SEARCHING QUESTION AND ANSWER ARCHIVES

A Dissertation Presented

by

JIWOON JEON

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

DOCTOR OF PHILOSOPHY

September 2007

Computer Science

SEARCHING QUESTION AND ANSWER ARCHIVES

A Dissertation Presented

by

JIWOON JEON

Approved as to style and content by:

W. Bruce Croft, Chair

James Allan, Member

Yanlei Diao, Member

Byung Kim, Member

Andrew Barto, Department Chair
Computer Science

To my parents
Hang-Jo Jeon and Sang-Ok Bae,

and to my life partner
Young-Hwa Kim,

for their endless love, encouragement and patience.

ACKNOWLEDGMENTS

I would like to thank my advisor Bruce Croft for his guidance and support. It was my honor and pleasure to work with the most respected person in the field. I could move forward without any hesitation because he was behind me all the time.

I would like to thank my formal advisor R. Manmatha. When I hardly knew anything about research, he kindly explained me so many things with patience. A large part of my Ph.D work was just applying what I learned from him for new tasks.

I would like to thank Dr. Joon-Ho Lee. This work was impossible without his strong support and encouragement. He also showed me the importance of seeing the wood behind the trees.

I'd also like to thank all of the people at CIIR who have made it such a great place to work.

This work was supported in part by NHN Corp. and the Center for Intelligent Information Retrieval. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect those of the sponsor.

ABSTRACT

SEARCHING QUESTION AND ANSWER ARCHIVES

SEPTEMBER 2007

JIWOON JEON

B.S., Comp.Sci., KOREA UNIVERSITY, 1997

M.S., Comp.Sci., UNIVERSITY OF MASSACHUSETTS AMHERST, 2004

Ph.D., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor W. Bruce Croft

Archives of questions and answers are a valuable information source. However, little research has been done to exploit them. We propose a new type of information retrieval system that answers users' questions by searching question and answer archives. The proposed system has many advantages over current web search engines. In this system, natural language questions are used instead of keyword queries, and the system directly returns answers instead of lists of documents. Two most important challenges in the implementation of the system are finding semantically similar questions to the user question and estimating the quality of answers. We propose using a translation-based retrieval model to overcome the word mismatch problem between questions. Our model combines the advantages of the IBM machine translation model and the query likelihood language model and shows significantly improved retrieval performance over the state of the art retrieval models. We also show that collections of question and answer pairs are good linguistic resources for learning reli-

able word-to-word translation relationships. To avoid returning bad answers to users, we build an answer quality predictor based on statistical machine learning techniques. By combining the quality predictor with the translation-based retrieval model, our system successfully returns relevant and high quality answers to the user.

TABLE OF CONTENTS

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

CHAPTER 1

INTRODUCTION

Many people think that question answering (QA) will be the next step beyond current search engines. People will use natural language questions to express their information needs in finer detail and the system will return answers instead of lists of relevant documents. However, open domain QA is known to be very difficult and even state of the art QA systems can answer only limited types of questions in small domains. It might not be possible to implement such systems in the near future unless we make a significant advance in artificial intelligence[1].

In this dissertation, we propose a new type of information system that behaves like an intelligent QA system but it does not rely on sophisticated semantic and contextual processing. The main idea is searching previously answered questions instead of generating answers that typical QA systems attempt. When we have a question, it is not hard to imagine that some one else might have posted the same question somewhere on the web and that the question might have already been answered. This kind of approach is feasible since it is possible to collect a large number of question and answer pairs from the web. Our approach converts the difficult QA problem into an information retrieval problem. We call this unique task of finding question and answer pairs as **Q&A Retrieval**.

While our final goal is to find answers, but the actual task is to find semantically similar questions to a given user question. This is not a trivial task because

[1] "Most AI researchers believe that new fundamental ideas are required, and therefore it cannot be predicted when human level intelligence will be achieved." - John McCarthy, 2004

semantically identical questions can be represented in many different ways. Therefore, our objective is to develop reliable similarity measures between questions that can overcome the word mismatch problem. We propose a machine translation-based information retrieval model to solve this problem.

The second challenge is in measuring the quality of answers as we want to return relevant as well as high-quality answers to the user. The quality problem is especially important when there are many answers to a given question. This happens in the case of popular questions where all the answers are relevant and we need to rank them in the order of their quality. We develop an answer quality prediction technique and successfully integrate it into our retrieval model.

In the remainder of this chapter, we present our contributions and provide a brief overview of this thesis.

Terminology Clarification

In this dissertation, "**QA**" and "**Q&A**" are used in different ways. "**QA**" is used to denote traditional question answering approaches that automatically construct answers from document collections or manually crafted knowledge databases. "**Q&A**" stands for a question and its associated answer. "**Q&A Service**" means human-based (community or expert) question answering services such as Yahoo! Answers[2] and Live QnA[3].

[2] `http://answers.yahoo.com`

[3] `http://qna.live.com`

1.1 Contributions

1.1.1 New type of Information System

We propose a new type of information system that can answer users' questions. This system has many advantages over current web search engines and has great potential for real world applications. Main advantages of the system are:

- Ability to handle natural language questions.
- Ability to handle various types of questions whose answers are facts, procedural steps and explanations.
- Does not rely on hand crafted rules or collection specific heuristics.
- Both the relevance and the quality of answers can be modeled.

1.1.2 New Translation-based Retrieval Model

We also propose a new translation-based retrieval model to handle the word mismatch problem. We recognize similarities and differences between the IBM translation model and the query likelihood language model and carefully combine both approaches. The proposed model consistently outperforms other state of the art models. We believe combination of machine translation and information retrieval can lead us to more advanced IR systems that can model semantics. This research can be regarded as early work in that direction. Important features of the propose model are:

- Model is based on statistical generative models: the query likelihood language model and a statistical machine translation model.
- Addresses the word mismatch problem.
- Can be used as general purpose information retrieval framework.
- An efficient implementation is possible.

1.1.3 New Document Quality Estimation Method

Estimating document quality is an important problem to return good quality answers to users. In this dissertation, we propose a framework that can estimate the quality of documents using non-textual features. Since the framework is based on statistical machine learning techniques and does not rely on collection or task specific heuristics, it can be used in any web service that tracks many non-textual features such as click counts and recommendation history. The advantage of the proposed method is that the quality score returned is a probability that can be easily integrated into other statistical frameworks like the language modeling approach to information retrieval. Interesting properties of the framework are:

- Can handle various types of non-textual features: monotonic, non-monotonic, integers and real numbers.
- Can process large number of features quickly.
- Robust to noisy features.
- Adding more features is easy.
- Quality estimations are probabilities, which can easily be integrated into other statistical models.
- Based on statistical machine learning and document classification methods.

1.1.4 Integration of Advances in Multiple Research Areas

Real world information retrieval systems have to consider multiple factors such as relevance of retrieved items, quality of contents (filtering out spam) and response times. However, most traditional information retrieval frameworks focus on only measuring relevance of contents. To satisfy multiple facets of real world information

retrieval problems, we need to exploit recent advances in related areas such as machine learning and natural language processing. In this dissertation, we use various techniques developed in multiple areas to build our system. The quality estimation is based on machine learning and document classification techniques and the word mismatch problem is addressed using machine translation techniques. All these component are based on statistical approaches, and therefore they can be seamlessly integrated under the language modeling framework for information retrieval. Our methods of integrating multiple component developed in multiple areas demonstrate how we can handle emerging information retrieval problems using advances in related research fields.

1.1.5 New Paraphrase Generation Method

To generate training samples for our system, we exploit unique properties of Q&A collections. Our idea is that if two answers are very similar then the corresponding questions should be semantically similar, even though the two questions are lexically very different. Using this idea, we can find a large number of semantically similar question pairs. One important advantage of this novel approach is that we can automatically find lexically very different paraphrases.

1.1.6 Utilizing Web as a Resource for Retireval

Another contribution is using web resources such as FAQs and Q&A pairs to train information retrieval systems. Our system learns word-to-word translation probabilities from Q&A archives and uses them to bridge the lexical chasm between questions. This is a successful demonstration of using web resources to improve IR systems. So far, most IR research has fallen under the category of unsupervised machine learning and paid little attention on exploiting these new types of text collections abundant on the web. Our work shows this kind of data driven-approach is feasible and promising.

1.2 Thesis Overview

We will define a novel information retrieval task, namely **Q&A retrieval** in **chapter 2**. The chapter starts by introducing Q&A collections available on the web. We discuss why Q&A retrieval is important and interesting. We also address challenges that we confront in implementing reliable and effective Q&A retrieval systems.

