvements in recall. This improvement in recall is critical to Question Answering, as it provides more opportunities for the second stage of the Question Answering system to extract and collect the correct answers.

5. ANSWER VALIDATION

This chapter describes our experiments and results as we apply the Aspect-Based Relevance Language Model and our Question Context Model to Answer Validation. To this end, we first motivate a novel concept called Answer Credibility, which is an integral part of our approach and methodology. As a part of this discussion, we present a brief summary of credibility in the computational sciences. We then describe two separate approaches to modeling Answer Credibility and incorporating our model into the Answer Validation process: a Naïve Approach and the Perspective Similarity Approach. For each approach, we discuss the foundational algorithms, the experiment methodology and the empirical results. We conclude the chapter with an in-depth view of the parameter estimation problem associated with our model for Answer Credibility.

5.1 Answer Credibility: Motivation

In order to incorporate Question Context into the second stage of Question Answering, we consider the relationship between Question Context and answers. We define Context Terms (CTs) to be words which are part of the Question Context, and related to an aspect z that is included in the Question Context by a probability $P(CT|z)$. Empirically, we can review the words that are added to the Question Context as we add CTs from aspects that are ranked by their $P(z|R)$ values. For example, in response to the question "Where was Wolfgang Amadeus Mozart born?" the words "music," "symphony," "opera," "Austria" and "Vienna" are added to the question context from the top-ranked aspect. We can see that the most correct answer to this question, "Salzburg"

is not one of the CTs that are included in the Question Context, but we can see that some CTs ("Austria", "Vienna") are directly related to the correct answer. Ideally, an Answer Context is a set of terms (and their associated probabilities) that are semantically related to the correct answer to a question. Our observations lead us to conclude that Question Context is not the same as Answer Context; if the two were one and the same the Question Context would include all CTs related to "Salzburg," not just those related to Wolfgang Amadeus Mozart and Salzburg. For example, "Salzburg" is the correct answer to the question "What was the setting for the film *Sound of Music*?" If the Question Context was exactly the same as the Answer Context, it might include terms related to *Sound of Music*.

Our observations lead us to conclude that Question Context is not the same as Answer Context. To this end, we believe that there is an overlap between Question Context and Answer Context. We define the region that Question Context overlaps Answer Context as the *Answer Perspective* associated with the question, and this relationship is depicted in Figure 19.

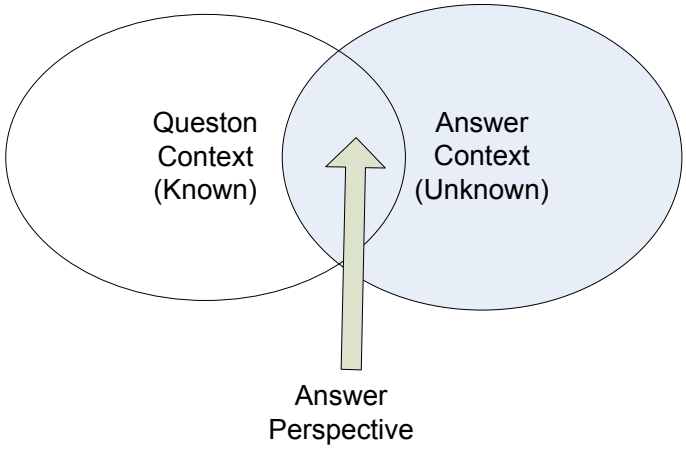

Figure 19: Graphical Representation of Question Context, Answer Context and Answer Perspective

Our approach to Answer Validation seeks to answer the following question: what is the strength of the relationship of a source document from which a candidate answer was derived to the Answer Perspective (eg. the overlap between the Question Context and the Answer Context)? If the relationship is strong, it stands to reason that the believability of the candidate answer is high. If the relationship is weak, we have less evidence that the candidate answer is valid. Our approach to Answer Validation is motivated by the desire to quantify the strength of the relationship between a source document and Answer Perspective; this relationship is what we call Answer Credibility.

5.1.2 Credibility in the Computational Sciences

Credibility has been extensively studied in the field of information science (Metzger, 2007). Credibility in the computational sciences has been characterized as being synonymous with believability (Fogg & Tseng, 1999), and has been broken down into the dimensions of trustworthiness and expertise in the human-computer interaction community. Fogg (Fogg & Tseng, 1999) hypothesizes that there are four different

aspects of a computer product, each of which can be assessed in terms of its credibility to the end user: (1) the device, (2) the interface, (3) the functionality, and (4) the information. A brief description of each of these aspects of credibility follows:

- Device credibility relates to the physical aspect of the computing product. For example, device credibility is related to the form of the device, its physical structure and appearance, its weight and its ruggedness.
- Interface credibility relates to the display of the computer product as well as to the interaction experience. Interface credibility is primarily related to what the user can see and/or touch when providing inputs to and receiving outputs from the device.
- Functional credibility relates to what a computer product does and how it is done. Functional credibility is closely related to the user's trust in the computational capabilities of the device.
- And finally, Information credibility relates to how believable the information is from the computing product.

Fogg (Fogg & Tseng, 1999) argues that the design of any model of credibility must be considered in light of these four categories. To this end, we categorize our model of credibility as falling squarely into the last category of Information credibility.

A very different approach is Meola's (Meola, 2004) contextual model of Web information evaluation. "Meola critiqued the idea of using checklists to evaluate online information because he believes they are unwieldy for users to perform (e.g., one checklist requires Internet users to answer over 112 questions per Web site viewed), and

thus are an unrealistic approach to teaching critical evaluation (Metzger, 2007)." Meola's contextual model focuses on information external to a particular site. By using external information to establish credibility, Meola (Meola, 2004) contends that online information "is located within its wider social context, facilitating reasoned judgments of information quality." Meola's approach recommends three techniques to determine the quality of online information:

- The first technique is promoting peer and editorially reviewed resources that are available online. Information intermediaries should inform Internet users of high-quality, vetted resources.
- The second technique is comparing information found on a Web site to other Web sites and/or to offline sources.
- The final technique is corroboration. Similar to comparison, corroboration involves seeking more than one source to verify information on a given topic. Users may assess the accuracy and reliability of information through corroboration as a convergence of facts and/or opinions from a variety of sources.

Meola argues that the contextual approach to online information evaluation is more practical than are traditional checklist approaches, and thus will be more likely to be used by online information seekers. It should be noted here that the last two techniques can be thought of conceptually as Answer Validation approaches, specifically when one considers statistical methods of Answer Validation such as the approach proposed by Magnini (Magnini, Negri, Pervete et al., 2002; Magnini, Negri et al., 2002a). Both of the

approaches to Answer Credibility presented in the following sections can be thought of as following as falling into the latter two classes of techniques.

5.2 Answer Credibility: Naïve Approach

Our first approach to modeling Answer Credibility and incorporating our model into the Answer Validation process is a Naïve Approach. In the following sections, we first discuss our theory and approach, then introduce our experiment methodology and finally discuss our experimental results. We also present a brief discussion of the OpenEphyra Question Answering system which was used as a framework for our experiments.

5.2.1 Theory and Approach

Our mathematical representation of Answer Credibility attempts to quantify the reliability of a source using the semantic Question Context. We define Answer Credibility to be a similarity measure between the semantic Question Context built via the Aspect-Based Relevance Language Model and the source document from which the answer was derived. As we derive a statistical model of Question Context, our model of Answer Credibility is likewise a statistical one.

From the Aspect-Based Relevance Language Model, we can view the Question Context (or Answer Perspective) as a document, having a document language model. Since the Aspect-Based Relevance Language Model provides us a mechanism to rank the aspects associated with a particular question, we include only the top-N most relevant aspects in the Question Context to circumvent semantic drift. Moreover, since we can

rank the words within each aspect by their posterior probability, *p(w|z)*, an we include only the top-*N* words, called Context Terms, associated with each relevant aspect in the Question Context. To ensure that the resulting model remains a probability distribution (i.e. sums to 1 over all *w*), however, we must perform some normalization.

Mathematically, we can represent this as $P_{norm}(w|QC)$, which denotes a normalized posterior probability for each word in the Question Context:

$$P_{norm}(w \mid QC) = \frac{\sum_{z \in QC} P(w \mid z)}{\sum_{z \in QC} \sum_{\mathrm{w'} \in \mathrm{CT}} P(\mathrm{w'} \mid \mathrm{z})} \qquad (44)$$

We normalize by summing P(w|z) for all aspects in the Question Context (QC) which include the word w, and then dividing this by the sum of P(w'|z) of all words w' which are included in the set of Context Terms (CT) for aspects that are included in QC.

Once this document language model for the Question Context is generated, we can use the well-known Kullback-Leibler divergence method (Lafferty, 2001) to compute the similarity between the Question Context document model and the document model for a document containing a candidate answer. This similarity measure is what we call the Answer Credibility:

$$AnswerCredibility = \sum_{w} P_{norm}(w \mid QC) \log \frac{P_{norm}(w \mid QC)}{P(w \mid d)} \qquad (45)$$

Here, P(w|d) is the document language model for a document containing a candidate answer. We propose that Answer Credibility is not an absolute metric which by itself can be used to determine the correctness or incorrectness of an answer in a QA system. The

nature of QA is such that there are numerous factors which are required to determine the right answer. The majority of these are, by their very nature, rooted in Natural Language Processing and Information Extraction. We do, however, believe that Answer Credibility should be an influencer of a QA system's answer score. We propose an interpolation technique that modulates an answer's score during the Answer Selection process using Answer Credibility. This interpolation method is given mathematically in the equation below:

$$score' = (1-\lambda) * score + \lambda * AnswerCredibility \qquad (46)$$

Here, *score* represents the answer score prior to interpolation with Answer Credibility, *score'* represents updated score value after interpolation, and λ represents an interpolation constant.

In our model, we set λ using *P(z|R)* distribution which is the outcome of the Aspect-Based Relevance Language Model. Fundamentally, *P(z|R)* describes the likelihood that a given aspect *z* is relevant to an information need given by a question. If we average the values of *P(z|R)* for all aspects which are included in the Question Context, we have a measure that describes how relevant the Question Context is to an information need. Since the Answer Credibility measures the similarity between a Question Context and a given document, we propose that its influence on the final answer score should be modulated by the *P(z|R)* measure of relevance provided by the Aspect-Based Relevance Language Model.

5.2.2 The OpenEphyra Question Answering System

Despite our early implementation efforts done as a part of the 2007 TREC Question Answering exercise described in Chapter 3, we found that in order to effectively conduct evaluate our Answer Validation methodology, we needed a proven Question Answering framework with a robust Answer Extraction capability. Our framework produced very low precision and recall on the TREC 2007 QA testset. In addition, we did not fully develop an Answer Extraction methodology, since Answer Extraction was not a focus area for our research. However, the Answer Validation process uses Answer Extraction results; the evaluation of our methodology would be meaningless if a well-developed Answer Extraction process did not fuel our Answer Validation algorithm.

Given this situation, we make use of the OpenEphyra framework which is introduced in (Schlaefer, Gieselmann, Schaaf, & A., 2006) to provide an Answer Extraction mechanism. OpenEphyra is an extensible Question Answering framework that was released into the open-source domain in February 2008 by Carnegie Mellon University. Ephyra (prior to release in the open source domain) had competed in several TREC QA exercises; the precision and recall results produced by Ephyra in 2007 placed it in the upper third of the competing systems in terms of factoid accuracy. As such, we can assume that the Answer Extraction component of Ephyra is well-developed and provides us a suitable platform on which to base our experiments.

Ephyra is organized as a pipeline composed of standardized components for query generation, answer extraction and answer selection. The components of the pipeline (called filters) can be combined and arranged arbitrarily, and different configurations can be used for different question types. This architecture facilitates experimenting with

various setups and allows integrating multiple Question Answering techniques into a single system. To incorporate a new technique for Answer Validation, one can simply plug in an additional filter into the pipeline. A filter eliminates inappropriate results and creates new results from existing ones (e.g. by extracting factoid answers from a text snippets) or rescores the results according to a specific feature. At each stage of the pipeline, the output is a ranked list of answers, each of which is associated with a source document and a score.

The following filters are provided by OpenEphyra to support processing of TREC QA question series. These filters are presented in the order in which they are processed as a part of the filter pipeline.

- Answer Type Filter: Extracts factoid answers of the expected answer types from text passages. This part of the filtering process is high recall, but low precision.
- Answer Pattern Filter: Uses known (pre-determined) answer patterns to extract factoid answers from text passages and to rank them. It is only applied if the question could be interpreted during the Question Analysis phase.
- Filter Predicate Extraction Filter: Extracts predicates which are similar to those in the question from documents in the corpus.
- Truncation Filter: This filter truncates the answer strings. It drops the following prefixes and suffixes: blanks and some special characters, articles, and prepositions. After truncation, similar answers are merged.

- Stopword Filter: Drops a result if the answer string contains only function words ("by", "for", "to"), single characters (except digits), an interrogative ("who", "what"), a single bracket, quotation mark or is an adverb.
- Question Keywords Filter: Drops a result if the answer string contains a keyword from the question. This is useful for eliminating answers that simply repeat the question.
- Score Normalization Filter: A filter that normalizes the scores of the answer candidates by applying a trained classifier. This classifier is trained on data that matches words to the category of data that they are associated with.
- Score Combination Filter: Combines and normalizes the score of a single answer that might have been extracted using different techniques. The data going into this filter is an un-normalized list of answer – this filter performs the data (as well as the score) normalization process.
- Factoid Subset Filter: Checks a set of factoid answers for subset relations. If a factoid answer is a subset of another factoid answer (i.e. its tokens are a subset of the tokens of another answer), then the former is dropped and its score is transferred to the latter. This filter ensures that the answers returned are as precise as possible.
- Duplicate Filter: Drops duplicate results. Results are considered equal if the normalized form of two answer strings is the same. The result with the higher score is kept.

- Score Sorter Filter: Sorts the results by their scores in descending order. The sort is guaranteed to be stable.

The output of the Duplicate Filter can be considered to be the final result of Ephyra's Answer Extraction processing. The Score Sorter Filter is a computational process that simply orders the final set of answers by their scores; we do not consider this to be a part of the core Answer Extraction mechanism.

5.2.3 Experiment Methodology

The experimental methodology we used is shown as a block diagram in Figure 20. To evaluation our approach, we have used factoid questions which from the Text Retrieval Conference (TREC) 2006 Question Answering Track question set. These questions were processed against the TREC AQUAINT dataset (E. M. Voorhees & D. Harman, 2005). The first step of our experimentation was to index and pre-process the AQUAINT dataset using some standard techniques such as stemming and stopword removal.