Chapter 3 is devoted to explaining our test collections. We introduce three Q&A collections and set of queries acquired from multiple sources. Criteria for relevance judgment are also presented. **Chapter 4** propose a new translation-based information retrieval model. The characteristics of the query likelihood language model and the IBM translation model are discussed and comparisons are drawn to the proposed model. In **Chapter 5**, we explain how to calculate word-to-word translation probabilities using Q&A collections. Various types of word relationships are introduced with examples. The method of generating training data is also discussed. **Chapter 6** presents our document quality predictor. We discuss why the quality prediction is important and how we can reliably estimate document (answer) quality using statistical methods. **Chapter 7** tests our approach on multiple real world collections. We compare our system with other state of the art information retrieval systems. Conclusions and future work are presented in **Chapter 8**.

CHAPTER 2

Q&A RETRIEVAL

In this chapter, we define the task of Q&A retrieval and explain why this task is interesting and important. We also discuss challenges that we encounter in solving this unique information retrieval problem. We begin this chapter by discussing various sources of Q&A pairs.

2.1 Question and Answer Archives

A huge number of questions answered by people exist today in electronic form. Many web sites have question and answer boards or FAQ (Frequently Asked Questions) pages. At the time of writing, we found more than 23 million web pages which have "FAQ" in their titles and more than 44 million web pages that have "FAQ" in their URLs from the Google search engine. Jijkoun and Rijke [27] claimed 76% of those web pages are true FAQ pages and each page contains on average 13 question and answer pairs. Soricut and Brill [62] also reported similar statistics. If we add other web pages that have "QA", "Q&A" or "Questions and Answers" in their titles or URLs, the number of pages becomes even larger.

Another important source of question and answer pairs are community-based question answering services where people answer other people's questions. These started as digital reference services such as MadSci Network and Ask Dr. Math, but have now become a popular part of Web search services. A large number of questions and answers can be easily collected from these services. For example, Yahoo! provides a

community-based question answering service[1], where users answer other users' questions for free. This service is extremely popular and Yahoo claimed that the service surpassed 60 million users and 160 million answers worldwide in less than one year. Similar services like Live QnA[2], AnswerBag[3], Wondir[4] and Naver Q&A[5] also have large numbers of question and answer pairs.

Question and answer archives are valuable information resources. In a survey done by Naver, they found almost half of all user information needs (excluding homepage-finding queries) could be satisfied by searching their Q&A archive that has over 60 million question and answer pairs. Jijkoun and Rijke [27] automatically collected 2.8 million FAQs from the web. By searching their FAQ collections, they could satisfy 36% of user queries, which are in the form of natural language questions, submitted to a web search engine. These results show the potential importance of Q&A archives as an information source.

2.2 Task Definition

Although Q&A collections are abundant and valuable, little research has been able to effectively utilize these collections. Most Q&A services use conventional information retrieval algorithms developed for document collections even though Q&A collections are different from document collections in many aspects such as length, structure, writing style, etc.

In this dissertation, we call the task of searching these collections **Q&A Retrieval**. Q&A retrieval is different from FAQ retrieval, QA (Question Answering)

[1]Yahoo Answers!, http://answers.yahoo.com

[2]http://qna.live.com

[3]http://www.answerbag.com/

[4]http://www.wondir.com

[5]http://kin.naver.com

Table 2.1. Problem Clarification: Q&A Retrieval

	Adhoc Retrieval	Question Answering	FAQ Retrieval	Q&A Retrieval
Query Type	keywords	question	question	question
Collection	document	document	FAQ	Q&A
Collection Size	large	large	small	large
Collection Quality	noisy	noisy	clean	noisy
Output	documents	answer	answer	answer
Application	Web search	Factoid Answering	Consumer Support	General Answering
Previous Work	Extensive	Extensive	some	very little

and adhoc document retrieval. Table 2.1 summarizes differences between these tasks. In our definition of Q&A retrieval, we assume as input natural language questions and ranked lists of question and answer pairs as output. The question part of a Q&A pair is important to users when they judge the relevance of the results. Therefore, the system should return the question part together with the answer part. This is the reason that we call this task Q&A retrieval instead of question retrieval or answer retrieval. Q&A archives include FAQ collections but are not limited to FAQs. Questions and corresponding answers collected from other sources such as Q&A boards or Q&A services can be part of the archives.

2.2.1 Advantages of Q&A Retrieval

Q&A retrieval systems have many advantages over traditional adhoc information retrieval systems. Q&A retrieval systems handle natural language questions unlike traditional retrieval systems which force users to come up with keywords that accurately describe their information needs. Q&A retrieval systems directly return answers instead of a list of relevant documents; so users can save time otherwise spent on browsing and summarizing lists of documents. Another significant advantage is that even complicated information needs such as asking for advice, a summary or an

opinion can be satisfied if there are matching questions in the Q&A archive. These types of queries are very hard to answer with current web search engines.

Automated question answering systems share some of the advantages of Q&A retrieval systems. However, implementation of high performance QA systems is very difficult and even state of the art QA systems can answer only limited types of factoid questions. FAQ retrieval systems are similar to Q&A retrieval systems, but their goal and collections are different. Typically, FAQ retrieval systems are specific to a domain and use small number of FAQs (usually less than a few hundred). These FAQs are maintained by domain experts and the quality of questions and answers are good. Q&A collections tend to be much broader in coverage and cannot benefit from domain experts.

2.2.2 Shortcomings of Q&A Retrieval

The most serious shortcoming of a Q&A retrieval system is that it can answer only previously asked questions. If there is no matching question in the archive, there is no way to answer this question automatically. This problem can be treated in multiple ways. Firstly, we can increase the chance of finding matching questions by increasing the size of the archive. The second method is to use supplemental web search engines. If there is no matching question, we can convert the question into a keyword query and forward it to web search engines and return the results to users. The method of last resort is posting the question to a community or expert-based question answering service and wait for answers from other users or experts. Figure 2.1 shows a possible architecture of future information retrieval systems that integrate Q&A retrieval systems with other supplemental components such as web search engines and community-based question answering services.

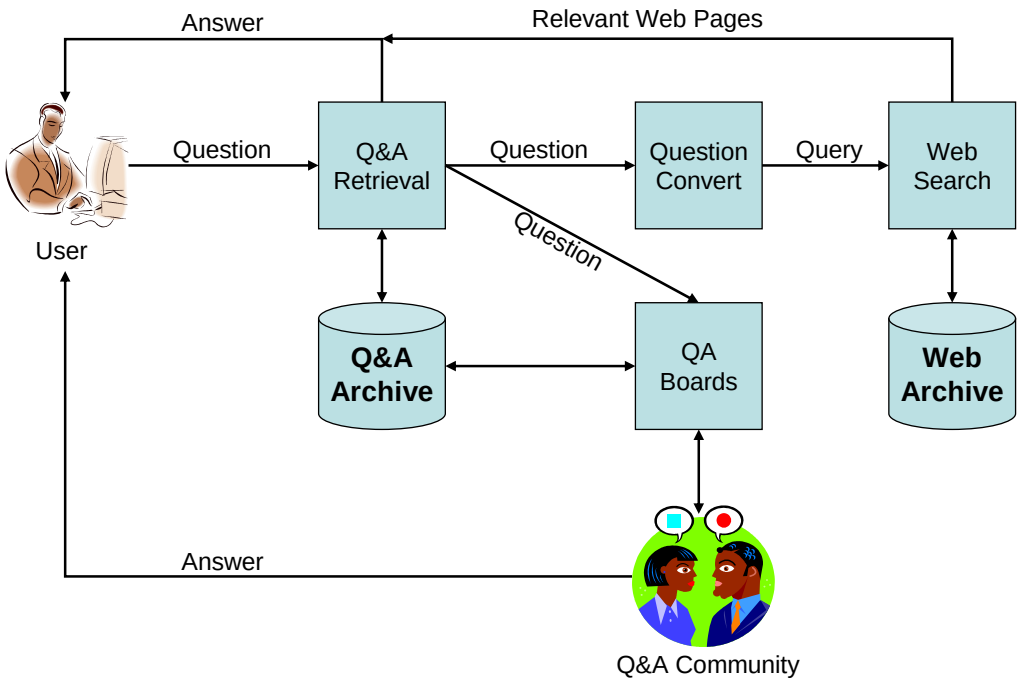


Figure 2.1. Example architecture of the Q&A retrieval system augmented with a web search engine and a community-based question answering service. If the Q&A retrieval system fails to find matching questions, the user question is forwarded to web search engines and Q&A services.

2.3 Challenges

In this section, we idnetify two main challenges in building effective Q&A retrieval systems. The first challenge is finding relevant question and answer pairs to the user question and the second is estimating the quality of answers. The answer quality is independent of the query and can be calculated in advance. Therefore, the estimation of quality can be separated from the estimation of relevance.

2.3.1 Finding Relevant Question and Answer Pairs

In Q&A retrieval, relevance of a Q&A pair is judged by the semantic similarity between the question part of the Q&A and the user query[6]. If the question part addresses the same information needs as the user query, we judge the Q&A pair as being relevant to the query. Our studies show that the answer part is always relevant to the question part. Therefore, we do not have to worry about the relevance of the answer part as long as the question part is relevant.