We then used the following strategy to perform Question Contextualization:

- For each topic in the TREC 2006 Question Answering Track question set, we obtained a set of 50 top-ranked documents using Indri to serve as the training documents for PLSA. For each topic in the question set, we determined a set of corpus-specific concepts (i.e., aspects) by running PLSA against the candidate documents collected. Our PLSA model was trained using 50 z-categories. In addition, we used part-of-speech tagging to limit the words which were included as a part of the PLSA training documents to include only nouns (including proper

nouns). We used the Lemur implementation (Ogilvie & Callan, 2001) of PLSA in our experiments.

- We then calculated the distribution that approximates *P(z|R)* for each question related to that topic by using the approach described in this paper.
- For each question, we ranked the concepts by *P(z|R)* and the words related to those concepts by *P(w|z)*. We considered only those aspects which had the highest values for *P(z|R)* as potential candidate concepts for inclusion into the Query Context Model.

Once we have obtained a set of top-ranked concepts, we can consider the words within those concepts with the highest posterior probabilities as candidate CTs to be included as a part of the Question Context.

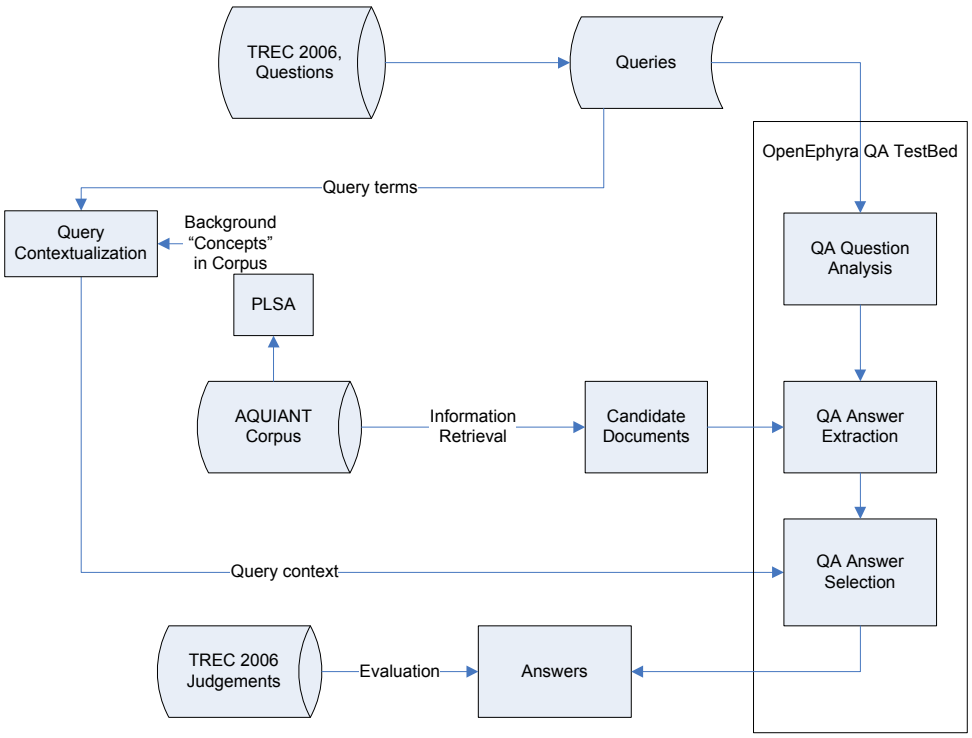

Figure 20: Experiment Methodology for Answer Validation Using the Naïve Approach to Modeling Answer Credibility

The OpenEphyra Question Answering framework (Schlaefer et al., 2006) was then used as the framework for our Answer Credibility implementation. Specifically, we inserted an Answer Credibility filter that implements the algorithm described in section 5.2.1 of this document as a step in the OpenEphyra filtering pipeline.

For the purposes of evaluating the effectiveness of our theoretical model, we use the accuracy and Mean Reciprocal Rank (MRR) metrics (E. M. Voorhees & D. Harman, 2005). Accuracy is a simple metric that is used to evaluate results for TREC factoid questions. An answer has an accuracy score of 1 if both the answer text as well as the document from which the answer is derived is correctly identified by the system. Otherwise, the answer has an accuracy of 0. MRR evaluates the rank at which the QA system returns the correct answer. MRR increases as the score of the correct answer increases in relationship to scores of incorrect answers. Mathematically, MRR is defined as follows:

$$MRR = \frac{1}{|N|}\sum_{i=1}^{N}\frac{1}{rank_i} \qquad (47)$$

Here, N is the set of questions being evaluated and $rank_i$ is the rank of the correct answer to question i. MRR is a meaningful measure for our methodology as it highlights any changes in answer ranking from the baseline, which is of critical importance when evaluating the effectiveness of Answer Selection methodologies.

The TREC 2006 judgments include a set of correct answers and documents associated with those answers for each question. We use these evaluation results to determine both answer accuracy and MRR. We consider an answer to be correct only if it is marked globally correct in the TREC QA judgments.

5.2.4 Results and Discussion

We compared the results of our approach against the baseline OpenEphyra Question Answering system. Our methodology evaluates the increase in MRR and accuracy that is achieved by adding out Answer Validation strategy on top of the baseline system that contains no specific Answer Validation components. Our results are presented in Table 6 and Table 7

Table 6: Average Accuracy of Baseline vs. Baseline Including Answer Credibility Using the Naive Approach

Question Category	Question Count	Baseline Accuracy	Baseline + Answer Credibility Accuracy
How	20	0.25	0.20
how many	58	0.12	0.07
how much	6	0.00	0.00
in what	47	0.64	0.55
What	114	0.23	0.28
what is	28	0.18	0.18
When	29	0.21	0.10
Where	23	0.30	0.30
Where is	6	0.33	0.33
Which	17	0.29	0.18
Who	17	0.47	0.59
who is	14	0.57	0.71
who was	24	0.38	0.50

To facilitate interpretation of our results, we sub-divided the set of factoid questions into categories by their question words, following the example of Murdock (Murdock, 2006). The shaded cells in both tables indicate categories for which improvements were observed. The paired Wilcoxon signed-rank test was used to measure significance in improvements for MRR; the shaded cells in Table 7 indicate results for which the results were significant ($p<0.05$). Due to the binary nature of the results for accuracy at the

question level, a similar significance test was found to be inappropriate at the question level.

Our results show the following:

- A 5% improvement in accuracy over the baseline for "what"-type questions.
- An overall improvement of 13% in accuracy for "who"-type questions, which include the "who," "who is" and "who was" categories
- A 9% improvements in MRR for "what" type questions
- An overall improvement of 25% in MRR for "who"-type questions, which include the "who," "who is" and "who was" categories

Table 7: Average MRR of Baseline vs. Baseline Including Answer Credibility Using the Naive Approach

Question Category	Question Count	Baseline MRR	Baseline + Answer Credibility MRR
How	20	0.33	0.28
how many	58	0.21	0.16
how much	6	0.08	0.02
in what	47	0.68	0.60
What	114	0.30	0.33
what is	28	0.26	0.26
When	29	0.30	0.19
Where	23	0.37	0.37
where is	6	0.40	0.40
Which	17	0.38	0.26
Who	17	0.51	0.63
who is	14	0.60	0.74
who was	24	0.43	0.55

While our methodology does show a slight improvement in MRR over all categories of questions over the baseline, this improvement was not significant at the $p <$

0.05 level. In general, our methodology showed a positive effect on the results of the Answer Selection process in terms of both accuracy and MRR for "who" and "what" type questions.

5.3 Answer Credibility: Perspective Similarity Approach

Our second approach to modeling Answer Credibility and incorporating our model into the Answer Validation process is the Perspective Similarity Approach. In the following sections, we first discuss our theory and approach, then introduce our experiment methodology and finally discuss our experimental results.

5.3.1 Theory and Approach

From the Aspect-Based Relevance Language Model, we can view the Question Context Model (QCM) as a document, having a document language model. Since the Aspect-Based Relevance Language Model provides us a mechanism to rank the aspects associated with a particular question, we include only the top-*N* most relevant aspects in the Question Context Model to circumvent semantic drift.

Mathematically, we represent the Question Context Model as a distribution $P_{norm}(w|QCM)$, which denotes a normalized posterior probability for each word in the Question Context Model:

$$P_{norm}(w \mid QCM) = \frac{\sum_{\substack{w \in CCT, \\ z \in QCM}} P(w \mid z)}{\sum_{\substack{w' \in CT, \\ z \in QCM}} P(w' \mid z)} \qquad (48)$$

Since the Question Context Model contains only CCTs in the top-*k* ranked aspects, it does not include every aspect that is part of the original PLSA model, nor every word that was associated with those aspects. Therefore, we must normalize the *P(w|z)* values that were extracted from PLSA if our model is to remain a probability distribution. We normalize the *P(w|z)* values by summing *P(w|z)* for all aspects in the QCM which include the Candidate Context Term (CCT), *w*, and then dividing this by the sum *P(w'|z)* of all words *w'* which are included as Context Terms (CT) for aspects that are included as a part of the QCM.

Once this document language model for the Question Context is generated, we can turn our attention to developing a model for Answer Context and Answer Perspective. To this end, we propose that we can build an *Answer Topic Model* from the top-*N* documents associated with the candidate answer using a document modeling approach (Lafferty, 2001). Since the query used to build the Answer Topic Model includes only the terms in the candidate answer, we can assume that all aspects associated with the candidate answer are included in the Answer Topic Model. As such, we propose that the Answer Topic Model is an approximation of the Answer Context Model (ACM), for the candidate answer.

At this point, we can use the well-known Kullback-Leibler divergence method (Lafferty & Zhai, 2001) to compute the similarity between the Question Context Model and the Answer Context Model. We define this similarity as the *Perspective Similarity* between the Question Context Model and a candidate Answer Context Model. *Perspective Similarity* (PS) is defined mathematically by the following equation:

$$PS = \sum_{w} P_{norm}(w \mid QCM) \log \frac{P_{norm}(w \mid QCM)}{P(w \mid ACM)} \qquad (49)$$

Here, $P_{norm}(w|QCM)$ is the normalized Question Context Model and P(w|ACM) is the distribution which represents the Answer Context Model (ACM). Now, let us suppose that we add the source document from which the candidate answer was derived to the Answer Context Model. Then, we can calculate a new Perspective Similarity (PS') using the resulting model. If PS' > PS, we can deduce that the source document augments the similarity between the Question Context Model and the Answer Context Model. Likewise, if PS' < PS, we can deduce that source document detracts from the similarity between the Question Context Model and the candidate Answer Context Model. That is, the likelihood that the source document is a part of the Answer Perspective is a function of the change in Perspective Similarity (PS' – PS). This change in Perspective Similarity is defined as our model of Answer Credibility. Our approach is represented graphically in Figure 21.

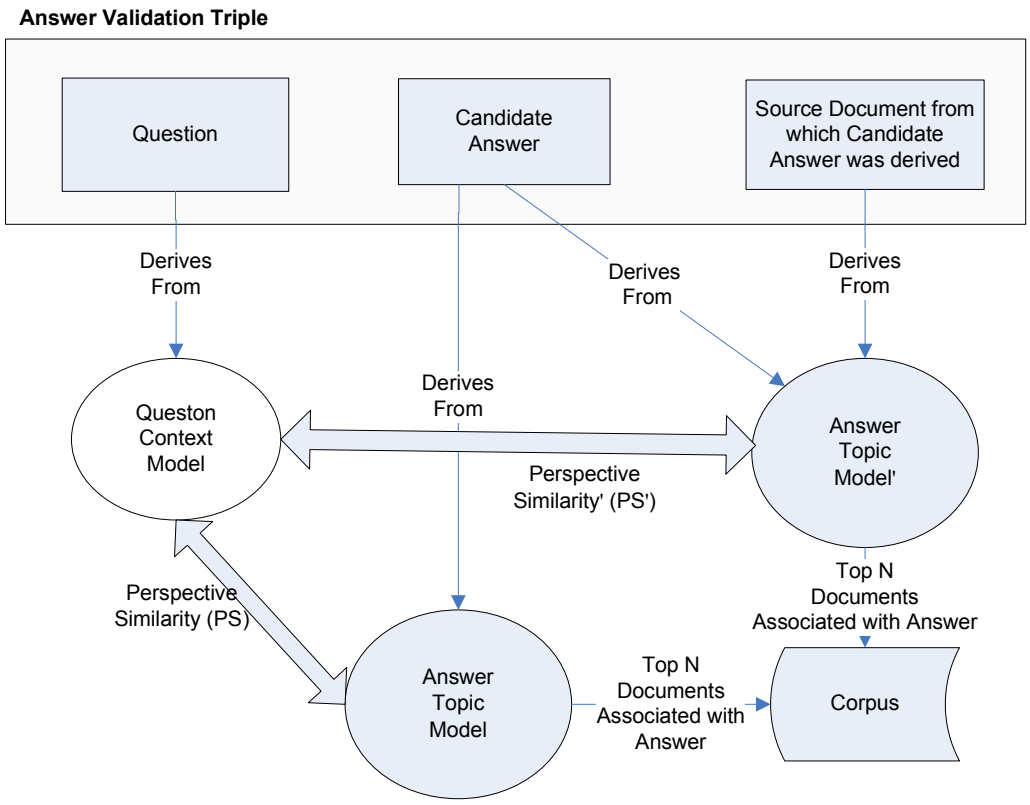

Figure 21: Graphical Representation of the Perspective Similarity Approach to Modeling Answer Credibility

We propose that Answer Credibility is not an absolute metric which by itself can be used to determine the correctness or incorrectness of an answer in a QA system. The nature of QA is such that there are numerous factors which are required to determine the right answer. The majority of these are, by their very nature, rooted in Natural Language Processing and Information Extraction. We do, however, believe that Answer Credibility should be an influencer of a QA system's answer score. We propose an interpolation technique that modulates an answer's score during the Answer Selection process using Answer Credibility. This interpolation method is given mathematically in the equation below:

$$score' = (1-\lambda) * score + \lambda * AnswerCredibility \qquad (50)$$

Here, score represents a QA system's answer score for a candidate answer prior to interpolation with Answer Credibility, score' represents an updated candidate answer score value after interpolation, and λ represents an interpolation constant.

In these experiments, we set λ using the average of the $P(z|R)$ values for all aspects which are included in the Question Context, as we did in the Naïve approach. Experiments presented later in this chapter describe the results of more sophisticated techniques to parameter estimation.

5.3.2 Experimental Methodology

The experimental methodology we used is shown as a block diagram in Figure 22. To validate our approach, we have used factoid questions which from the Text Retrieval Conference (TREC) 2006 Question Answering Track question set. These questions were processed against the TREC AQUAINT dataset (E. M. Voorhees & D. Harman, 2005). The first step of our experimentation was to index and pre-process the AQUAINT dataset using some standard techniques such as stemming and stop-word removal.

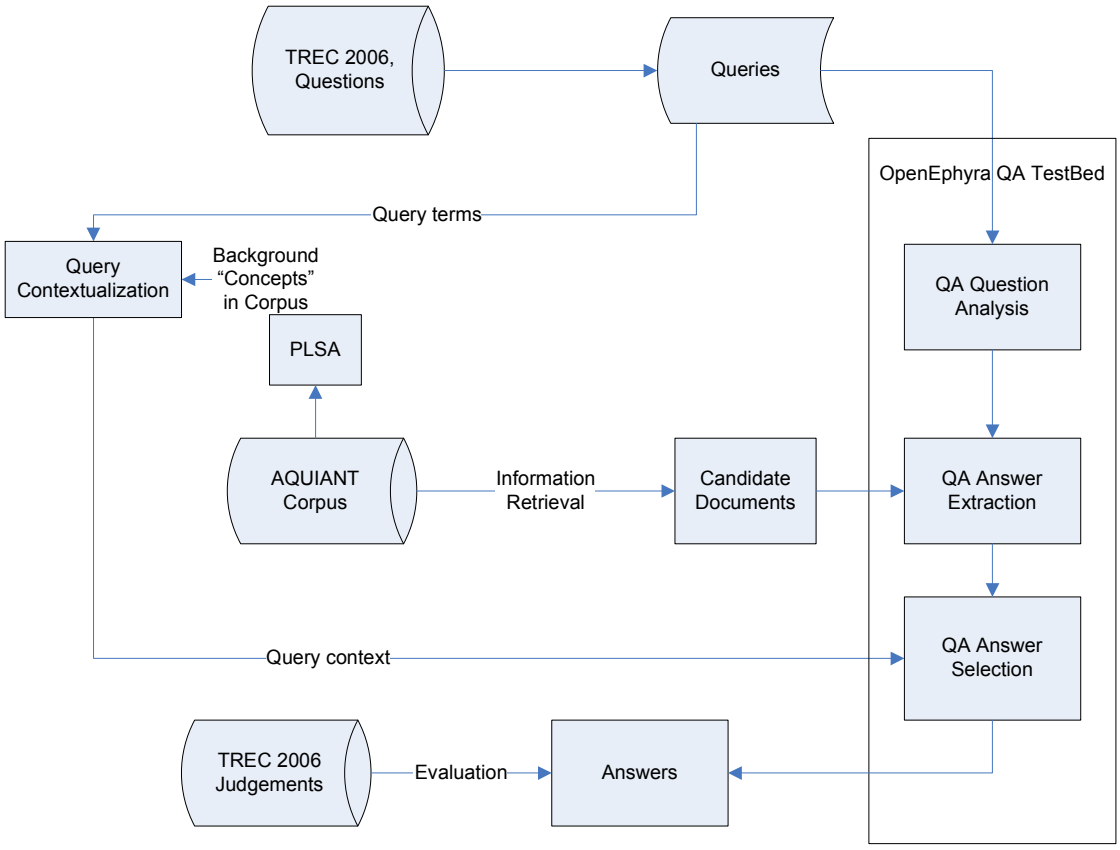

Figure 22: Experimental Methodology for the Perspective Similarity Approach to Modeling Answer Credibility

We performed Question Contextualization as described in section 5.2.4. Our PLSA model was trained using 50 z-categories. In addition, we used part-of-speech tagging to limit the words which were included as a part of the PLSA training documents to include only nouns (including proper nouns). We used the Lemur implementation (Ogilvie &

Callan, 2001) of PLSA in our experiments. Lemur was also used to implement the Answer Context Model; we used part-of-speech tagging so that only nouns (including proper nouns) were the only words included in the Answer Context Model.

The OpenEphyra Question Answering testbed (Schlaefer et al., 2006) was then used as the framework for our Answer Credibility implementation. Specifically, we inserted an Answer Credibility filter that implements the algorithm described in section 5.3.1 of this paper as a step in the OpenEphyra filtering pipeline.

For evaluation, we use the accuracy and Mean Reciprocal Rank (MRR) metrics, as described in section 5.2.2. The TREC 2006 judgments include a set of correct answers and documents associated with those answers for each question. We use these evaluation results to determine both answer accuracy and MRR. We consider an answer to be correct only if it is marked globally correct in the TREC QA judgments.

5.3.3 Results and Discussion

We compared the results of our approach against the baseline OpenEphyra Question Answering system. Our results are presented in Table 8 and Table 9 below. To facilitate interpretation of our results, we sub-divided the set of factoid questions into categories by their question words, following the example of Murdock (Murdock, 2006). The shaded cells in both tables indicate categories for which improvements were observed. The paired Wilcoxon signed-rank test was used to measure significance in improvements for MRR; the shaded cells in Table 9 indicate results for which the results were significant ($p<0.05$). Due to the binary

nature of the results for accuracy at the question level, a similar significance test was found to be inappropriate at the question level.

Table 8: Average Accuracy of Baseline vs. Baseline Including Answer Credibility Using the Perspective Similarity Approach

Question Category	Question Count	Baseline Accuracy	Baseline + Answer Credibility Accuracy
How	20	0.25	0.15
How many	58	0.12	0.10
How much	6	0.00	0.00
in what	47	0.64	0.57
What	114	0.23	0.29
what is	28	0.18	0.18
When	29	0.21	0.10
Where	23	0.30	0.48
Where is	6	0.33	0.33
Which	17	0.29	0.12
Who	17	0.47	0.65
Who is	14	0.57	0.71
Who was	24	0.38	0.54

Table 9: Average MRR of Baseline vs. Baseline Including Answer Credibility Using the Perspective Similarity Approach

Question Category	Question Count	Baseline MRR	Baseline + Answer Credibility MRR
How	20	0.33	0.30
how many	58	0.21	0.19
how much	6	0.08	0.02
in what	47	0.68	0.62
What	114	0.30	0.33
what is	28	0.26	0.26
When	29	0.30	0.12
Where	23	0.37	0.42
Where is	6	0.40	0.40
Which	17	0.38	0.18
Who	17	0.51	0.65
who is	14	0.60	0.79
who was	24	0.43	0.65

Our results show the following:

- A 6% improvement in accuracy over the baseline for “what”-type questions.
- A 17% improvement in accuracy over the baseline for “where”-type questions.
- An overall improvement of 19% in accuracy for “who”-type questions, which include the “who,” “who is” and “who was” categories
- A 9% improvements in MRR for “what” type questions
- A 12% improvements in MRR for “where” type questions
- An overall improvement of 37% in MRR for “who”-type questions, which include the “who,” “who is” and “who was” categories.

At this point, it can be seen that the Perspective Similarity approach yields improvements over the baseline approach. In the remainder of this section, we examine

some examples of questions that showed improvement to better understand and interpret our results.

First, we examine a "who" type question which was not correctly answered by the baseline system, but which was correctly answered after Answer Credibility was incorporated. For the question "Who is the host of the Daily Show?" the baseline system correctly determined the answer was "Jon Stewart" but incorrectly identified the document that this answer was derived from. For this question, the Question Context included the terms "stewart," "comedy," "television," "news," and "kilborn." (Craig Kilborn was the host of Daily Show until 1999, which makes his name a logical candidate for inclusion in the Question Context since the AQUAINT corpus spans 1996-2000). In this case, the correct document that the answer was derived from was actually the third one in the list. The Answer Credibility filter was able to correctly increase the score of that document so that it was ranked as the most reliable source for the answer and chosen as the correct final result. It should be noted here that the document containing the correct result actually described the transition in hosts on the Daily Show between Craig Kilborn and Jon Stewart; the inclusion of the term "kilborn" in the Question Context was likely the cause of the change in scoring. This example is promising because it shows the Answer Credibility filter performing the exact function that it was intended.

Next, we consider a case where the correct answer was at a lower position in the answer list in the baseline results and correctly raised higher, though not to the top rank, after the application of the Answer Credibility filter. For the question "What position did Janet Reno assume in 1993?" the correct answer ("attorney general") was ranked 5 in the

list. However, in this case the score associated with the answer was lower than the top-ranked answer by an order of magnitude. The Question Context for this question included the terms “miami,” “elian,” “gonzales,” “boy,” “attorney” and “justice.” After the application of the Question Context filter, the score and rank of the correct answer did increase, but the increase was not enough to overshoot the original top-ranked answer.

5.3.4 Limitations of Approach

Categories for which the Answer Credibility had negative effect included “how much” and “how many” questions. For these question types, the correct answer or correct document was frequently not present in the answer list. In this case, the Answer Credibility filter had no opportunity to increase the rank of correct answers or correct documents in the answer list. Furthermore, since we limited the terms included in the Question Context early on to only nouns, our methodology has limited applicability to those questions which require numerical answers. This same reasoning also limits our applicability to questions that require a date in response.

Finally, it is important to note here that the very nature of news data makes our methodology applicable to some categories of questions more than others. Typically, current events flood the news with a large number of articles that report similar information in a short time window. Since our methodology relies on the ability to derive semantic relationships via a statistical examination of text, it performs best on those questions for which some amount of supporting information is available.

To further understand the reasons for the relative advantages and disadvantages of each method, we next need to undertake a detailed examination of the factors that

contribute to the successes of these approaches. This in-depth examination is presented in the following section.

5.4 Parameter Estimation Studies

The interpolation parameter λ plays a critical role in out approach to incorporating Answer Credibility into the Answer Validation process. In this section, we explore four methods for setting λ:

1. A Naïve approach, where we set λ to a constant value across all questions
2. The QCM-Based Approach, where we set λ to the average of the P(z|R) values for those aspects that are included in the Question Context Model. It should be noted that this is the methodology that we used in the previous two sections. The intuition behind this approach is that we want to "add-in" the Answer Credibility score based on the likelihood of relevance of the Question Context Model to the user's question. This approach implies that every question
3. The Perspective Similarity Approach, where we set λ to the calculated value of the Perspective Similarity. The intuition behind this approach is that we want to "add-in" the Answer Credibility score based on how closely related the Question Context is to the Answer Topic Model. In this case, the Answer Topic Model does not include the source document from which the candidate answer was derived.
4. The Modified Perspective Similarity Approach, where we set λ to the calculated value of Perspective Similarity'. The intuition behind this approach is that we

want to "add-in" the Answer Credibility score based on how closely related the Question Context is to the Answer Topic Model and the source document for the candidate answer.

Mathematically, the four cases are be represented as follows:

1. $\lambda = Cons\tan t$ (51)

2. $\lambda = 1/N\sum_{i=1}^{N} P(z_i \mid R)$ (52)

3. $\lambda = \sum_{w} P_{norm}(w \mid QCM)\log\frac{P_{norm}(w \mid QCM)}{P(w \mid ACM)}$ (53)

4. $\lambda = \sum_{w} P_{norm}(w \mid QCM)\log\frac{P_{norm}(w \mid QCM)}{P(w \mid ACM')}$ (54)

Here, N is the number of aspects included in the Question Context Model. The definition of P(z|R) is the same as that previously provided in equation (36), and the definition of P_{norm}(w|QCM) and P_{norm}(q|ACM) are the same as that provided in equation (44) and (45). The difference between ACM and ACM' is that ACM includes documents other documents that are a part of the Answer's context, but not the specific source document related to this question. ACM' includes the source document for the candidate answer.

The results of our experiments are presented in Figure 23 and Figure 24 and in Table 10 and Table 11.

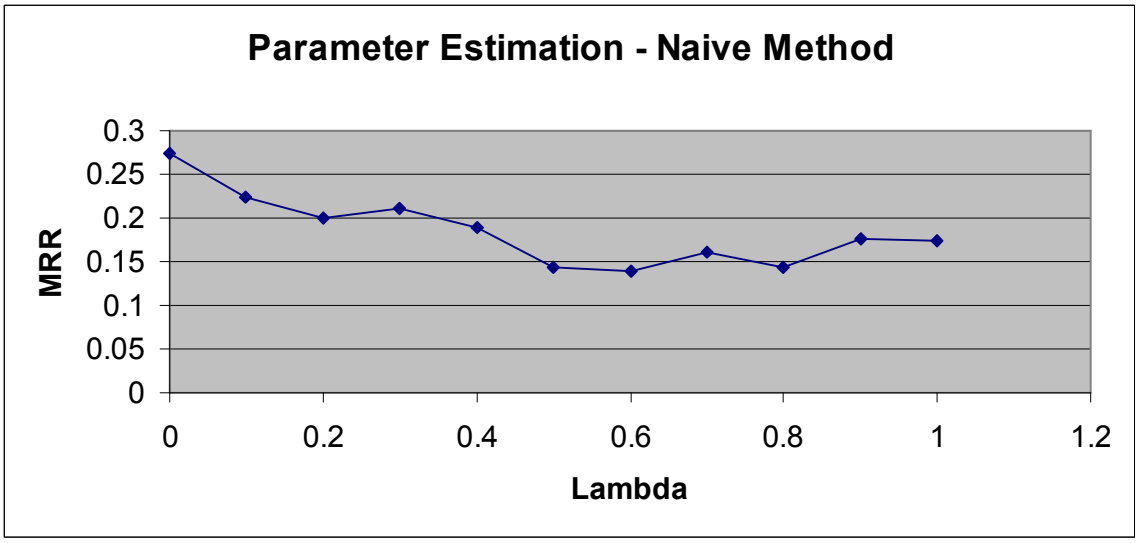

Figure 23: Average MRR using the Naïve Approach for Setting the Interpolation Parameter (λ)

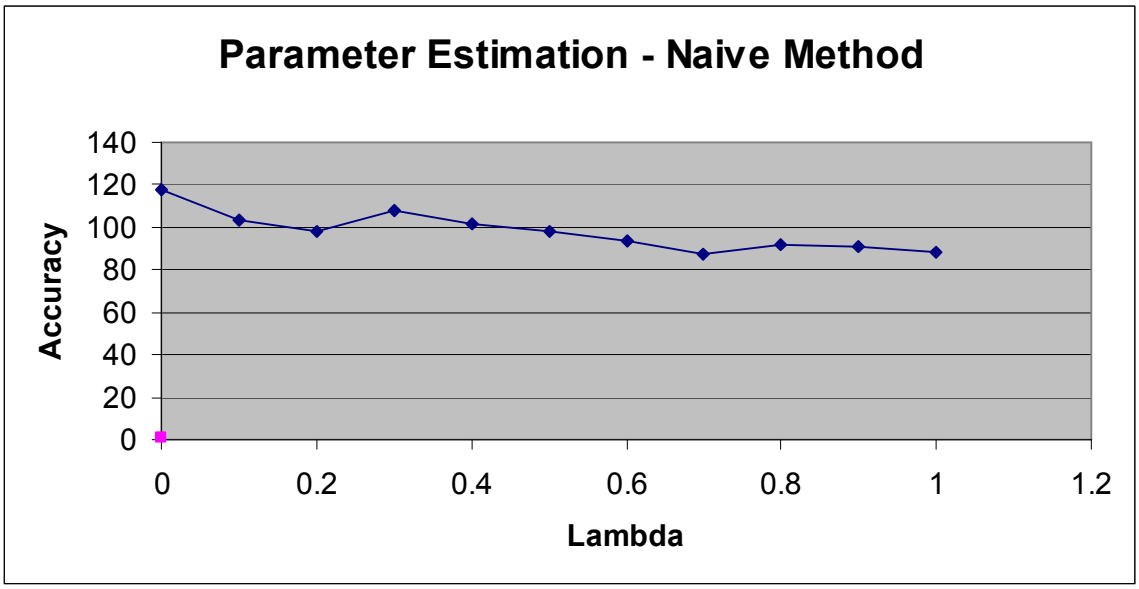

Figure 24: Average Accuracy using the Naïve Approach for Setting the Interpolation Parameter (λ)