2.3.1.1 Importance of Question Parts

From the definition of the relevance, it is obvious that the question part is much more important than the answer part when we search Q&A collection. Previous research [13, 27] in FAQ retrieval gives similar indications. To prove our claim we performed a simple experiment.

For our experiment we used two test collections: the Wondir collection and the WebFAQ collection. The Wondir collection has 1 million Q&A pairs collected from the Wondir service while the WebFAQ collection contains 3 million FAQs automatically collected from the web using a specialized web crawler. Each collection has 50 test queries (questions). For each query, our annotators found relevant Q&A pairs using

[6]In our task, user queries are actually questions. In this thesis, we use **user query** and **user question** in the same meaning.

Table 2.2. Retrieval performance of searching individual fields. In both models, the best performance can be achieved by searching the question field. The answer field is least useful.

Collection	Search Field	MAP
Wondir	Question	0.3024
	Answer	0.1201
	Combined	0.3068
WebFAQ	Question	0.1924
	Answer	0.0671
	Combined	0.1921

the pooling method[7]. A detailed explanation about test collections is given in chapter 3.

In the first experiment, we removed all the answer parts from the collection and searched only the question parts using the popular query likelihood language model[8] [55] with Dirichlet smoothing [76]. In the second experiment, we removed all question parts and searched only the answer parts. The last experiment was combining both fields. We used the linear mixture model proposed by Ogilvie and Callan [52]. This method linearly combines two different language models at the word level and shows good performance in many different field combination tasks.

Table 2.2 shows retrieval results measured by the Mean Average Precision (MAP)[9]. The reported results are the best performance that each retrieval model achieves after parameter tuning. The combination approach has an additional parameter that controls the mixing of the two language models. In both collections, searching the question field is much more important than searching the answer field. Because of the significant performance gap between the two fields, the combination method does not

[7]Manually finding all the relevant document for a given query is expensive. The pooling method collects retrieval results from multiple search engines and makes a pool of documents from the top n retrieved documents from each system. Only the documents in the pool are manually judged.

[8]A detailed explanation about the model is in section 4.3.1.

[9]The definition of MAP is explained in section 7.1

help. The best combination results are achieved when we give the maximum weight (0.95) to the question field. These results show the answer field is not as useful for the task of finding relevant Q&A pairs[10]. Therefore, in this dissertation, we focus on similarity between the question and the query.

2.3.1.2 Word Mismatch Problem Between Questions

As shown in our experiments, the question part (field) is the key when we find relevant Q&A pairs. Therefore, the effectiveness of Q&A retrieval system depends on its ability to accurately measure semantic similarities between two questions: the user question and the question in the Q&A pair. This is not an easy task because the same questions can be written in many different ways. For example, "Is downloading movies illegal?" and "Can I share a copy of DVD online?" are semantically similar but lexically very different. No single word occurs in both questions. Most information retrieval algorithms rely on word overlap between the user query and documents and fail to capture the semantic relationship between the two questions. This word mismatch problem is more serious in Q&A retrieval because questions are much shorter than documents. Using synonyms or alternative expressions sometimes makes it impossible to retrieve relevant questions. We propose a machine translation-based retrieval algorithm to address the word mismatch problem. The advantage of a machine translation-based technique is that relationships between words are explicitly modeled.

2.3.2 Estimating Answer Quality

The quality of a Q&A retrieval systems depends on both the question part and the answer part. The following are examples of bad questions that can be found from community-based Q&A services.

[10]In our earlier work [25], when we used very short keyword queries (on average 2.1 words), we found the answer field to be helpful.

- What is one plus one?
- Who is more handsome than me?
- I am sad.
- All you immoral people know exactly where you are going, don't you?

Users cannot get any useful information by reading answers for these bad questions. We found that bad questions always lead to bad quality answers. Answers for these bad questions usually blame the questioner with short insulting words. Therefore, we decide to estimate only the quality of answers and consider it as the quality of the Q&A pair.

2.3.2.1 Importance of Quality Estimation

For popular questions, many relevant question and answer pairs can be found because these questions tend to be asked frequently in many different places. The quality of the answers vary even though they are all relevant to the user question. Obviously, it is very important to return question and answer pairs that have good answers. Therefore, we need to develop and integrate quality measures in Q&A retrieval.

Quality estimation is also required to filter out spam. The spam problem usually happens in community-based Q&A services and Q&A boards where anybody can answer any question. Some people just make fun of other people by posting some insulting or irrelevant answers. The following are a few examples found from the Wondir sevice.

Q: What is the minimum positive real number in Matlab?

A: It is equivalent to your IQ.

Q: What is new in Java2.0?

A: Nothing new.

Q: Can I get a router if I have a usb dsl modem?

A: Good question but I do not know.

We use document classification algorithms based on statistical machine learning to predict answer quality. We focus on exploiting non-textual features such as click counts and recommendation counts because they have strong correlation with quality. We rely on statistical approaches and calculate the probability of a good answer for a given Q&A pair. This probability can be easily integrated into other statistical models like the language modeling framework in information retrieval. Quality estimation is independent of user queries and can be done at indexing time. Chapter 6 presents our approach of estimating answer quality in greater detail.

CHAPTER 3

TEST COLLECTIONS

We use Q&A collections acquired from three difference sources. Two collections are from community-based question answering services and one is from the web. In this chapter, we explain how we built our test collections from these Q&A collections. Typically, a test collection consists of three components: a set of documents, a set of information needs (queries), and a set of relevance judgment. We used the pooling method to find relevant Q&A pairs for a given query. The following is a brief explanation about the method.

The Pooling Method

Finding all relevant documents in a collection for a given query is expansive because we have to read all the documents in the collection. The pooling method examines only top n documents returned from each search engine (or algorithm). The basic assumption is the pool contains enough number of relevant documents that can represent all the relevant documents in the collection. Typically annotators do not know which document is returned from which search engine and manually judge the relevance of documents in the pool. This method significantly saves the time and effort to build test collections. This method was initially outlined by Sparck-Jones and Van Rijsbergen in mid-1970s [29, 30] and used as a standard method for building test collections in the TREC experiments [21]. Zobel [80] investigated the reliability of this method with the TREC results and reported the relevance judgment was indeed reliable.

3.1 Wondir Collection

Wondir[1] is one of the earliest community-based QA service in the US. The service is free and no membership is required to use it. They claimed more than 100,000 different people had answered questions and more than 60,000 people were registered member of the community[2] at the time we acquired data from them.

Their aim is connecting people with questions to the people with answers in real time. When a user type a question, the question appears on the front page of the web site and other users answer the question in real time. Answers are usually short and succinct because they have to be returned in a short time. Although there is no limitation in topics, dominating questions are about human relationships.

3.1.1 Collection

We got about 1 million question and answer pairs from the service provider. The average question length is 27 words and the average answer length is 28. The following are typical Q&A pairs (un-edited) found in the collection.

- **Q:** boyfriend and i got into a little fight and then he exploded saying he needed to be alone for a long time and doesnt want to go out with me anymore. then he says he wasn;t serious, like a half hour later. whats up w this?
  A: sounds like he has a nasty temper. I'm sure he doesn't mean things when he's heated, but that doesn't make it right. Show him, the right way to ....

- **Q:** why do femal wear shorts, whats the big deal about wearing them ? I don't understand ? is it just everyone copying each other why they all wearing shorts
  A: because they are comfortable and attractive.

---

[1] http://www.wondir.com

[2] `http://www.wondir.com/wondir/jsp/news/pr_041905.htm`

- **Q:** what are good facial cleanzers

  A: noxema, clean and clear, clearisal. I recommend rotating, just incase you get immune to one of them

- **Q:** Where can I find used mercedesonline?

  A: www.kbb.com www.autotrader.com www.usedcars.com

The above examples show that spelling errors are very common because the user interface does not check for spelling errors when users enter questions or answers. Spelling errors worsen the word mismatch problem. The first two questions have little value because of the personal nature. The last two Q&A pairs have some value but because of the short answer length, they do not give enough information. Because of the bias to topics on personal problems and poor answer quality, the value of this collection as a knowledge database is not big. However, this collection reflects concerns and interests of real users in the community.

3.1.2 Queries

For this collection, we selected the first 50 queries from the 500 test queries used at the TREC9 QA track [68]. The QA track queries are questions chosen by NIST[3] from search engine logs. All questions are closed-class questions that ask fact-based short answers. The following is the brief explanation about the query generation process from the official report of the track.

> "For the TREC-9 track, NIST obtained two query logs and used those as a source of questions. An Encarta log, made available to NIST by Microsft, contained grammatical questions. The other log was a log of queries submitted to the Excite search engine on December 20, 1999. Since the Excite log contains relatively few grammatically well-formed

[3] `http://trec.nist.gov/data/qa/t9_qadata.html`

questions, the log was used as a source of ideas for NIST staff who created well-formed questions from query words without referring to the document collection."

We choose these queries because the Wondir collection contains many factoid questions asked by school students. The following is examples queries.

- Where is Belize located?
- How much folic acid should an expectant mother get daily?
- What type of bridge is the Golden Gate Bridge?
- What is the population of the Bahamas?
- Who invented the paper clip?