Table 10: Comparison of Accuracy Across Three Methods for Setting the Interpolation Parameter (λ). The approaches compared are a) the QCM-Based Approach b) the Perspective Similarity Approach and c) the Perspective Similarity' Approach

Question Category	Question Count	Baseline Correct	λ = Average P(z|R)	λ = PS	λ = PS'
How	20	5	3	3	3
How many	58	7	6	6	6
How much	6	0	0	0	0
in what	47	30	27	27	29
What	114	26	33	35	38
what is	28	5	5	5	7
When	29	6	3	4	4
Where	23	7	11	11	12
Where is	6	2	2	2	2
Which	17	5	2	4	4
Who	17	8	11	12	12
Who is	14	8	10	10	11
Who was	24	9	13	15	15
Total	403	118	126	134	143

Table 11: Comparison of MRR Across Three Methods for Setting the Interpolation Parameter (λ). The approaches compared are a) the QCM-Based Approach b) the Perspective Similarity Approach and c) the Perspective Similarity' Approach

Question Category	Question Count	Baseline MRR	λ = Average P(z|R)	λ = PS	λ = PS'
How	20	0.33	0.30	0.30	0.30
How many	58	0.21	0.19	0.23	0.23
How much	6	0.08	0.02	0.02	0.02
in what	47	0.68	0.62	0.65	0.67
What	114	0.30	0.33	0.36	0.42
what is	28	0.26	0.26	0.26	0.32
When	29	0.30	0.12	0.17	0.17
Where	23	0.37	0.42	0.42	0.42
Where is	6	0.40	0.40	0.15	0.26
Which	17	0.18	0.12	0.17	0.17
Who	17	0.51	0.65	0.68	0.68
Who is	14	0.60	0.79	0.72	0.79
Who was	24	0.43	0.65	0.71	0.71
Average		0.27	0.32	0.36	0.37

In the remainder of the section, we consider each method of setting λ and examine the advantages and disadvantages of each approach. First, we consider the Naïve approach where Answer Credibility is added to the Answer Extraction score by a constant value of λ for all questions. From Figure 23 and Figure 24 we can see a degradation in both accuracy and MRR from the OpenEphyra baseline (the case where λ = 0). As a result, we can conclude that the Naïve approach is not appropriate methodology for setting λ. In our model, the success of Answer Credibility is tied to how well the Question Context Model "fits" the question. If λ is set to a constant value for all questions, we attempt to treat all questions as though they had Question Context Models

and Answer Context Models with equal likelihoods of relevance. This "one size fits all" model is not successful in the general case.

However, when considering the significance of our Answer Credibility model, it is valuable to consider the accuracy and MRR values for the two extreme cases of λ; that is, in Table 12 we specifically consider accuracy and MRR for the cases where λ =0 and λ =1. While performance degrades in general, we do see a marked improvement in certain categories for which our model is well-suited. Specifically we see:

- *A 2% improvement in accuracy and a 9% improvement in MRR in "what" type questions,*
- *A 21% improvement in accuracy and a 12% improvement in MRR for "where" type questions*
- *A 23% improvement in accuracy and a 21% improvement in MRR for "who" type questions*
- *A 21% improvement in accuracy and a 13% improvement in MRR for "who is" type questions*
- *A 25% improvement in accuracy and a 36% improvement in MRR for "who was" type questions*

These results underscore the value of the Answer Credibility model for certain question categories in absence of the score provided by the baseline system. These results are further re-enforced when we consider a graph of average Answer Credibility values broken out by question category, as depicted in Figure 25. Figure 25 shows the average Answer Credibility values for candidate answers for each question category; some of

these candidate answers correctly answer the question and some (the majority) of these candidate answers do not correctly answer the question. We can see that for the categories of question for which our methodology performs the best, there is a large difference in average Answer Credibility values between the candidate answers that correctly answer the question and those that do not. Thus, our methodology performs (for these categories of questions), the proper Answer Validation mechanism of being able to distinguish correct answers from incorrect answers.

Table 12: Accuracy and MRR using the Naïve Approach for Setting the Interpolation Parameter (λ) for the Cases where λ =0 and λ =1

Question Category	Question Count	Baseline Accuracy (λ = 0)	Baseline MRR (λ = 0)	Answer Credibility Accuracy λ = 1	Answer Credibility MRR λ = 1
How	20	5	0.33	1	0.16
How many	58	7	0.21	2	0.12
How much	6	0	0.08	0	0.02
in what	47	30	0.68	6	0.26
What	114	26	0.30	28	0.37
what is	28	5	0.26	2	0.17
When	29	6	0.30	2	0.16
Where	23	7	0.37	10	0.42
Where is	6	2	0.40	0	0.14
Which	17	5	0.38	0	0.13
Who	17	8	0.51	12	0.62
Who is	14	8	0.60	11	0.68
Who was	24	9	0.43	14	0.58
Average		118	0.37	91	0.183

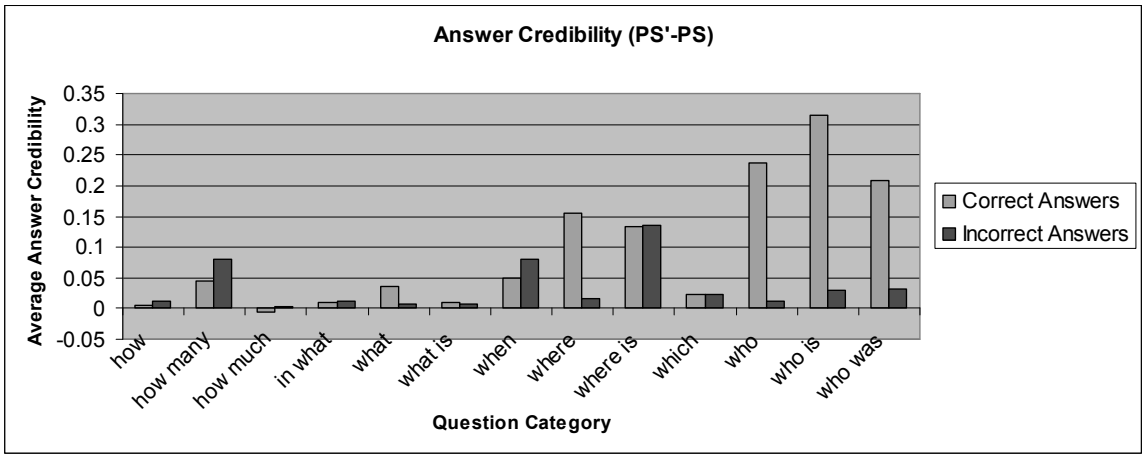

Figure 25: Histogram of the Average Answer Credibility Values, Split by Question Category and Correct/Incorrect Answers

When considering the average Answer Credibility score values, we find that we can also better understand our negative results. In many cases, ("how," "how many," "which") there is little to no difference in Answer Credibility values between correct and incorrect answers. This leads us to believe that our approach is not able to discern correct answers from incorrect answers for these question categories.

Next we consider the QCM-Based approach where Answer Credibility is added to the Answer Extraction score by a factor directly proportional to the likelihood of relevance of the terms in the Question Context Model. From Table 10 and Table 11, we can see that this approach does better than the OpenEphyra baseline, both in terms of specific categories and overall, for all question categories. To further examine why this approach is successful, it is useful to consider the distribution of the interpolation parameter across the various question categories. Figure 26 shows the average values for the interpolation parameter (λ) respectively, grouped by question category and grouped by correct and incorrect answers. That is, for a given Question Category, we consider the

values of λ for the entire list of candidate answers for every question in that category, some of which correctly answer the question and most of which do not.

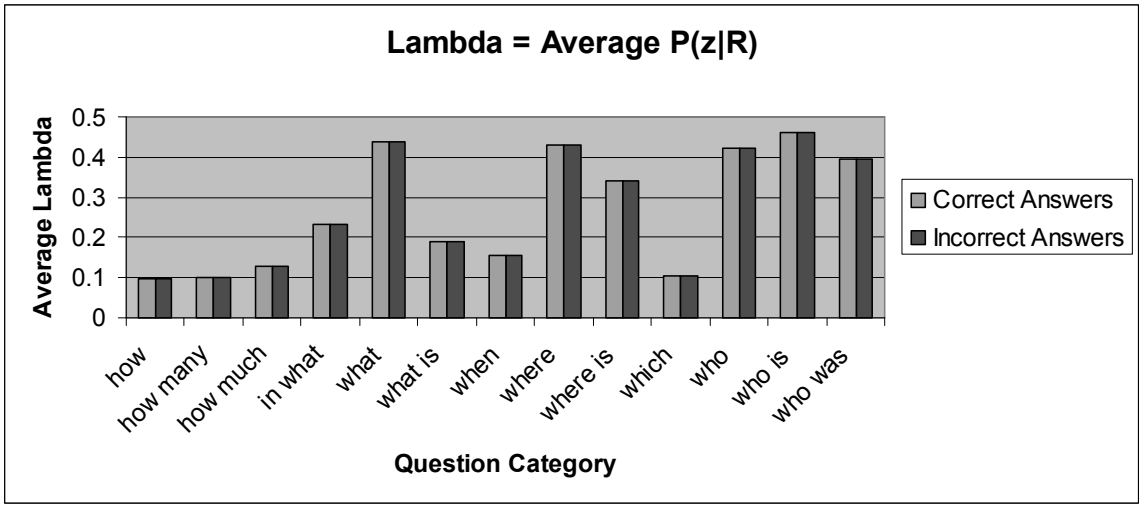

Figure 26: Histogram of the Average Values for the Interpolation Parameter (λ) for the QCM-Based Approach, Split by Question Category and Correct/Incorrect Answers

We see that for a given Question Category, the QCM-based approach makes no distinction between in interpolation parameters between correct and incorrect answers. That is, in this approach, all answers for a given question are given the same interpolation value. This is true because the Question Context Model is assigned at the question level, and remains the same for every candidate answer that is considered for the question. As such, in this approach, it is only Answer Credibility that provides us to the ability to distinguish between correct and incorrect answers. Furthermore, this approach does not take into account our original premise that Question Context is not the same as Answer Context; in this approach, Answer Context does not play any role in the determination of the interpolation parameter.

Next we consider the Perspective Similarity-based approach where Answer Credibility is added to the Answer Extraction score by an interpolation parameter that is set to be the same value as the Perspective Similarity. From Table 10 and Table 11, we can see that this approach does better than the OpenEphyra baseline. In this case, the interpolation parameter is graphed in Figure 27 and we start to see a divergence between the values of λ for correct and incorrect answers. This delta in λ values, on top of the deltas in Answer Credibility values observed in Figure 25 shows how this approach is able to distinguish between correct and incorrect answers. Furthermore, since Answer Credibility is added to a candidate answer's Answer Extraction score by a factor directly proportional to the similarity between the Question Context and Answer Context, we are able to vary the interpolation parameter at the level of an individual candidate answer. However, this approach falters because there is no influence from the specific source document from which the candidate answer was derived. This means that we are looking at a possibly incomplete picture of the full Answer Context.

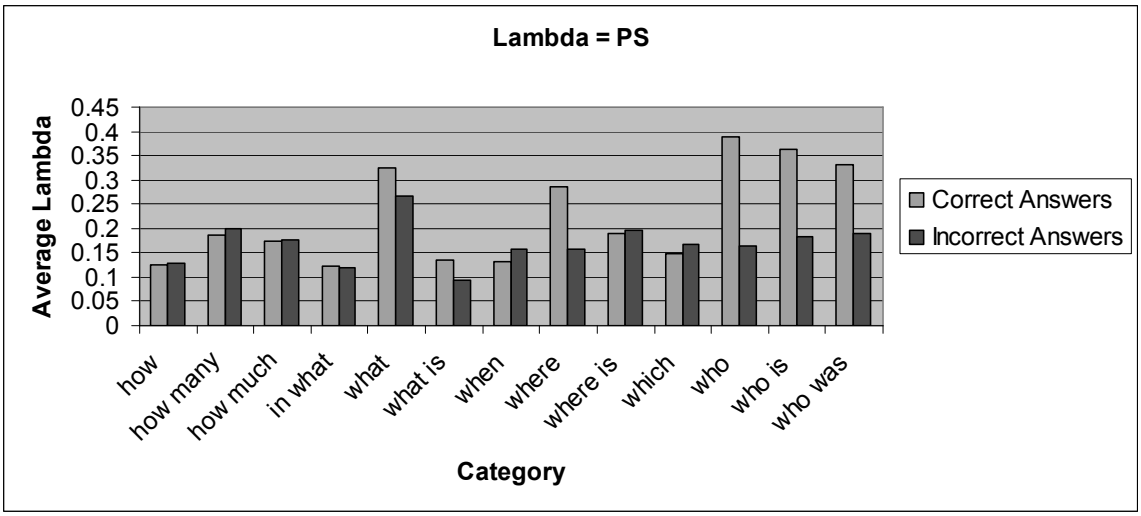

Figure 27: Histogram of the Average Values for the Interpolation Parameter (λ) for the Perspective Similarity Approach, Split by Question Category and Correct/Incorrect Answers

Next we consider the Perspective Similarity'-based approach where Answer Credibility is added to the Answer Extraction score by an interpolation parameter that is set to be the same value as the Perspective Similarity'. In this case, we see a greater divergence between the values of λ for correct and incorrect answers than in the previous case. This approach most accurately reflects our proposed model for Question Context and Answer Context. The success of Answer Credibility is tied to the similarity between Question Context and Answer Context, which includes the specific source document from which the candidate answer was derived.