3.1.3 Relevance Judgment

We found 220 relevant Q&A pairs for the 50 queries using the pooling method. We ran our queries through multiple search engines including baseline retrieval models and our model. By changing parameter values, we could generate many different rank lists. Some runs only searched the question parts and other runs searched only the answer parts. Searching the answer parts helps find Q&A pairs that cannot be found by searching only the question parts because of the word mismatch problem. All Q&A pairs returned in the top 20 in any ranked list were added to the pool. The pool had 3,220 Q&A pairs in total. The following is the criteria that we used to decide the relevance of a Q&A pair.

- **Relevance Judgment Criteria**

  When we manually judged the relevance of Q&A pairs, we ignored the correctness of answers. As long as the question addresses the same information needs, we judged the Q&A pair to be relevant to the query. Some questions

Query: How many dogs pull a sled in the Iditarod?

#	Rel	Question
1	R	What is the fewest number of dogs that a team in the iditarod can run?
2	R	Who maney dogs can be run in the iditarod?
3	R	Howare the dogs arrranged on a dog sled?
4	N	How many checkpoints are on the Iditarod route?
5	N	What was the closest finish in the iditarod?
6	N	what are the people who control the dog sled called?
7	N	who is the youngest mushe to ever complete the iditarod

Table 3.1. Relevance Judgment Examples. Wondir Collection. (R: Relevant, N: Non-Relevant).

are about more specific topics while others are about broader topics given in the user query. In such cases, we measure the overlap of two information needs represented by the question and the query. If the overlap covers 50% of both information needs, then the question is judged as relevant. Questions containing many query terms were often judged as non-relevant because their information needs had little relation to the user query even though they were lexically very similar to the query.

Table 3.1 shows examples of the judgment results. The first question is closely related to the query and the user's information needs can be satisfied by reading the answers to this question. The third question was judged as relevant because the dog arrangement problem is closely related to the number of dogs. The second question has many spelling errors but it is obvious that the query and the user questions are the same. The last four questions were judged as non-relevant because they are different questions even though they are related to the Iditarod and contains many query terms like dog, sled, and Iditarod.

3.2 WebFAQ Collection

3.2.1 Collection

Jijkoun and Rijke [27] collected approximately 3 million FAQs from the web using specialized web crawlers and made them publicly available for research purposes[4]. They first found web pages that contain the word "FAQ" in their titles using the Google search engine. Then they used a few heuristic methods to automatically extract question and answer pairs from the web pages. Since the collection was constructed automatically, it contains detection errors and noise information but the collection has fewer spelling errors compared to the Wondir collection. The average answer length (101) is much longer than the average answer length (28) of the Wondir collection. The following is example FAQs in the collection.

> **Q:** What is the American Discovery Trail?
>
> **A:** The American Discovery Trail (ADT) is a new breed of national trail. part city, part small town, part forest, part mountains, part desert ? all in one trail. It is 6,300+ miles of adventure, discovery and fun, and ...

> **Q:** How do molds grow in my home?
>
> **A:** Once mold spores settle in your home, they need moisture to begin growing and digesting whatever they are growing on. There are ...

> **Q:** Anything else you want to say about yourself?
>
> **A:** Not on the first date.

> **Q:** How will our submissions be graded?
>
> **A:** Most points are given to the trace files and the reports. The trace files

[4] `http://ilps.science.uva.nl/Resources/WazDah/`

can show whether you did the exercises correctly. Some questions ...

Q: Do NOT send "unsubscribe", "remove", or other such requests.

A: Top of page | Return to FAQ

The first and the second examples show useful FAQs. The third and the fourth are FAQs that have vague questions. Most questions in the Wondir collection are self contained. Therefore, in most cases, we can understand the intent of users just by reading questions. However, some questions in FAQ collections are hard to understand unless we read whole FAQ pages. Such FAQs have little value when presented alone. The last example shows an example of a wrong FAQ generated by detection errors.

3.2.2 Queries and Relevance Judgment

The authors of the WebFAQ collection provided a set of sample queries with their FAQ collection. These queries are questions selected from the queries submitted to the MetaCrawler search engine[5]. These queries contain various types of questions whose answers are facts, procedural steps and explanations. The following is a brief description of the query set by the authors.

> "To this end, we obtained samples from query logs of the MetaCrawler search engine (http://www.metacrawler.com) during September December 2004, and extracted 44,783 queries likely to be questions: we simply selected queries that contained at least one question word (what, how, where, etc.). For our retrieval experiments, we randomly selected 100 user questions from this sample." [27]

Some examples queries:

[5] `www.metacrawler.com`

- what is the role of calcium in the contraction of skeletal muscle?
- how to beat traffic radar?
- what language do philipinos use?
- how to make cheerleading pompoms?

They also provided a set of relevance judgments for this query set. However, the judgment was far from complete and each query had only 1 relevant FAQ on average. We randomly sampled 50 queries from the query set and found relevant FAQs using the pooling method. We used the same criteria applied to the Wondir collection. We found 262 relevance FAQs from the WebFAQ collection.

3.3 Naver Collection

Naver[6] is a leading portal site in South Korea and its community-based question answering service is very popular. Over time, the service has accumulated more than 60 million question and answer pairs written in Korean. Topics are very broad from restaurant recommendations to Superstring theory. Q&A pairs in this service contain many spelling errors but the quality of answers is better than the other collections (Wondir and WebFAQ). Table 3.2 compares basic statistics of three Q&A collections.

We got two different Q&A collections from Naver: collections A and B. Naver collection A contains category information. We used this collection to test category-specific translations. The collection B contains non-textual information such as click counts. Therefore, we use this collection to build our answer quality prediction technique because the predictor exploits non-textual information. The following is a brief summary of two collections.

[6]http://www.naver.com

Table 3.2. Collection Statistics

Collection	**Naver A**	**Wondir**	**WebFAQ**
Provider	naver.com	wondir.com	U of Amsterdam
Q&A Source	Community-based QA service	Community-based QA service	FAQs from the web web crawler
Language	Korean	English	English, Dutch
#(Q&A Pairs)	8 million	1 million	3 million
#(Uniq Terms)	9,354,612	176,078	1,978,238
Length (word)	Question Title(6) Question Body(53) Answer(187)	Question(27) Answer(28)	Question(9) Answer(101)
Etc	decent quality answers	short answers	detection errors

Naver Q&A Collection A:

8 million Q&A pairs. Category information is available.

Received May, 2005.

Naver Q&A Collection B:

6.8 million Q&A pairs. Non-textual information is available.

Received February, 2005.

Each question in the service has a title and an optional body that describes the question title in more detail. We merged the question title and the body to make a question. If there are multiple answers for a question, all the answers are merged. The following is an example Q&A pair in the collection (translated from Korean).

Question Title: Do you have to take prenatal vitamins when trying to conceive?

Question Body: Is prenatal vitamins helpful in preparing your body for conception? Or is it unnecessary?

Answer: Prenatal vitamins are wonderful for taking before you get pregnant, and you should take a folic acid supplement. That helps prevent spinal cord malformations in a growing baby, and ...

3.3.1 Naver Test Collection A

Naver test collection A was constructed from the collection A. We randomly sampled 100 Q&A pairs from the held-out portion of the Naver collection. These Q&A pairs in the held-out portion were submitted to the service after we acquired the collection A. Each pair was automatically converted into a topic. The question title was used as a query and the question body was used as a narrative or a description of the query. After removing vague or private queries such as asking homework solutions, we got 50 queries.

We ran these queries through multiple search engines[7] to build a pool of candidate Q&A pairs for every query. We pooled the top 20 Q&A pairs from each ranked list returned from each engine and did manual relevance judgments. We followed the same guidelines used to build test collections for the Wondir and the WebFAQ collections. The correctness of the answer was ignored. As long as the question was semantically identical or very similar to the query, we judged the Q&A pair as relevant. Our annotators sometimes looked up the narrative part of the query to clarify the exact meaning of the query. We could find more number of relevant Q&A pairs in this collection for the same number of queries. We found 815 relevant Q&A pairs.

3.3.2 Naver Test Collection B

Naver test collection B is quite different from other test collections because this collection was designed to evaluate system performance with short keyword queries. We randomly sampled 125 queries from the search log of the portal site run by Naver. All queries were submitted in the same day. Because most users of the portal service issued short keyword queries, the average query length is 2.1 words.

---

[7]Query likelihood language models, Relevance Models, Okapi BM25 and Our model. Multiple retrieval results were made from a single method by changing parameter values and the searching field.

To build the judgment pool we ran the queries through multiple search engines and the top 20 Q&A pairs from each search engine were gathered into the pool. Because of the short query, it was hard to guess the intent of the query. Therefore, we had to employ different strategies to judge the relevance of Q&A pairs. The following describe the criteria that we used.

- **Relevant**
  - The question is semantically related to the query and the question contains all the query terms.
  - The question is semantically related to the query and the Q&A pair was clicked on multiple times for the query.