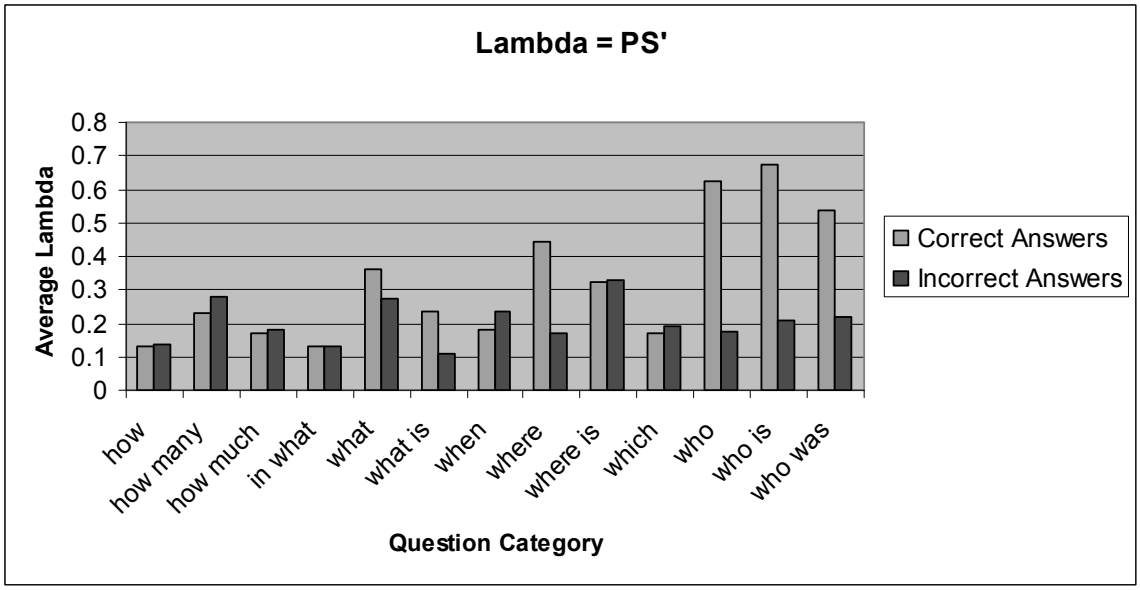

Figure 28: Histogram of the Average Values for the Interpolation Parameter (λ) for the Perspective Similarity' Approach, Split by Question Category and Correct/Incorrect Answers

A summary table that shows the improvements in accuracy and MRR achieved by each variation of our Answer Credibility model is shown by Question Category in Figure 29 and Figure 30.

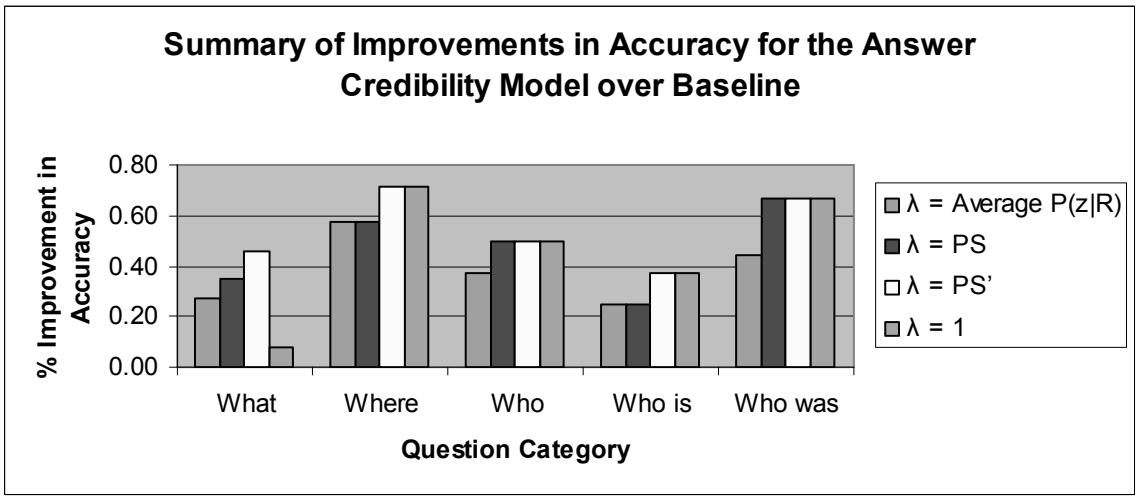

Figure 29: Summary of Percent Improvements in Accuracy of the Answer Credibility Model over Baseline for all Parameter Estimation Approaches

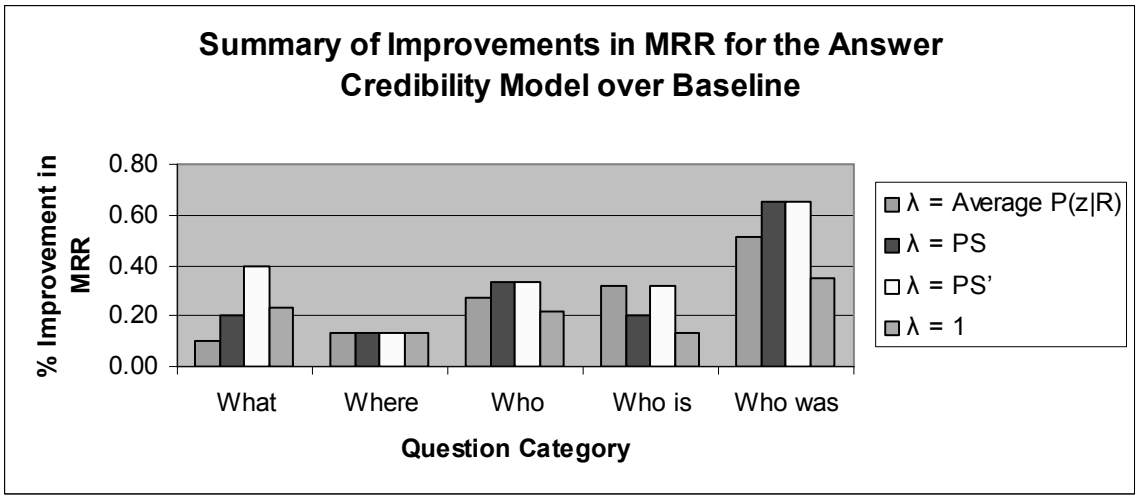

Figure 30: Summary of Percent Improvements in MRR of the Answer Credibility Model over Baseline for all Parameter Estimation Approaches

The following improvements were achieved by our Answer Credibility model across all Question Categories:

- A 7% improvement in accuracy and a 12% improvement in MRR for the QCM-based interpolation approach (where λ = average P(z|R))
- A 14% improvement in accuracy and a 17% improvement in MRR for the Perspective Similarity-based interpolation approach (where λ = PS)
- A 20.1% improvement in accuracy and a 19.4% improvement in MRR for the Perspective Similarity'-based interpolation approach (where λ = PS')

We would like to discuss one further insight here regarding the success of our methodology on "who," "what" and "where" type questions. We believe that some part of our success on these questions has to do with the ability of our baseline system (OpenEphyra) which uses a pattern-matching methodology to acquire candidate answers for "who," "what" and "where" type questions. We notice that for these question categories there are about typically more candidate answers retrieved per question than

for other question categories. Furthermore, OpenEphyra has a set of approximately 120 pattern types that are used to extract candidate answers. Of these, we believe that the majority of the patterns are tied to "who," "what" and "where" type questions, which further reinforces our insight. We speculate that given a baseline system that had similar performance on "how," "which" and "where is" type questions, we might be able to improve accuracy and MRR for those question categories as well. Since our approach eliminates numbers and dates as a part of the Question Contextualization process, we do not expect to be able to achieve any improvements in accuracy or MRR for "how many", "how much" or "when" type questions.

5.4.1 Comparison with State-of-the-Art Approaches

In this section, we compare our results with the results reported by state-of-the-art approaches to Answer Validation. Specifically, we consider the results presented in (S Harabagiu & Hickl, 2006) using a Textual Entailment Method of Answer Validation. Using TREC 2006 data, the methods used by Harabagiu show a 20% improvement in accuracy and a 23% improvement in MRR over their baseline methods. Our most promising results are for the case where the interpolation parameter is set using the Perspective Similarity'-based approach. In that case, we see that accuracy overall increases from 118 (baseline OpenEphyra) to 142, which is an improvement of 20%. This is comparable to the results reported by Harabagiu. Furthermore, our results show an average increase in MRR from 0.27 to 0.37, which is an improvement of 19.4% over the baseline. This is slightly less than the reported increase in MRR from Harabaigu. These results show that in general, our methodology is best suited to the case where the

correct answer was highly ranked, but perhaps not the top answer in the list. Our methodology is less successful than the Harabagiu method in elevating answers that were originally ranked very low in the list of candidate answers to top positions.

6. CONCLUSIONS AND FUTURE WORK

In this chapter we conclude this dissertation work by highlighting our research contributions and by discussing future directions. We provide some suggestions about how our current work may be extended for application to other research areas such as inclusion of structured data sources into the Question Context Model, domain-specific QA, and incorporation of data fusion techniques with our methodologies.

6.1. Summary of Contributions

The research presented in this dissertation presents our work in applying statistical language modeling approaches, which have been proven successful in Information Retrieval applications, to Question Answering. Specifically, we define a theoretical model, the Aspect-Based Relevance Language Model, and then describe an approach that derives a Question Context Model from that theoretical framework. We define a semantic Question Context Model to be background knowledge derived from the corpus that can be used to represent the user's information need more completely than the terms in the query alone. We incorporate this semantic Question Context Model into the both the first and second stages of the QA architecture.

In the second stage of the Question Answering architecture, we model a novel concept called Answer Credibility. Conceptually, Answer Credibility can be thought of as the similarity between the semantic Question Context Model and an Answer Topic Model associated with a candidate answer and its source document. We propose that the higher this similarity score, the more believable the candidate answer. Our approach is significant because:

1. We extend the usage of statistical language modeling techniques, which have been successfully applied to the first stage (the IR stage) of QA, into the second stage (the NLP and IE stage) of QA
2. We use the corpus itself as a knowledge source, rather than relying on externally available semantic resources.

The primary contributions of this research are as follows:

1. Design of the Aspect-Based Relevance Language Model, which provides a mechanism of relating an aspect (as defined by PLSA) to a question, based on its likelihood of relevance. We base our approach to modeling the Aspect-Based Relevance Model on the same set of assumptions that guide the Relevance-based Language Modeling approach. We assume that for every information need (or question) there exists an underlying relevance model R. In our model, we assume that R is assigned probabilities P(z|R) where z is a latent aspect of an information need, as defined by the PLSA Aspect Model. Thus, the relevance to an information need is described not in terms of words, but in terms of the latent aspects or concepts associated with the information need. Conceptually, this alternative representation of the information need includes context information via the inclusion of aspects that extends beyond knowledge related to the specific words themselves. If we use P(z|R) and P(w|z) as ranking mechanisms, we can arrive at a Question Context Model that includes only the most relevant aspects for an information need, and only the words that are most closely related to those aspects. We thus propose that the Aspect-Based Relevance Language Model can serve as the basis for a Question Context model.

2. Design of a Question Context Model, and the incorporation of the Question Context Model into the Candidate Document Selection process of a Question Answering architecture. To incorporate the Question Context Model (QCM) into an IR framework, we consider that there are essentially two sides to the "zero-frequency" problem : (1) A query input to a QA system rarely contains all of the words that would be relevant to the information need that the query represents; and (2) A document rarely contains all of the words that are related to the information content of the query. To examine the effectiveness of the proposed Query Context Model (QCM), we need to understand its effectiveness when applied to both facets of the "zero-frequency" problem: context-based query expansion and context-based document smoothing. Our results show that when the top 1 and top 5 aspects are included as a part of the Query Context Model, both precision and recall results compare favorably against the baseline language modeling retrieval methods, and there is a significant improvement in recall performance as measured by the Wilcoxon signed rank test. These results also validate our model at a very basic level; in other words, our ranking of aspects by $P(z|R)$ does accurately reflect the ordering of aspects in order of relevance to a query.

3. Design of an Answer Credibility Model, and the incorporation of the Answer Credibility Model into the Answer Validation process of a Question Answering architecture. Our mathematical representation of Answer Credibility attempts to quantify the reliability of a source using the semantic Question Context Model as a basis. We first propose an approach that describes the relationship between the Question

Context Model, a calculated Answer Context Model and a novel concept called the Answer Perspective, which we define to be the overlap between the Question Context and the Answer Context. We define Answer Credibility to be a similarity measure between the semantic Question Context built via the Aspect-Based Relevance Language Model and the source document from which the answer was derived. As we derive a statistical model of Question Context, our model of Answer Credibility is likewise a statistical one. Our results show improvements in accuracy and MRR for "what," "where," "who is," "who was," and "who" type questions. Our approach does not show improvements in question types that require numerical or date answers. However, this is as expected result given that our methodology for Question Contextualization uses part-of-speech tagging to strip out any word that is not a noun.

6.1.1 Answers to Research Questions

It was shown that language modeling techniques can be applied to both Candidate Document Selection (QA stage 1) and Answer Validation (QA stage 2). When we applied our methodology to Candidate Document Selection, we show a statistically significant improvement in recall at of 50 and 100 documents retrieved. When we applied our methodology to Answer Validation, we show improvements in accuracy of 20% and improvements in MRR of 19%, both of which are comparable with state-of-the-art textual entailment methods for Answer Validation. However, unlike textual entailment methods, our approach does not require significant training time or training data.

6.2. Conclusions and Future Work

The notion of context is very complex and the design of an overarching model for context the Question Answering process is not simple. In this work, we have proposed the creation of a Question Context Model using statistical methods based on only un-structured data sources. Furthermore, we have not considered the incorporation of external knowledge into our Question Context Model in this effort. While this has ensured that our approach is not dependent on domain-specific resources such as ontologies, we may want to consider an approach that includes such resources in the future for domains in which ontologies or thesauri are available. Furthermore, the development of a holistic Question Context for Answer Extraction and Answer Validation may be thought of as an Information Fusion problem. In our controlled environment in the TREC testbed, we were limited to only a single source of data; in the realistic environment of the World Wide Web we must consider a more hetero-geneous set of information sources – each with potentially different levels of credibility. In the remainder of this section, we highlight four areas which are potential opportunities for enhancement of our research in Question Context going forward.

Further Development of Models: Our research has shown that there is a correlation between the interpolation parameter (λ) and question category. We propose that an extension of our work would be to quantify the correlation between the interpolation parameter and question category, and possibly introduce question category as another factor that changes the interpolation parameter on a question by question basis. Furthermore, in this work, we established that the "correctness" of the aspects that are constructed by PLSA are critical to the success of our models. Future research may need

to consider the methods by which we can quantify "correctness" of an aspect, and if there are any alternative methods that produce aspects better suited to question enhancement than PLSA. In addition, more elaborate investigation may be conducted into the PLSA parameters set during the course of these experiments. While we used perplexity in our preliminary work to determine the number of PLSA categories and words selected from each category, we may want to consider additional metrics to validate our parameter selecions.