Our annotators often had to look up click through data to check whether the Q&A pair was clicked for a given query. In all, we found 1,700 relevant Q&A pairs. Very detailed records of the judgment process for this collection can be found in [63].

After judging relevance, our annotators read all the answers of relevant Q&A pairs and manually judged the quality of answers in three levels: good, medium or bad. They also manually checked whether the answer part is relevant to the question part. We found 98% of answers are relevant to the questions. This confirms our assumption that if the question part is relevant to the query then the answer part is almost always relevant to the query too. This assumption is later used to build test collections for answer passage retrieval. The results of the quality judgment are used in chapter 6 to evaluate our answer quality predictor. A detailed description about the quality judgment is in section 6.2.

CHAPTER 4

TRANSLATION-BASED Q&A RETRIEVAL FRAMEWORK

In this chapter, we describe our retrieval model that is designed to address the word mismatch problem mentioned in section 2.3. We borrow machine translation techniques to solve this problem. The idea of using machine translation techniques for information retrieval is attractive since the word mismatch problem can be explicitly addressed. Although translation and retrieval are conceptually related, they are different tasks and we need to understand and properly handle these differences to build effective translation-based retrieval systems. We recognize similarities and differences between statistical translation models and query likelihood language models and show how we can take advantage of both approaches.

4.1 Introduction

Most traditional information retrieval algorithms use simple term and document statistics to rank documents and fail to return relevant information if there are no matching terms between the query and the document. This so-called word mismatch problem has been one of the main factors impacting retrieval performance. The problem becomes more serious in Q&A retrieval because questions are much shorter than usual documents. Short documents (questions) have little supplemental text for the main content and there is less chance of describing the same concept using different wording.

To solve the word mismatch problem, many different approaches have been proposed. In this thesis, we focus on translation-based approaches because the rela-

tionships between words can be explicitly modeled. However, direct application of existing machine translation methods can causes problems. In machine translation, slow processing speeds can be tolerated, but information retrieval systems have to process huge amounts of data in a short time to interact with users. Therefore, we cannot use complex and expensive translation techniques.

Berger and Lafferty [6] proposed using the classic IBM translation model 1 for information retrieval tasks. The IBM model is attractive since no language specific knowledge is required and fast implementation is possible in the form of query expansion after learning word-to-word translation relationships. However, because of various fundamental differences between machine translation and information retrieval, the pure IBM model performs worse than other state of the art retrieval algorithms.

We explain the reasons for the poor performance of the pure IBM model as comparison to the query likelihood language model. This comparison also gives us insights that enable us to address problems with the IBM model. We propose a mixture model that leverages the benefit of both approaches.

Most previous studies on translation-based information retrieval did not recognize the weakness of the original translation model and adopted the IBM model "as is" or used toy data sets and weak baselines. We believe this work is the first successful application of the translation-based approach on large-scale real world retrieval problems.

Another important problem in using statistical machine translation is the difficulty of getting enough training samples. In this thesis, we propose to use collections of question and answer pairs as training data. Chapter 5 explains our solution for this problem in greater detail.

4.2 IBM Statistical Machine Translation Models

Statistical machine translation assumes a stochastic process that can generate translations of a source text. The parameters of the process (model) are automatically learned from bilingual corpora. This idea was initially introduced by Warren Weaver [71] in late 1940s and resurrected in early 1990s by researchers at IBM. Brown et al. [11] introduced a set of statistical machine translation models, namely IBM machine translation models, inspired by statistical speech recognition [2].

4.2.1 From Model 1 to Model 5

IBM models does not require any linguistic knowledge of the source or the target language and exploits only co-occurrence statistics of terms (or phrases) in training data. Depending on alignment strategies, they proposed 5 different models: from model 1 to model 5.

Model 1 treats every possible word alignment equally. Therefore, the word order does not matter. **Model 2** assumes only positions of terms are related to the word alignment. All term pairs positioned exactly at the same places in the source and the target respectively have the same alignment probability. For example, if the second term in the source tends to be connected to the first term in the target, the model tries to generate the first target term using the second source term in the generative process. In model 1, term to term translation probabilities are the only hidden parameters, but in model 2, alignment probabilities given the source and the target positions must be estimated.

Model 3,4 and 5 assume that a single term can be connected to multiple terms (fertility). Therefore a single term can generate multiple terms. These models have a significantly more complex structure than the model 1 and 2. In **model 3**, the first term and the second term generated from the same term are independent, so this model is called as a zero-order alignment model. The **model 4** is a first order

alignment model and every word is dependent only on the previous aligned word. Both models ignore whether a source word has been chosen or not and some portion of probability mass is assigned to the positions outside the source string boundary. Because of these problems, the probabilities for all valid assignments do not add up to one. **Model 5** fixes this problem by reformulating model 4 but it has to introduce significantly more number of parameters to fix this problem. Higher models use the outputs of lower models to estimated additional parameters.

4.2.2 Advantages of Model 1 in Information Retrieval

In this dissertation, we mix IBM model 1 with the query likelihood language model and introduce a new translation-based retrieval model. There are a few reasons for using the model 1 instead of more advanced models.

First, an efficient implementation of IBM model 1 is possible using a form of query expansion. In section 4.4, we provide implementation details. Higher level IBM models are significantly more expensive compared to model 1 and it is almost impossible to translate millions of documents in a few seconds. Since typical information retrieval systems have to return documents in a couple of seconds, such slow processing times are unacceptable.

Second, the performance gain of using higher level translation models is small. Only the model 1 can learn optimal parameters and all other models can get stuck on local maxima. Higher models have considerably larger number of parameters compared to model 1 and accurate estimations are very hard even with a large number of training samples. Because of this difficulty of training, model 1 often gives better performance over more advanced models [50].

Third, IBM model 1 can be easily integrated into the query likelihood model because of its simple structure. It is hard to integrate higher level IBM models into

other statistical frameworks because of stronger assumptions and tricky estimation methods.

4.2.3 IBM Model 1 - Equations

Following Brown et al., the probability that a query Q of length m is the translation of a document D (of length n) is given as,

$$P(Q|D) = \frac{P(m|D)}{(n+1)^m} \sum_{a_1=0}^{n} \sum_{a_2=0}^{n} \cdots \sum_{a_m=0}^{n} \prod_{i=1}^{m} P(q_i|d_{a_j}) \tag{4.1}$$

where, $P(m|D)$ is the probability that the length of the translation of D is m, q_i is the i^{th} term in Q, d_i is the i^{th} term in D, $P(q_i|d_{a_j})$ is the term translation probability and and $a_i = j$ denotes that i^{th} term in Q is connected (aligned) to the j^{th} term in D. Since the IBM model assumes that each source string has a special *null* term at position 0, each target term starts its iteration from position 0 in the above equation.

IBM model 1 further assumes that $P(m|D)$ is a constant and every alignment is equally likely. Because of these simplifications, after algebraic manipulations, equation 4.1 can be rewritten as,

$$P(Q|D) = \prod_{w \in Q} P(w|D) \tag{4.2}$$

$$P(w|D) = \frac{|D|}{|D|+1} P_{tr}(w|D) + \frac{1}{|D|+1} P(w|null) \tag{4.3}$$

$$P_{tr}(w|D) = \sum_{t \in D} P(w|t) P_{ml}(t|D) \tag{4.4}$$

where, $|D|$ is the length of D and $P(w|null)$ is the probability that the term w is translated (generated) from the *null* term.

4.3 TRANSLATION-BASED LANGUAGE MODELS

Before presenting our retrieval model, we need to explain the language modeling approach to information retrieval [55] because our retrieval framework is based on this approach.

4.3.1 Language Modeling Approach to IR

The language modeling approach to information retrieval has been successfully applied to many different applications because of its flexibility and theoretically solid background. A language model is a mechanism for generating text. In information retrieval, probabilities of sampling the query from the document (document language model) are used to rank documents. The unigram language model is commonly used as a document language model. The unigram language model assumes that each term is generated independently. It concerns only the probabilities of sampling a single word. The sampling probabilities are estimated by the maximum likelihood estimator. In the maximum likelihood estimator, unseen words in a document have zero probability and to remedy this problem, the idea of smoothing is introduced. The smoothing process transfers some probability mass from the seen words to the unseen words. Dirichlet smoothing [76] is popular because of its good performance and cheap computational cost. The ranking function for the query likelihood language model with Dirichlet smoothing can be written as

$$Sim(D,Q) = P(D|Q) = P(D)P(Q|D)/P(Q) \approx P(D)P(Q|D) \tag{4.5}$$

$$P(Q|D) = \prod_{w \in Q} P(w|D) \tag{4.6}$$

$$P(w|D) = \frac{|D|}{|D|+\lambda} P_{ml}(w|D) + \frac{\lambda}{|D|+\lambda} P_{ml}(w|C) \tag{4.7}$$

$$P_{ml}(w|D) = \frac{\#(w,D)}{|D|} \ , \ P_{ml}(w|C) = \frac{\#(w,C)}{|C|} \tag{4.8}$$

where Q is the query, D is the document, C is the background collection, λ is the smoothing parameter, $|D|$ and $|C|$ are the lengths of D and C, respectively. $\#(t, D)$ denotes the frequency of term t in D.

In equation 4.5, P(Q) is ignored because it has no effect on the ranking of documents. A constant has been used for P(D) in many information retrieval tasks but P(D) can be used to integrate query independent document features like quality or authority. Chapter 6 discusses the estimation of P(D) to incorporate the quality score into the retrieval process. In this chapter, we explain the calculation of $P(Q|D)$: relevance score of document D for a given query Q or semantic similarity between D and Q.

4.3.2 IBM Model 1 vs. Query Likelihood

It is easy to see that the equations used to describe the query likelihood language model and the IBM model look similar to each other. There are three different comparable components in the two models.

- $P_{ml}(w|C)$ vs. $P(w|null)$
- λ vs. 1
- $P_{ml}(w|D)$ vs. $P_{tr}(w|D)$

Let us discuss these differences one by one.

1. $P_{ml}(w|C)$ vs. $P(w|null)$

   $P(w|null)$ is introduced in the IBM model to generate spurious terms in the target sentence. $P_{ml}(w|C)$ has a very similar role in the language models. This background distribution generates common terms that connect content words. Therefore, they play the same role in both approaches. However, the concept

of the spurious term is a little awkward and the estimated values are less stable compared to the background distribution used in the language modeling framework. So we choose $P_{ml}(w|C)$ instead of $P(w|null)$ for our model.

2. λ vs. 1

   The translation model assumes only one null word and it is not easy to control the effect of background smoothing. On the other hand, the language modeling approach explicitly uses the parameter λ to adjust the amount of smoothing. Smoothing parameters have been shown to have a significant impact on retrieval performance. The lack of a mechanism to control background smoothing in the IBM model leads to the relatively poor performance. Therefore, we decided to use λ in our model.

3. $P_{ml}(w|D)$ vs. $P_{tr}(w|D)$

   The third difference comes from different sampling strategies. The query likelihood model uses the maximum likelihood estimator to calculate the probability of sampling words from the document. This method gives zero probabilities for unseen words in the document. The word mismatch problem occurs because of this naive sampling method. The IBM model uses a more sophisticated sampling method. Every word in the document has some probability of being translated into a target word and these probabilities are added up to calculate the sampling probability. Therefore, if a document has many semantically related terms to a target term, then the term gets high probability from the document. This sampling approach helps to overcome the word mismatch problem by considering word-to-word relationships. However, the sampling method used in the IBM model has a problem known as the self translation problem.

4.3.3 Self-Translation Problem

Since the target and the source languages are the same, every word has some probability to translate into itself. This self-translation probability cannot be 1 because the source word must have some probability of being translated into other words. Sometimes, low self-translation probabilities deteriorate retrieval performance by giving very low weights to the matching terms. In the opposite case, very high self-translation probabilities do not exploit the merits of the translation approach.

To overcome this problem, a few different approaches have been proposed. Murdock and Croft [48] use the translation-based sampling method only when the target term does not exist in the document and use the maximum likelihood estimator if the target term is presented in the document. This method does not fully exploit the power of the translation method. Jeon et al. [25] set $P(w|w) = 1$ for all w while maintaining other word translation probabilities unchanged. This approach produces inconsistent probability estimates and makes the model unstable. Jin et al. [28] force other terms to have lower translation probabilities than self translations: $P(w|w) \geq P(w' \neq w|w)$. This constraint can reduce the problem but very low or very high self-translations are still possible. All these heuristic modifications gave significant improvements over the original translation model. Instead of using makeshift solutions, here, we propose linear mixture of two different estimations: maximum likelihood estimation and translation-based estimation.

4.3.4 TransLM

Our final ranking function looks like

$$Sim(D,Q) = P(D|Q) = P(D)P(Q|D)/P(Q) \approx P(D)P(Q|D) \tag{4.9}$$

$$P(Q|D) = \prod_{w \in Q} P(w|D) \tag{4.10}$$

$$P(w|D) = \frac{|D|}{|D|+\lambda} P_{mx}(w|D) + \frac{\lambda}{|D|+\lambda} P_{ml}(w|C) \tag{4.11}$$

$$P_{mx}(w|D) = (1-\beta) P_{ml}(w|D) + \beta \sum_{t \in D} P(w|t) P_{ml}(t|D) \tag{4.12}$$

In our translation-based language model (TransLM), we can control the impact of the translation component by β. If we set a small value for β, the model behaves like the query likelihood model and the importance of matching terms is emphasized. This is similar to increasing the self translation probability. Thus, we can control the amount of self translation using β. The amount of background smoothing is adjusted using λ. Experimental results show our approach reduces the word mismatch problem and outperforms other modified translation models.

4.4 Efficiency Issues and Implementation of TransLM

A naive implementation of TransLM may rely on document-at-a-time methods [10, 64] that visit every document to calculate the similarity scores between documents and the query. For a document of length m and a query of length k, the system have to look up the word translation table mk times. The naver collection A has about 2 billion terms. This implies the system has to look up the table more than billions of times even for a very short query. Obviously this is unacceptable. In this section, we convert the document-at-a-time algorithm into a term-at-a-time algorithm.

4.4.1 Flipped Translation Tables

The main idea is to limit the number of lookups. We want to use only source terms that have high translation probability to query terms (target terms). To efficiently find these terms, we have to construct a flipped word translation table. A normal word translation table consists of tuples in the form of <source word , target word , P(target word | source word)> and the first element (source word) in the tuple is the primary key. The flipped table save tuples in the form of <target word , source word , P(target word | source word)> and the target word is the primary key.

To efficiently access translation tables the tuples are saved using a data structure depicted in Figure 4.1. First, all tuples are sorted by the primary key in ascending order. Tuples sharing the same primary key construct a block and tuples in the same block is further sorted by the descending order of the translation probability. Sorted blocks are saved in consecutive disk space. A B-tree is built to access blocks with the primary key. Each leaf node in the tree has a pointer to the corresponding block on the hard-disk. In most cases, the whole B-tree can be loaded into memory. A block is loaded into memory only when it is required.

For a given query term, our system looks up the flipped translation table and loads a block that has the term as a key. Since tuples in the block are sorted in the

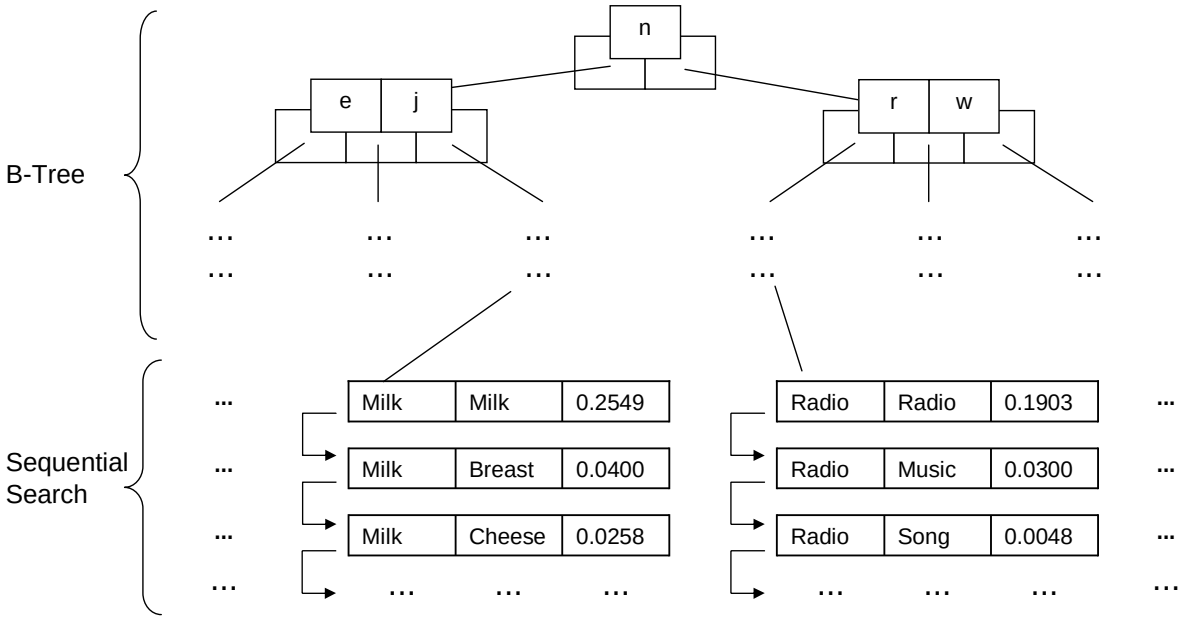


Figure 4.1. Data structure for translation tables.

descending order of the translation probability. The first tuple contains the word that has the highest translation probability to this query term. We read tuples sequentially until we reach the point that the translation probability is smaller than a threshold value. Using this date structure, we can efficiently find a list of terms that have high translation probability to the query terms. A single look up could be done in a few milliseconds in our experiments.