Inclusion of Structured Data into Question Context Model: At present, a significant amount of information is structured in databases of different formats and different semantics. This is true especially in corporate environments, where customer records and financial data is stored in large scale databases and data warehouses. In the past, data warehouses have been designed to aid corporate decision making processes, and are rich information sources. Of extreme value would be to have a Question Answering system access these data sources, and to be able to fuse the information in such sources with textual data. Moreover, even for the textual data, different file formats should be considered. Instead of only ASCII-based XML files, textual information could also be retrieved from PDF files, postscript files, Word documents, Excel files, Powerpoint presentations and even UML (Unified Modeling Language) models. The problem is to gather and organize large-scale collections of files in such varied formats. Furthermore, sometimes structured information (such as tables) may be contained within an unstructured document, which adds to the complexity of this processing.

Extension into Genomics and/or Biomedical Domains: The Genomics and Biomedical domains are unique in that they are richly populated with generally accepted ontological resources. When applying our techniques to such domains, it becomes necessary to take advantage of resources such as the Gene Onotology and the Unified Medical Language System (UMLS) which define a structured vocabulary of terms and concepts. We had originally considered extending our work to the Genomics Domain as a part of the present work; however, we found that such an extension represented too much of a deviation from the core focus of our effort in developing a statistical model of Question Context and Answer Credibility. A further opportunity with this approach would be to use a different baseline system for our experiments. This would allow us the ability to determine what parts of our methodology are truly dependent on the baseline Answer Extraction system.

Data Fusion Approaches to Modeling Question Context: In a heterogeneous information environment, data fusion techniques have been useful for combining data from disparate information sources. Specifically, in domains such as Intelligence Analysis for the Department of Defense, data fusion has been used effectively to integrate information from disparate information sources. A particularly interesting problem is presented by questions that have answers distributed across different data collections. Such questions and their underlying data sources pose research problems that add new complexity to QA processing; for such questions, a two-stage QA architecture may not be sufficient. Combining information from different data sources is not a trivial problem. The answer may be either scattered throughout the collections, but its generation may not require any inference; or the answer may be provided only by inferring new information

or relations between its subparts. The fusion of information from different documents and from different sources often relies on complex inferential processes. Furthermore, variation in reliability of different sources and potentially conflicting information across sources adds to the complexity of this processing.

Complex Question Answering: To date, this work has focused on Question Answering for fact-seeking questions. A natural extension of this work would be in the area of Complex Question Answering which deals with questions that require the use of complex reasoning processes and/or return complex answers consisting of more than one fact. One approach to Complex QA that has been adopted by many researchers is to decompose a complex question into a series of fact-seeking questions and reuse techniques developed for answering simple questions. An extension of this work may explore how applicable our techniques are to such decomposed complex questions.

LIST OF REFERENCES

Amati, G., & Van Rijsbergen, C. J. (2002). Probabilistic Models of Information Retrieval Based on Measuring the Divergence from Randomness. *ACM Transactions on Information Systems (TOIS), 20*(4), 357-389.

Bahl, L. R., Brown, P. F., de Souza, P. V., & Mercer, R. L. (1989). A tree-based statistical language model for natural language speech recognition. *Communications of the ACM, 37*(7), 1001-1008.

Balog, K., Weerkamp, W., & de Rijke, M. (2008). *A few examples go a long way: constructing query models from elaborate query formulations.* Paper presented at the Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval.

Berger, A., & Lafferty, J. (1999). Information retrieval as statistical translation. *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, 222-229.

Bilotti, M. W., Katz, B., & Lin, J. (2004). What works better for question answering: Stemming or morphological query expansion, *ACM SIGIR Workshop on Information Retrieval for Question Answering (IR4QA)*.

Bontcheva, K., Tablan, V., Maynard, D., & Cunningham, H. (2004). Evolving GATE to meet new challenges in language engineering. *Natural Language Engineering, 10*(3-4), 349-373.

Brown, P., Della Pietra, S., Della Pietra, V., Jelinek, F., Lafferty, J., Mercer, R., et al. (1990). A statistical approach to machine translation. *Computational Linguistics, 16*(2), 79-85.

Buchholz, S. (2001). Using Grammatical Relations, Answer Frequencies and the World Wide Web for TREC Question Answering, *Online proceedings of 2001 Text Retrieval Conference*.

Buchholz, S., & Daelemans, W. (2002). Complex answers: a case study using a WWW question answering system. *Natural Language Engineering, 7*(04), 301-323.

Buckley, C. (2004). *Why current IR engines fail.* Paper presented at the Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval Sheffield, UK.

Callan, J. P., Croft, W. B., & Harding, S. M. (1992). The INQUERY Retrieval System. *Proceedings of DEXA-92, 3rd International Conference on Database and Expert Systems Applications*, 78-83.

Caporaso, J. G., Baumgartner Jr., W. A., Kim, H., Lu, Z., Johnson, H. L., Medvedeva, O., et al. (2006). Concept Recognition, Information Retrieval, and Machine Learning in Genomics Question-Answering, *Online proceedings of 2006 Text Retrieval Conference*.

Chomsky, N. (1956). Three models for the description of language. *IEEE Transactions on Information Theory, 2*(3), 113-124.

Cleverdon, C. W., Mills, J., & Keen, M. (1966). *Aslib Cranfield Reserach Project: Factors Determining the Performance of Indexing Systems*: Cranfield University, College of Aeronautics

Craswell, N., & Hawking, D. (2002). TREC 2002 WEB Track Overview, *NIST Special Publication*.

Cui, H., Kan, M. Y., Chua, T., & Xiao, J. (2004). A Comparative Study on Sentence Retrieval for Definitional Questions Answering, *ACM SIGIR Workshop on Information Retrieval for Question Answering (IR4QA)* (pp. 90-99).

Dalmas, T., & Webber, B. (2004). Answer Comparison: Analysis of Relationships between Answers to 'Where'-Questions, *Proceedings of the 7th Annual Colloquium for Computational Lingusitics*. UK.

Dalmas, T., & Webber, B. (2005). Information fusion for answering factoid questions.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science, 41*(6), 391-407.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society, 39*(1), 1-38.

Derczynski, L., Wang, J., Gaizauskas, R., & Greenwood, M. (2008). A Data Driven Approach to Query Expansion in Question Answering, *Proceedings of COLING 2008 IR4QA Workshop*. Manchester, UK.

Doran, C., Aberdeen, J., Damianos, L., & Hirschman, L. (2001). *Comparing several aspects of human-computer and human-human dialogues.* Paper presented at the Proceedings of the Second SIGdial Workshop on Discourse and Dialogue - Volume 16.

Fogg, B. J., & Tseng, H. (1999). The elements of computer credibility, *Conference on Human Factors in Computing Systems* (pp. 80-87). Pittsburgh, PA: ACM Press New York, NY, USA.

Gaizauskas, R. (1998). Information Extraction: Beyond Document Retrieval. *Journal of Documentation, 54*(1), 70-105.

Gaizauskas, R., Greenwood, M. A., Harkema, H., Hepple, M., Saggion, H., & Sanka, A. (2005). The University of Sheffield's TREC 2005 Q&A Experiments, *Online proceedings of the 2005 Text Retrieval Conference*.

Gaizauskas, R., Greenwood, M. A., Hepple, M., Roberts, I., & Saggion, H. (2004). The University of Sheffield's TREC 2004 Q&A Experiments, *Online proceedings of the 2004 Text Retrieval Conference*.

Gaizauskas, R., Greenwood, M. A., Hepple, M., Roberts, I., Saggion, H., & Sargaison, M. (2003). The University of Sheffield's TREC 2003 Q&A Experiments, *Online proceedings of the 2003 Text Retrieval Conference*.

Gaizauskas, R., Hepple, M., & Greenwood, M. (2004). IR4QA: Information Retrieval for Question Answering, *ACM SIGIR Workshop on Information Retrieval for Question Answering (IR4QA)*.

Glöckner, I. (2006). University of Hagen at QA@CLEF 2006: Answer Validation Exercise, *On-line Proceedings of CLEF 2006*.

Glöckner, I. (2007). University of Hagen at CLEF 2007: Answer Validation Exercise, *On-line Proceedings of CLEF 2007*.

Glöckner, I. (2008). University of Hagen at CLEF 2008: Answer Validation Exercise, *On-line Proceedings of CLEF 2008*.

Godbole, N., Srinivasaiah, M., & Skiena, S. (2007). *Large-scale sentiment analysis for news and blogs.* Paper presented at the Proceedings of International Conference on Weblogs and Social Media.

Green, B., A., W., Chomsky, C., & Laughery, K. (1961). BASEBALL, an automatic question answerer, *Proccedings of Western Joint Computer Conference* (pp. 219-224).

Greenwood, M. (2008). Introduction to the IR4QA Workshop, *Proceedings of COLING 2008 IR4QA Workshop*. Manchester, UK.

Greenwood, M. A., Stevenson, M., & Gaizauskas, R. (2006). *The University of Sheffield's TREC 2006 Q&A Experiments*. Paper presented at the Online proceedings of the 2006 Text Retrieval Conference.

Harabagiu, S., & Hickl, A. (2006). *Methods for using textual entailment in open-domain question answering.* Paper presented at the International Conference of the Association for Computational Linguistitics (ACL) 2006, Philadelphia, PA, USA.

Harabagiu, S., Moldovan, D., Clark, C., Bowden, M., Hickl, A., & Wang, P. (2005). Employing Two Question Answering Systems in TREC 2005, *Online proceedings of the 2005 Text Retrieval Conference*.

Harabagiu, S., Moldovan, D., Rus, V., & Morarescu, P. (2001). Falcon: Boosting Knowledge for Answer Engines, *Proceedings of 2001 Text Retrieval Conference*.

Harman, D., & Buckley, C. (2004). *The NRRC reliable information access (RIA) workshop.* Paper presented at the Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, Sheffield, UK.

Harter, S. (1975). A probabilistic approach to automatic keyword indexing. Part I: On the distribution of specialty words in a technical literature. *Journal of the American Society for Information Science, 26*(4), 197-206.

He, B., & Ounis, I. (2005). A study of the dirichlet priors for term frequency normalisation, *Proceedings of the 28th annual international ACM SIGIR conference on Research and Development in Information Retrieval* (pp. 465-471). Salvador, Brazil.

He, B., & Ounis, I. (2005). Term Frequency Normalisation Tuning for BM25 and DFR Model, *Leture Notes in Computing Science, Proceedings of the 27th European Conference on Information Retrieval* (pp. 200-214). Santiago de Compostela, Spain.

He, B., & Ounis, I. (2007). Setting Per-field Normalisation Hyper-parameters for the Named-page Finding Search Task, *Proceedings of the 29th European Conference on Information Retrieval (ECIR07).* . Rome, Italy.

Hersch, W., & al., e. (2004). TREC 2004 Genomics Track Overview, *Thirteenth Text Retrieval Conference*.

Hersch, W., & al., e. (2005). TREC 2005 Genomics Track Overview, *TREC 2005 Genomics Track Overview*.

Hersh, W., Cohen, A., Roberts, P., & Rekapalli, H. K. (2006). TREC 2006 Genomics Track Overview, *Online proceedings of the 2006 Text Retrieval Conference*.

Hersh, W., Yang, Bhupatiraju, R. T., Cohen, A., & Roberts, P. (2005). TREC 2005 Genomics Track Overview, *Online proceedings of the 2005 Text Retrieval Conference*.

Hiemstra, D. (1999). *A linguistically motivated probabilistic model of information retrieval.* Paper presented at the Research and Advanced Technology for Digital Libraries - Second European Conference, ECDL'98.

Hirschman, L., & Gaizauskas, R. (2002). Natural language question answering: the view from here. *Natural Language Engineering, 7*(04), 275-300.

Hirschman, L., Light, M., Breck, E., & Burger, J. D. (1999). Deep Read: a reading comprehension system, *Proceedings of 37th Annual conference of the Association for Computational Linguistics* (pp. 325-332): Association for Computational Linguistics.

Hofmann, T. (1999). Probabilistic latent semantic indexing. *Proceedings of the 22nd Annual International SIGIR Conference on Research and Development In Information Retrieval*.

Humphreys, K., Gaizauskas, R., Hepple, M., & Sanderson, M. (1999). University of Sheffield TREC-8 Q&A System, *Online proceedings of the 1999 Text Retrieval Conference*.

Jelinek, F., & Mercer, R. L. (1980). Interpolated estimation of Markov source parameters from sparse data, *Proceedings of the Workshop on Pattern Recognition in Practice*. Amsterdam, Netherlands.

Jelinek, F., Mercer, R. L., Bahl, L. R., & Baker, J. K. (1977). Perplexity—a measure of the difficulty of speech recognition tasks. *Journal of the Acoustical Society of America, 62*(S1), S63.

Jin, R., & Hauptmann, A. (2001). Learning to Select Good Title Words: An New Approach based on Reversed Information Retrieval. *Proceedings of the Eighteenth International Conference on Machine Learning*, 242 - 249

Jurafsky, D., & Martin, J. H. (2000). *Speech and language processing*: Prentice Hall Upper Saddle River, NJ.

Kalt, T. (1996). *A New Probabilistic Model of Text Classification and Retrieval*. Amherst, Massachusetts: University of Massachusetts.

Katz, S. (1987). Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics Speech and Signal Processing, 35*(3), 400-401.

Khalid, M., & Verberne, S. (2008). Passage Retrieval for Question Answering using Sliding Windows *Proceedings of COLING 2008 IR4QA Workshop*. Manchester, UK.

Kozareva, Z., Vázquez, S., & Montoyo, A. (2006). Adaptation of a Machine-learning Textual Entailment System to a Multilingual Answer Validation Exercise, *On-line Proceedings of CLEF 2006*.

Kraaij, W., & Spitters, M. (2003). Language Models for Topic Tracking. In W. B. Croft & J. Lafferty (Eds.), *Language Models for Information Retrieval*: Kluwer Academic Publishers.

Kullback, S., & Leibler, R. A. (1951). On Information and Sufficiency. *The Annals of Mathematical Statistics, 22*(1), 79-86.

Kurland, O., Lee, L., & Domshlak, C. (2005). *Better than the real thing?: iterative pseudo-query processing using cluster-based language models.* Paper presented at the Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, Salvador, Brazil.

Lafferty, J., & Zhai, C. (2001). Document language models, query models, and risk minimization for information retrieval, *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 111-119). New Orleans, Louisiana: ACM Press.