4.4.2 Term-at-a-time Algorithm

Table 4.1 is the pseudocode of the fast version of TransLM. For each query term, we look up the translation table and retrieve a list of source words that have high translation probability to the query term. Then, for each source term, we extract a list of documents that contains the source term from the inverted index. The probability of sampling the query term from each document in the list is calculated and this probability is added up to produce the final score for each document.

```
for every query term q in Q
    // look up translation table, T: term list, P: probability list
    < T, P > = lookup(q,threshold)
    // add term q into the term list and the probability list
    < T, P > = < T, P > + (q, 1.0)
    for each term t in T
        // get document list D from inverted index
        D = getDocList(t)
        for every document d in D
            // Maximum likelihood estimation
            P_ml = frequency(t,d) / docLength(d)
            // translation-based estimation
            if t == q  // matching term
                termScoreBuf[d] += (1-beta)*P_ml
            else       // expanded terms
                termScoreBuf[d] += beta*P_ml*P(q|t)
            end
        end
    for  every document d in the collection
        // back ground smoothing
        termScoreBuf[d] = termScoreBuf[d]*(1-lambda) + lambda*collectionLM[q]
        // add term score into the final results
        results[d] += log(termScoreBuf[d])
    end
end
```

Table 4.1. Fast implementation of TransLM.

This algorithm uses only subset of documents in the collection that have high chance of being relevant. In our experiments, we choose source terms whose translation probability is over 0.01. We tested a few different threshold values and found 0.01 to be good enough. We can expand the query with more source terms by lowering the threshold value but the performance gain is minimal once the threshold value reaches 0.01. In all experiments in this thesis the threshold value is 0.01. Even in worst cases (very long queries), we could search 8 million questions in less than 2 seconds using a single PC (Pentium4 2.0GHz, 4GB memory).

4.5 Comparison with Relevance Models

Our experimental results presented in chapter 7 show relevance models works poorly for our task. In this section, we briefly describe the weekness of relevance models and show why our model can avoid these problems.

- **Relevance models depend on the initial search result.**

  Relevance models use the query likelihood language model for initial search and leverage highly ranked documents to build relevance models. If initial search fails and highly ranked documents are not relevant to the query, then a relevance model built using these non relevant documents is actually worse than the original query model. The quality of initial retrieval using the query likelihood model is sometimes poor due to the word mismatch problem and is worse if the query uses synonyms or different wording. Therefore, the relevance model has been shown to perform poorly for the task of searching short text snippets where the word mismatch problem is serious [47, 48]. However, our approach does not rely on the performance of other search engines and uses word relationships learned from question and answer pairs. For example, we could retrieve questions about 'moon' using a query 'mooon' because our model

uses the misspelled word 'mooon's relationship to the word 'moon'. This kind of correction is not impossible in the relevance model.

- **Relevance models treat every term in a document equally.**

  Relevance models weight documents differently depending on similarity between the document and the query. However, all terms in a document are treated equally and get the same weight when they are added to the relevance model. For example, for the query 'what is the population of bahamas?', the query likelihood model returned 'what is the nickname and flower for the bahamas?' at the first rank from the Wondir collection. The relevance model blindly adds all terms in the question into the query, therefore the updated relevance model contains some non-relevant terms like 'nickname' and 'flower'. However, our model always expands terms that have high semantic relationship to the query.

- **Relevance models are less stable than TransLM.**

  Depending on the quality of initial search, sometimes, relevance models produce much worse results compared to the query likelihood model. However, our model consistently shows similar or better results compared to the query likelihood language model across different queries and collections. One reason is that our model contains the query likelihood model as a part of the model and the translation component usually play the role of tie-break when multiple documents have similar number of matching terms. TransLM does not radically change the original ranking of the query likelihood model. TransLM maintains the advantage of the query likelihood model and carefully adds the translation component to it. Such adjustment is possible because both components are based on statistical estimations and the mixing parameter in the model carefully combines both components.

CHAPTER 5

LEARNING WORD-TO-WORD TRANSLATIONS

It is obvious that success of our retrieval model depends on the quality of word-to-word translations. $P(w|t)$ in equation 4.12 can be interpreted in different ways. In the translation point of view, $P(w|t)$ is the probability that term w is the translation of term t. In the language modeling point of view, $P(w|t)$ is the probability that term w is in the query generated from documents that contain term t. In either case, $P(w|t)$ denotes semantic similarity between two terms. In this chapter, we describe how we can calculate these word-to-word translation relationships (word relationships[1]) from collections of question and answer pairs.

5.1 Properties of Word Relationships

Before discussing the estimation (learning) method, we present a few important properties of the word relationships.

1. Firstly, they are not symmetric. This means $P(w|t)$ is not always equal to $P(t|w)$. Actually, $P(w|t)$ is different from $P(t|w)$ in most cases. For example, $P(animal|tiger)$ may be bigger than $P(tiger|animal)$ because 'animal' has broader meaning and 'tiger' is only one kind of object that it can encompass. Therefore, 'tiger' is not that meaningful to 'animal' but 'animal' is an important property of 'tiger'.

[1]In this thesis, 'word relationship' is the abbreviation of 'word to word translation relationship'.

2. Secondly, the relationships between words are not fixed. For example, different groups of people use words differently. In a shopping community, $P(rolex|watch)$ may have high value. However, in computer geeks community, $P(rolex|watch)$ may have very low value, instead $P(tray|watch)$ may get much higher value since Windows systems have a 'tray' at the right bottom of the screen that contains a watch to show current time.

3. The relationships also change depending on retrieval or translation tasks. For the task of translating answer to question (or retrieving answers for a given question), $P(29035|everest)$ is small, but in the opposite task of translating question to answer, $P(29035|everest)$ can be much bigger because many people ask about the height of Mt Everest and most answers contain 29035. Therefore the chance of observing 29035 in the answer given that the question contains "everest" is high. However, few people include 29035 in their question (if they know the exact height, there is no reason to ask), so the chance of seeing this number is very low even if corresponding answer contains "everest".

4. Since, we use statistical approaches in the retrieval process, the word relationships must be given as probability values. This means $\sum_{w \in V} P(w|t) = 1$. In other words, the summation of the word translation probabilities over all words in the vocabulary for any given source word must be one.

5.2 Training Sample Generation

In the previous section, we showed that the word translation probabilities vary depending on context. This implies that a good way of learning word relationship for a community-based question answering service is to use questions and answers submitted to the service because they reflect interests and word usages of the community better than other general resources that we can get outside like WordNet[19].

To learn the word translation probabilities, we need training data. Since our task is calculating semantic distances between two questions, the best training data might be a large number of semantically equivalent question pairs. They do not have to be lexically similar. Actually lexically different and semantically similar pairs are better training data since we want to bridge the lexical chasm between questions using the training samples.

However, such data is not readily available. Instead, we have a large number of question and answer pairs. The answer is not semantically equal to the question but it is at least semantically related and topically relevant to the question. Therefore, our first approach is using Q&A pairs instead of question pairs to learn the word translation probabilities.

The second approach is to automatically collect semantically similar question pairs from existing question and answer archives by comparing answers. These pairs serve as training data for our translation-based retrieval model. Following subsections describe this method in greater detail.

5.2.1 Key Idea

Many people do not carefully check whether the same question has been asked before and post their questions on Q&A boards. Therefore, many semantically identical questions can be found in question and answer archives. Our assumption is if

two answers are very similar than the corresponding questions should be semantically similar, even though the two questions are lexically very different.

5.2.2 Similarity Measures

To find similar answer pairs, reliable similarity measures between answers are required. One thing to be careful is that any similarity measure seriously affected by length is not appropriate because the lengths of answers vary significantly. Answers can be very short, especially, for factoid questions. Some answers are very long because some people generate answers by copying multiple web pages. We test three different similarity measures.

1. **Cosine similarity with TF.IDF weights**. This measure has been extensively used for various IR and NLP tasks. An advantage of using the cosine similarity is that the measure is symmetric.

2. **Query likelihood scores between two answers**. We convert every answer into a query and retrieve other answers using the query likelihood language model. Every pair has two different scores depending on which answer becomes a query. We just pick the maximum value of the two scores. We call this measure **LM-SCORE**.

3. The third measure uses ranks instead of scores to resolve the problem of non-symmetric scores in the second method. If answer A retrieves answer B at rank r_1 and answer B retrieves answer A at rank r_2, then the similarity between two answers is defined as the reverse of the harmonic mean of r_1 and r_2. $sim(A, B) = \frac{1}{2}(\frac{1}{r_1} + \frac{1}{r_2})$. We use the query likelihood language model to rank answers. We call this measure **LM-HRANK**.