Lavrenko, V., Choquette, M., & Croft, W. B. (2002). *Cross-Lingual Relevance Models.* Paper presented at the Proceedings of the 25th annual international ACM SIGIR Tampere, Finland.

Lavrenko, V., & Croft, W. B. (2001). Relevance based language models. *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, 120-127.

Lehnert, W. (1977). *The Process of Question Answering*: Lawrence Erlbaum Associates.

Li, F., Zhang, X., & Zhu, X. (2008). Answer Validation by Information Distance Calculation, *Proceedings of COLING 2008 IR4QA Workshop*. Manchester, UK.

Li, M., & Vitanyi, P. (1997). *An introduction to Kolmogorov complexity and its applications*: Springer.

Li, X. (2005). Improving the Robustness of Relevance Based Language Models. *CIIR Technical Report*.

Liu, X., & Croft, W. (2004). *Cluster-based retrieval using language models.* Paper presented at the Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, Sheffield, UK.

Liu, X., & Croft, W. B. (2005). Statistical Language Modeling For Information Retrieval.

Macdonald, C., Lioma, C., & Ounis, I. (2007). Terrier takes on the non-English Web, *Proceedings of iNEWS'07 SIGIR Workshop*. Amsterdam, The Netherlands.

Magnini, B., Negri, M., Pervete, R., & Tanev, H. (2002). Comparing statistical and content-based techniques for answer validation on the web. *Proceedings of the VIII Convegno AI* IA*.

Magnini, B., Negri, M., Prevete, R., & Tanev, H. (2002a). *Is it the right answer? exploiting web redundancy for answer validation.* Paper presented at the Association for Computational Linguistics (ACL) 2002, Philadelphia. PA, USA.

Magnini, B., Negri, M., Prevete, R., & Tanev, H. (2002b). Is It the Right Answer? ExploitingWeb Redundancy for Answer Validation, *Association for Computational Lingustistics (ACL) 2002* (pp. 425-432). Philadelphia, PA.

Mann, G. S. (2001). A Statistical Method for Short Answer Extraction, *Proceedings on Association for Computational Linguistics Workshop on Open-Domain Question Answering* (pp. 23-30).

Manning, C. D., Raghavan, P., & Schutze, H. (2007). *Introduction to Information Retrieval*: Cambridge University Press.

Manning, C. D., & Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*: The MIT Press.

Meola, M. (2004). Chucking the Checklist. *portal: Libraries and the Academy, 4*, 331-344.

Metzger, M. (2007). Making sense of credibility on the Web: Models for evaluating online information and recommendations for future research. *JOURNAL-AMERICAN SOCIETY FOR INFORMATION SCIENCE AND TECHNOLOGY, 58*(13), 2078.

Miller, D. R. H., Leek, T., & Schwartz, R. M. (1999). A hidden Markov model information retrieval system. *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, 214-221.

Miller, G. A. (1995). WordNet: A Lexical Database for English. *Communications of the ACM, 38*(11), 39-41.

Mitra, M., Singhal, A., & Buckley, C. (1998). *Improving automatic query expansion.* Paper presented at the Proceedings of the 21st annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Melbourne, Australlia.

Moldovan, D., Clark, C., & Harabagiu, S. (2003). COGEX: a logic prover for question answering. *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, 1*, 87-93.

Moldovan, D., & Harabagiu, S. (2000). LASSO: A Tool for Surfing the Answer Net, *Online proceedings of the 2000 Text Retrieval Conference*.

Moldovan, D., Harabagiu, S., Girju, R., Morarescu, P., Lacatusu, F., Novischi, A., et al. (2002). *LCC Tools for Question Answering.* Paper presented at the Online proceedings of the 2002 Text Retrieval Conference.

Molla-Aliod, D., Berri, J., & Hess, M. (1998). *A Real World Implementation of Answer Extraction.* Paper presented at the Proceedings of the 9th International Conference on Database and Expert Systems.

Monz, C. (2004). Minimal Span Weighting Retrieval for Question Answering, *ACM SIGIR Workshop on Information Retrieval for Question Answering*.

Murdock, V. (2006). *Aspects of Sentence Retrieval.* University of Massachusetts, Amherst, Massachusetts.

Murdock, V., & Croft, W. B. (2004). Simple translation models for sentence retrieval in factoid question answering, *ACM SIGIR Workshop on Information Retrieval for Question Answering* (Vol. 2004).

Negri, M. (2004). Sense-based blind relevance feedback for question answering, *ACM SIGIR Workshop on Information Retrieval for Question Answering*.

Ogilvie, P., & Callan, J. (2003). Using Language Models for Flat Text Queries in XML Retrieval, *INEX 2003 Workshop Proceedings*.

Ogilvie, P., & Callan, J. (2006). Parameter Estimation for a Simple Hierarchical Generative Model for XML Retrieval. In *Advances in XML Information Retrieval and Evaluation* (Vol. 3977, pp. 211): Springer.

Ogilvie, P., & Callan, J. P. (2001). Experiments Using the Lemur Toolkit, *Online proceedings of the 2001 Text Retrieval Conference*.

Ounis, I., Amati, G., Plachouras, V., He, B., Macdonald, C., & Johnson, D. (2006). Terrier Information Retrieval Platform, *Proceedings of ACM SIGIR'06 Workshop on Open Source Information Retrieval (OSIR 2006)}, .* Seattle, Washington, USA.

Peters, C. (2006). What happened in CLEF 2006 Introduction to the Working Notes, *Online Proceedings of CLEF 2006.* http://www.clef-campaign.org/2006/working_notes/workingnotes2006/petersCLEF2006.pdf.

Peters, C. (2007). What happened in CLEF 2007 Introduction to the Working Notes, *Online Proceedings of CLEF 2007.* http://www.clef-campaign.org/2007/working_notes/CLEF2007WN-Contents.html.

Peters, C. (2008). What happened in CLEF 2008 Introduction to the Working Notes, *On-line Proceedings of CLEF 2008*. http://www.clef-campaign.org/2008/working_notes/CLEF2008WN-Contents.html.

Ponte, J. M., & Croft, W. B. (1998). *A language modeling approach to information retrieval.* Paper presented at the 21st annual international ACM SIGIR conference on Research and development in information retrieval Melbourne, Australia.

Roberts, I., & Gaizauskas, R. (2004). Evaluating Passage Retrieval Approaches for Question Answering, *Proceedings of 26th European Conference on Information Retrieval*. University of Sunderland, U.K.

Robertson, S., & Jones, K. (1997). *Simple proven approaches to text retrieval*: Cambridge University Computer Laboratory.

Robertson, S. E. (1997). The Probability Ranking Principle in IR. In K. Spärck Jones & P. Willett (Eds.), *Readings in Information Retrieval* (Vol. Multimedia Information And Systems Series pp. 281-286): Morgan Kaufman.

Robertson, S. E., Walker, S., & Beaulieu, M. (2000). Experimentation as a way of life: Okapi at TREC (Vol. 36, pp. 95-108): Elsevier Ltd.

Rodrigo, A., Peñas, A., & Verdejo, F. (2006). The Effect of Entity Recognition in Answer Validation, *On-line Proceedings of CLEF 2006*.

Rosenfeld, R. (2000). Two decades of statistical language modeling: where do we go from here? *Proceedings of the IEEE, 88*(8), 1270-1278.

Saggion, H., Gaizauskas, R., Hepple, M., Roberts, I., & Greenwood, M. (2004). Exploring the performance of boolean retrieval strategies for open domain question answering, *ACM SIGIR Workshop on Information Retrieval for Question Answering (IR4QA)*. .

Salton, G., & McGill, M. J. (1986). *Introduction to Modern Information Retrieval*: McGraw-Hill, Inc. New York, NY, USA.

Schlaefer, N., Gieselmann, P., Schaaf, T., & A., W. (2006). A Pattern Learning Approach to Question Answering within the Ephyra Framework, *In Proceedings of the Ninth International Conference on TEXT, SPEECH and DIALOGUE (TSD)*.

Scott, S., & Gaizauskas, R. (2000). *University of Sheffield TREC-9 Q&A*. Paper presented at the Online proceedings of the 2000 Text Retrieval Conference.

Shannon, C. E. (1951). Prediction and entropy of printed English. *Bell System Technical Journal, 30*(1), 50-64.

Simmons, R. F. (1965). Answering English questions by computer: a survey. *Communications of the ACM, 8*(1), 53-70.

Simmons, R. F. (1970). Natural language question-answering systems: 1969. *Communications of the ACM, 13*(1), 15-30.

Smucker, M., & Allan, J. (2007). *An Investigation of Dirichlet Prior Smoothing's Performance Advantage*. Amherst, Massachusetts: University of Massachusetts, Amherst.

Song, F., & Croft, W. B. (1999). A general language model for information retrieval. *Proceedings of the eighth international conference on Information and knowledge management*, 316-321.

Spärck Jones, K. (2004). *Language modelling's generative model: Is it rational*: Working paper, Computer Laboratory, University of Cambridge.

Srihari, R., & Li, W. (2000). Information extraction supported question answering, *Online proceedings of the 2000 Text Retrieval Conference*.

Stoyanchev, S., Song, Y. C., & Lahti, W. (2008). Exact Phrases in Information Retrieval for Question Answering, *Proceedings of COLING 2008 IR4QA Workshop*. Manchester, UK.

Strohman, T., Metzler, D., Turtle, H., & Croft, W. B. (2004). Indri: A language model-based search engine for complex queries. *Proceedings of International Conference on New Methods in Intelligence Analysis*.

Strohman, T., Metzler, D., Turtle, H., & Croft, W. B. (2005). Indri: A language model-based search engine for complex queries, *presented as a poster at the International Conference on Intelligence Analysis*. McLean, VA.

Tatu, M., Iles, B., & Moldovan, D. (2006). Automatic Answer Validation using COGEX *On-line Proceedings of CLEF 2006*.

Téllez-Valero, A., Juárez-González, A., Montes-y-Gómez, M., & Villaseñor-Pineda, L. (2008). INAOE at QA@CLEF 2008: Evaluating Answer Validation in Spanish Question Answering, *On-line Proceedings of CLEF 2008*.

Téllez-Valero, A., Montes-y-Gómez, M., & Villaseñor-Pineda, L. (2007). INAOE at AVE 2007: Experiments in Spanish Answer Validation, *On-line Proceedings of CLEF 2007*.

Tiedemann, J., & Mur, J. (2008). Simple is Best: Experiments with Different Document Segmentation Strategies for Passage Retrieval *Proceedings of COLING 2008 IR4QA Workshop*. Manchester, UK.

Trieschnigg, D., Kraaij, W., & Schuemie, M. (2006). Concept Based Document Retrieval for Genomics Literature, *Online proceedings of the 2006 Text Retrieval Conference*.

Usunier, N., Amini, M. R., & Gallinari, P. (2004). Boosting Weak Ranking Functions to Enhance Passage Retrieval For Question Answering *ACM SIGIR Workshop on Information Retrieval for Question Answering (IR4QA)*.

van der Plas, L., & Tiedemann, J. (2008). Using Lexico-Semantic Information for Query Expansion in Passage Retrieval for Question Answering *Proceedings of COLING 2008 IR4QA Workshop*. Manchester, UK.

Voorhees, E. M. (1999). The TREC-8 Question Answering Track Report, *Online proceedings of 1999 Text Retrieval Conference*.

Voorhees, E. M. (2005a). *Overview of the TREC 2005 Question Answering Track.* Paper presented at the Online proceedings of the 2005 Text Retrieval Conference.

Voorhees, E. M. (2005b). Overview of the TREC 2005 Robust Retrieval Track. *NIST Special Publication*.

Voorhees, E. M., & Harman, D. (1999). Overview of the Eighth Text REtrieval Conference (TREC-8), *Online proceedings of 1999 Text Retrieval Conference*.

Voorhees, E. M., & Harman, D. (2005). *TREC: Experiment and Evaluation in Information Retrieval*: The MIT Press.

Voorhees, E. M., & Harman, D. K. (2005). *TREC: Experiment and Evaluation in Information Retrieval (Digital Libraries and Electronic Publishing)*: The MIT Press.

Wang, R., & Neumann, G. (2007). DFKI–LT at AVE 2007: Using Recognizing Textual Entailment for Answer Validation, *On-line Proceedings of CLEF 2007*.

Wang, R., & Neumann, G. (2008). Information Synthesis for Answer Validation, *On-line Proceedings of CLEF 2008*.

White, K., & Sutcliffe, R. F. E. (2004). Seeking an Upper Bound to Sentence Level Retrieval in Question Answering, *ACM SIGIR Workshop on Information Retrieval for Question Answering (IR4QA)*.

Winograd, T. (1972). *Understanding Natural Language*: Academic Press, Inc. Orlando, FL, USA.

Witten, I. H., & Bell, T. C. (1991). The zero-frequency problem: estimating the probabilities of novel events in adaptive text compression. *IEEE Transactions on Information Theory, 37*(4), 1085-1094.

Xu, J., & Croft, W. (1999). *Cluster-based language models for distributed retrieval*.

Zhai, C., & Lafferty, J. (2001). Model-based Feedback in the Language Modeling Approach to Information Retrieval, *Proceedings of the tenth international*

conference on Information and knowledge management (pp. 403-410). Altanta, GA: ACM Press New York, NY, USA.

Zhai, C., & Lafferty, J. (2004). A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems (TOIS), 22*(2), 179-214.

Zhang, J., Ghahramani, Z., & Yang, Y. (2005). A probabilistic model for online document clustering with application to novelty detection. In Y. W. Lawrence K. Saul, Leon Bottou (Ed.), *Advances in Neural Information Processing Systems 17* (Vol. 17, pp. 1617–1624): MIT Press.

Zhou, X., Hu, X., Zhang, X., Lin, X., & Song, I. Y. (2006). Context-sensitive semantic smoothing for the language modeling approach to genomic IR, *29th annual international ACM SIGIR* (pp. 170-177). Seattle, WA, USA.

Zipf, G. K. (1949). *Human behavior and the principle of least effort*: Addison-Wesley Press Cambridge, Mass.

Zweigenbaum, P. (2003). Question answering in biomedicine, *EACL 2003 Workshop on Natural Language Processing for Question Answering*.

APPENDIX A: AQUAINT DOCUMENT FROM THE NYT DATASET

The document below is considered to include text that correctly supports the following Question/Answer pair from the TREC 2006 dataset:

Target: Warren Moon

Question: What position does Warren Moon play?

Answer: Quarterback

The text segment that supports the Question/Answer pair is marked in boldface.

<DOC>
<DOCNO> NYT20000107.0002 </DOCNO>
<DOCTYPE> NEWS STORY </DOCTYPE>
<DATE_TIME> 2000-01-07 00:04 </DATE_TIME>
<HEADER>
A9103 Cx1f; ttj-z
u s BC-FBN-THECHOKE-HNS LR; 01-07 2245
</HEADER>
<BODY>
<SLUG> BC-FBN-THECHOKE-HNS </SLUG>

(For use by New York Times News Service clients)
By JOHN McCLAIN
c. 1999, Houston Chronicle
BUFFALO, N.Y. _ When the Buffalo Bills found out this week they would play the Tennessee Titans in the first round of the playoffs, the wild-card game rekindled memories of the greatest game in franchise history, a game forever etched in the psyche of sports fans in Western New York.
<TEXT>
<P>
The Bills' 41-38 overtime victory over the Houston Oilers is famous in New York and infamous in Texas. Buffalo made the biggest comeback in NFL history, overcoming a 35-3 third-quarter deficit.
</P>
<P>
Last week, the NFL announced that game had been voted the fifth-most memorable in league history.
</P>
<P>
The timing was ideal.
</P>
<P>
The Bills are playing at Adelphia Coliseum on Saturday morning in the first game of the weekend. Although the Oilers moved to Nashville and became the Titans, the first playoff game between these teams since Jan. 3, 1993, has caused local and national media to focus on that last playoff game.
</P>
<P>

If you were a fan of the Oilers, you will never forget the game that became known as ``The Choke.''
</P>
<P>
If you are a fan of the Bills, you will never forget the game that became known as ``The Comeback.''
</P>
<P>
``That game was a defining moment in the Buffalo Bills' history,'' Buffalo receiver Andre Reed said this week. ``It was such a special game for everyone who played and everyone who watched it.''
</P>
<P>
Reed, who became the second-leading receiver in NFL history in the last game of the regular season, caught three touchdown passes to ignite the comeback, which was not televised in Buffalo because the game didn't meet the NFL blackout deadline.
</P>
<P>
Actual attendance was 75,141, about 4,000 shy of capacity at what was then Rich Stadium but has been renamed Ralph Wilson Stadium after the team's owner. At halftime, fans headed for the exits. In the third quarter, the stadium was only half full.
</P>
<P>
By the end of the game, every seat was taken and the parking lot was overflowing with fans who wanted to be part of a history-making experience.
</P>
<P>
The Bills had lost the last game of the regular season to the Oilers in the Astrodome. Quarterback Jim Kelly and outside linebacker Cornelius Bennett didn't play because of injuries. Coach Marv Levy was replacing Kelly with Frank Reich.
</P>
<P>
The Oilers, coached by Jack Pardee, led 28-3 at halftime. **Quarterback Warren Moon threw four touchdown passes**, two to Haywood Jeffires and one each to Webster Slaughter and Curtis Duncan. Buffalo's only score came on Steve Christie's 36-yard field goal.
</P>
<P>
On the Bills' first series of the third quarter, safety Bubba McDowell intercepted Reich and returned it 58 yards for a touchdown to make it 35-3. At that point, the Bills' comeback began. And it began innocently enough with a one-yard run by Kenneth Davis.
</P>
<P>
When Buffalo recovered an onside kick the Oilers weren't prepared for, the Bills became excited. After Reich threw a 38-yard touchdown pass to Don Beebe, the Bills trailed 35-17. The touchdown shouldn't have counted, because Beebe ran out of bounds and came back in, which should have made him an ineligible receiver, but the infraction went uncalled.
</P>
<P>

From that point, Murphy's Law took over for the Oilers. They did nothing right. Reich threw three touchdown passes to Reed to give Buffalo a 38-35 lead late in the game. The Oilers had to drive to the Bills' 12 for a chance to win. Moon's passes into the end zone were incomplete, and Al Del Greco's 26-yard field goal with 12 seconds remaining tied the score.
</P>
<P>
In overtime, cornerback Nate Odomes intercepted Moon, and Christie won the game with a 32-yard field goal.
</P>
<P>
``Wow! Are all playoff games like this?'' Christie said in the dressing room after participating in the first playoff game of his career.
</P>
<P>
No matter where you reside, whether in the Buffalo or Houston area, you probably have a story about that incredible afternoon that, in Bills lore, was ``Un-Bill-ievable'' as the headline in the Buffalo News blared the next day.
</P>
<P>
Over the last seven years, you've probably heard a thousand stories from Oilers fans. Now, here are some stories told from the other side _ from members of the Bills organization and their fans.
</P>
<P>
Steve Tasker, former Bills receiver and special-teams star, played in the game. He is now an NFL analyst for CBS.
</P>
<P>
``In our part of the country,'' Tasker notes, ``that game is like, `Where were you when JFK was shot?' But it's, `Where were you when the Bills came back to beat the Oilers?'
</P>
<P>
``When they had us down 35-3, we thought it was over. We were coming off two Super Bowls in a row, and we had a lot of pride, and even though the game was over, we wanted to leave with some dignity if we could.
</P>
<P>
``When we made it 35-17, I thought, `Well, at least we won't get blown out now and look like total idiots.' Then we scored again. And again. We went from trying to preserve our dignity to trying to be respectful to trying to win. It seemed like one supernatural thing after another kept happening. The Oilers had turned off the afterburners, and they couldn't turn them back on.
</P>
<P>
``I remember looking into the stands in the third quarter and seeing so many empty seats. But when we made the comeback, people were climbing the fence trying to bust back into the stadium.
</P>
<P>
``One thing that was so special about that game _ about our

team, really _ is that after games, the whole team would go to Jim Kelly's house. Jim has a great house. His basement is like a huge bar with pool tables and big-screen TVs. We got to Jim's house and watched the game, and what a feeling of euphoria that was.
</P>
<P>
``You know, that game was so big in Western New York that it became ingrained in the consciousness of everyone. It became legendary, woven in the fabric of the Buffalo Bills. It became bigger than the Ice Bowl and the Immaculate Reception in our part of New York.''
</P>
<P>
John Butler is the Bills' general manager. He was the personnel director in 1992.
</P>
<P>
``I hate to admit it,'' Butler says, ``but I remember thinking, `It's been a great year, and they just got us today.' When they made it 35-3, I got an empty feeling in the pit of my stomach. I started thinking about bringing in some players in the offseason and what we were going to do with the roster, things you always think about at that time of season.
</P>
<P>
``You know, it was a game of emotion like I've never experienced. When we scored a couple of touchdowns, I thought how proud I was that the guys had the heart to go down fighting. Then, when we went ahead, and they were on their last drive with a chance to win, I said to myself, `Oh, God. To come this far and not win would be so bitter. It'd be too tough to take.'
</P>
<P>
``Afterward, it was phenomenal. People were singing in the dressing room and celebrating like I'd never seen. I couldn't sleep that night. I had no strength left because I was emotionally drained. The game was just a blur.
</P>
<P>
``The next morning, I had to get on a plane and go to the East-West Shrine Game in Palo Alto (Calif.). When I got out there, everyone wanted to talk about the game. I tried hard to be humble.''
</P>
<P>
Thurman Thomas plays running back for the Bills. The Houston native started the playoff game but was injured in the first half and watched the rest of the game on the sideline.
</P>
<P>
``My family back in Houston was watching the game on television,'' Thomas remembers. ``When we got so far behind, my mother left and went to the store. She was gone and didn't know we had come back. When she came home, she asked my dad what the final score was, how bad the Oilers had beaten us, and when he told her what had happened, she didn't believe it at first. When they convinced her it was true, she started crying.

</P>
<P>
``When I called home that night, she said, `Oh, baby, I'm sorry. I'm sorry I left.' I told her it was OK.
</P>
<P>
``To tell you the truth, I didn't think there was any way we could come back and win. I was on the bench thinking about when I'd be leaving Buffalo and coming home to Houston and how much grief I'd have to take from my family and friends.
</P>
<P>
``I still can't believe we actually won. Sometimes it just doesn't seem real. People seem to forget that we had to win at Pittsburgh and Miami to go to our third straight Super Bowl. After we lost the Super Bowl (52-17 to Dallas), I went home to Houston a couple of days later.
</P>
<P>
``I was prepared for everyone to give me a hard time about us losing another Super Bowl, but the subject hardly came up. All they wanted to talk about was the Oilers game. That was fine with me. I preferred to talk about that game instead of the Super Bowl.''
</P>
<P>
Scott Berchtold is the Bills' vice president of communications. He was the director of media relations in 1992.
</P>
<P>
``I've heard so many stories over the years,'' Berchtold says. ``I think 250,000 people will swear they were at that game. I've heard so many stories from people who left. One guy even said he left in the second quarter and drove 90 miles home, then turned around and sped back in time for overtime.
</P>
<P>
``One thing that really stands out for me is that at halftime all the Houston writers were getting on the telephone and making reservations for the next game in Pittsburgh. But I remember thinking that we still had a chance. I even told one guy that, and he looked at me like I was crazy.
</P>
<P>
``It was 28-3 at halftime. I asked one of my interns to look up the biggest comeback in playoff history. When they scored to make it 35-3, I started thinking about things like the end-of-season interviews the next day. When we scored and got the onside kick and scored again, the intern told me that if we won, it would be the biggest comeback in NFL history, not just NFL playoff history. I didn't tell anyone for a while. I didn't want to jinx us.
</P>
<P>
``When I think back, I can still see everything going on in the press box. When we'd score, there'd be a collective roar from the media on the Buffalo side of the press box. It wasn't a cheer, just something like a disbelieving roar. At the same time, there'd be a collective groan on the Houston side. We had to announce three

times the NFL rule that there was no cheering in the press box. After a while, we just gave up.
</P>
<P>
``During practice last week, the televisions in the dressing room were showing parts of the comeback, because it had been voted the fifth-most memorable game in NFL history. The players just stopped and watched it. The young guys all knew that game. They started talking with (defensive end) Bruce (Smith) and Thurman and Andre about it. Rather than their teammates, the young guys were looking at them as heroes again.
</P>
<P>
``Another thing I remember is that at halftime, our backup quarterback, Gale Gilbert, reminded Frank Reich that he had engineered the biggest comeback in college history (for Maryland vs. Miami in 1984). Frank kind of nodded, like, `Yeah, right.'
</P>
<P>
``During the comeback, we had a fourth-down play. Marv had wanted to kick the field goal, but Frank talked him out of it on the sideline. Frank convinced Marv that he had a play that would work. It was a pass to Andre down the middle. Marv changed his mind. And it did work. Not only did we get a first down, but Andre caught the pass down the middle for the touchdown.
</P>
<P>
``After the game, Frank was at his locker room doing more interviews. I remember him telling the media that, other than getting married and having his children, it was the greatest day in his life. And I remember Kelly standing behind the camera trying to break him up by putting his finger down his throat as if Frank was making Jim gag with those kinds of remarks.
</P>
<P>
``After the game, Charles Osgood wrote a poem about the comeback and read it on CBS Radio. I still have a tape of it. It was about never giving up in life. It was almost like a Casey at the Bat for football. He used the Buffalo Bills as an example of guys that refused to give up against tremendous odds.''
</P>
<P>
Vic Carucci is a writer for NFL Insider. He was the Bills beat man for the Buffalo News in 1992.
</P>
<P>
``When the Oilers led 35-3, I was talking to my editor about plans for Marv's postseason press conference the next day,'' Carucci says. ``We had to worry about the game, plus our end-of-the-season wrapup. When you're covering a blowout, you don't tend to take notes as thoroughly as usual, and when the Bills started scoring, I really had to scramble.
</P>
<P>
``When the Bills started coming back, I was awestruck. When they looked like they might actually have a chance to win, we had to scramble. We had to change our coverage. Every time the Bills did

something right and the Oilers did something wrong, I had to blink my eyes to make sure it was really happening.
</P>
<P>
   ``We didn't really get ourselves together until after we came back upstairs and had a meeting to see what we had and how we were going to use it. People remember our headline _ `Un-Bill-ievable.'
</P>
<P>
   ``I've heard so many interesting stories. One friend left, and when he got to his subdivision, he saw cars turning around. He thought there was a wreck. He was so mad at the Bills, he didn't have the radio on. He turned it on to try to find out about the accident, and every station was talking about the comeback. He said he turned around immediately and came right back. I know a lot of people who did the same thing but just won't admit it.''
</P>
<P>
   $$$
</P>
</TEXT>
</BODY>
<TRAILER>
NYT-01-07-00 0004EST QL;
</TRAILER>
</DOC>

VITA

Education

2009 Drexel University

Ph.D., Information Science and Technology

- Thesis: Language Modeling Approaches to Question Answering
- Advisor: Dr. Hyoil Han, Drexel University

1997 Rensselaer Polytechnic Institute

M.S., Computer Science

1995 Cornell University

B.S., Electrical Engineering

Publications

Refereed Conference Papers

- Banerjee, P. and Han, H. 2009. "From Question Context to Answer Credibility: Modeling Semantic Structures for Question Answering Using Statistical Methods," to appear in IKE 2009. Refereed.
- Banerjee, P. and Han, H. 2009. "Answer Credibility: A Language Modeling Approach to Answer Validation," (short paper) in NAACL-HLT 2009, Boulder, Colorado, USA. Refereed.
- Banerjee, P. and Han, H. 2009. "Modeling Semantic Question Context for Question Answering," in FLAIRS-22, Sanibel Island, Florida, USA. Refereed.
- Banerjee, P. and Han, H. 2008. "Incorporation of Corpus-Specific Semantic Information into Question Answering Context," (short paper) in CIKM 2008 - Ontologies and Information Systems for the Semantic Web Workshop Napa Valley, USA.. Refereed.
- Banerjee P., Hu X., Yoo I., Discovering the Wealth of Public Knowledge: An Approach to Early Threat Detection, Dept of Homeland Security R&D Conference, Boston, MA, May, 2005. Refereed.
- Yoo I., Banerjee P., Shelfer K., Hu X., Semantic Data Mining for Homeland Security: Architecture and First Steps, in Proceedings of the 2004 IEEE WI Semantic Web Mining and Reasoning Workshop. Refereed.
- Hu X., Yoo I., Banerjee P., Document Clustering and Summarization Using Biomedical Ontologies, in Proceedings of the 2004 IEEE ICDM Foundation of Data Mining Workshop. Refereed.

Refereed Journal Papers

- Banerjee, P. and Han, H. 2009. "Language Modeling Approaches to Information Retrieval," to appear in Journal of Computer Science and Engineering.

Refereed Book Chapters

- Banerjee P., Hu X., Yoo I., Semantic Data Mining, Encyclopedia of Data Warehousing and Data Mining, (Ed. John Wang), Idea Group Publishing, USA, 2005, 1010-1014. Refereed.

Other Publications

- Banerjee, P., Han, H., Drexel at TREC 2007: Question Answering, Question Answering Track of TREC 2007, NIST Special Publication.