Rank	Cosine	LM-SCORE	LM-HRANK
10	0.00	0.90	0.80
100	0.21	0.67	0.64
1000	0.27	0.41	0.48

Table 5.1. The ratio of correct answer pairs in top 10, 100 and 1000 positions for each similarity measure.

5.2.3 Experiments

To compare the three measures, we did simple experiments. We sampled 5,200 question-answer pairs from the Naver collection A. All the questions are from the 'Email' category. The average length of questions is 5.9 words and the average length of answers is 150.1. To calculate the cosine similarity and the query likelihood language models, we used the LEMUR[2] toolkit.

In total, 1,351,700 pairs of answers are possible from 5,200 answers. All of these pairs are ranked according to the three different similarity measures. We manually evaluated the top 1000 pairs for each method. If a question pair connected to an answer pair is semantically identical or very similar, we judge the answer pair to be a a correct match. Table 5.1 shows the ratio of the correct matches in the top 10, 100, and 1000 pairs for each similarity measures.

The cosine similarity works poorly because the measure favors short answers. For example, in our dataset, an answer has only two words ("Korean homework") and answer pairs containing this short answer usually have very high cosine similarity scores. Therefore, the cosine similarity cannot be a good similarity measure for answers.

Language modeling based measures show good performance. Using LM-SCORE, 90% of the answer pairs in the top 10 connect semantically equivalent questions. In the top 100, 67% of the answer pairs are correct matches. LM-HRANK shows better results than LM-SCORE in the top 1000 pairs.

[2]http://www-2.cs.cmu.edu/lemur/

While LM-SCORE and LM-HRANK show comparable performance, they retrieve different sets of answer pairs. The number of overlapping answer pairs between the top 100 pairs in LM-SCORE and the top 100 pairs in LM-HRANK is only 6. This implies that more correct answer pairs can be retrieved when both measures are used together.

5.2.4 Examples

For every Q&A collections introduced in chapter 3, we automatically built a training collection using the LM-HRANK measure. We selected all answer pairs whose LM-HRANK score is over 0.1. Table 5.2 shows examples of the question pairs found using our method. Each question pair in the examples contains semantically similar questions but they share very few common terms.

5.3 Algorithm

In the previous section, we discussed how we prepared our training samples. This section describes how we calculate the word-to-word translation probabilities from the training data.

5.3.1 Word Relationship Types

Whether we use question and answer pairs or question and question pairs, a training sample consists of two parts. We have to designate one part as source and the other part as target to learn the word relationships. As already shown in section 5.1, depending on the source and the target designation, we get different word-to-word translation probabilities.

To denote the source and target designation, we extend the original notation for word relationships. New notation explicitly specifies the source and the target domains. For example, $P(w, Q|t, A)$ denotes that this specific word relationship between term w and term t is obtained by assigning the answer part (A) as source and the

Naver 'Email' Collection

Can I attach a 5 mega byte file in my email? Sending big movie files to my friends over the net by email
Why do we have to use only English for email addresses? Why can't I use Korean in email IDs?
What is the best email service? Who provides the most popular and powerful email accounts?
Who invented email? The first person who used email
What cause corrosion What is the reaction of copper and oxygen

(Translated from Korean)

Wondir Collection

i want to know about tsunami can you tell me in which website i can find about tsunami
what is the www address for the white house do you know white house url
what meal can you make with pork chunck i m look for a different kind of main dish that main ingredient is pork
what was the first progamme on channel 4 the first ever show screen on channel 4
who the first one who fly to sky who was the first one who fly with plane

Table 5.2. Examples of question pairs found from the Naver collection and the Wondir collection using the LM-HRANK measure.

question part (Q) as target in the learning process. We generated four different types of word relationships.

1. $P(Q|A)$

   Source: Answer, Target: Question

   This type of word relationships is intended to be used in the task of translating answers into questions (or generating questions from answers).

2. $P(A|Q)$

   Source: Question, Target: Answer

   This type of word relationships is intended to be used in the task of translating questions into answers (or generating answers from questions).

3. $P(Q|Q)$

   This type of word relationships is intended to be used in the task of translating a question into another form of question[3]. This is the task that we are most interested in because we want to find similar questions for a given question. We calculate this word relationship by convoluting $P(Q|A)$ and $P(A|Q)$

   $P(w, Q|t, Q) = \sum_{s \in V} P(w, Q|s, A)P(s, A|t, Q)$

4. $P(Q \leftrightarrow Q)$

   This type of word relationships is intended to be used in the task of question retrieval for a given question. The difference between $P(Q \leftrightarrow Q)$ and $P(Q|Q)$ is that $P(Q \leftrightarrow Q)$ is calculated from the question pairs that we automatically find using the method that we described in the previous section.

Our experimental results in chapter 7 shows each type of word relationships has a different effect A detailed explanation can be found in chapter 7.

[3]This task can be thought of as transformation of the original question into another form.

5.3.2 EM Algorithm

A simple method of calculating word-to-word translation probabilities is by using co-occurrence statistics. For example, if a target term co-occurs many times with a given source term, then the target term must have a high translation probability from the source term. However, naive application of this idea always assign big probability mass to stops words because they co-occur many times with any source term. Therefore, we need more advance methods that can rule out meaningless co-occurrences.

Our approach is finding word relationships that maximize the likelihood of sampling the target text from the source text in the training samples. The likelihood function is given by,

$$L = log[\prod_{(T,S)\in J} \prod_{w\in T} \sum_{t\in S} P(w|t)P_{ml}(t|S)] \tag{5.1}$$

where, J denotes a set of training samples and (T, S) is the target string and the source string in a training sample. This likelihood function is a simplified version of the likelihood function that IBM model 1 uses to find word-to-word translation probabilities given a bilingual corpus. We use basically the same EM algorithm used in IBM model 1 to find optimal parameter values that maximize above likelihood function. The only difference is that we do not assume there is a 'null' word in every source string. We do not need this assumption any more because we explicitly use background smoothing in our retrieval model. The translation probability from a source word t to a target word w is given as

$$P(w, |t) = \lambda_t^{-1} \sum_{i=1}^{N} c(w|t; J^i) \tag{5.2}$$

where, λ_t is a normalization factor to make the sum of the probabilities equal to unity. N is the number of training samples. J^i is the ith pair in the training data.

$$c(w|t;J^i) = \frac{P(w|t)}{P(w|t_1) + \ldots + P(w|t_n)} \#(w, J^i) \#(t, J^i) \tag{5.3}$$

where $\{t_1, \ldots, t_n\}$ are words in the answer in J^i and $\#(w, J^i)$ is the number of times that w occurs in J^i.

As can be seen from the equations, we need the old translation probabilities to estimate the new translation probabilities. We initialize the translation probabilities with random values and then estimate new translation probabilities. This procedure is repeated until the probabilities converge. Brown et al. [11] showed that the procedure always converges to the same final solution regardless of the initial values.

5.4 Word-to-Word Translation Examples

Table 5.3 shows example word relationships learned from the Wondir collection. In all cases, the best target term that has the highest translation probability from the source term is the source term itself and this result validates the learning process. It is easy to see that most target words in the table are semantically related to the source words.

The first and the second columns in the table show the effect of switching source and target. Both examples have the same source word "cheat", but the top 10 target terms are different. If the answers contain the word "cheat", corresponding questions tend to contain "boyfriend(4)" and "husband(5)". This implies many questions about cheating are about the cheating behavior of boyfriends or husbands. In the opposite case of sampling answers, if questions contain "cheat" then answers tend to have "trust(2)", "forgive(3)", "dump(6)" or "leave(8)". Obviously, these verbs represent actions that someone can take when their partners cheat on them. Since "trust(2)" and "forgive(3)" have higher probabilities than "dump(6)" and "leave(8)", we can guess answerers in the Wondir community usually suggest forgiveness. These

example show that word relationships change depending on the source and the target designation.

Table 5.4 and 5.5 show another examples learned from the WebFAQ collection and the Naver collection. The examples for the Naver collection show word relationships learned from the automatically generated training samples described in section 5.2.

5.4.1 Category Specific Word Translation

Word relationships change depending on context. When we talk about IT business, 'apple' is a company name and 'computer' is a semantically close term. However, when we discuss agriculture, 'apple' is closer to 'fruit' than 'computer'. We can incorporate this context information into our system by calculating category specific word relationships.

Figure A.1 shows different lists of target terms for a given source term 'watch'. Each cell shows a Korean word and an English translation of the word. The left column shows the top 10 target terms learned from question and answer pairs in the 'shopping' category of the Naver collection. It is easy to see that most target terms are brand names of luxury watches. The right column shows the top 10 target words learned from the 'Computer Novice' category for the same source word. Many terms are related to how to change time using the watch icon in the tray of Windows systems. This table show the importance of using right word relationships that match the context of the question. Our experimental results show that we get better retrieval performance when we use category specific word relationships.

Example source term: 시계 (watch, clock)
Top 10 target terms

	`Shopping' Category		`Computer' Category	
1	시계	watch	시계	watch
2	차고	wear	msconfig	msconfig
3	브랜드	brand	건전지	battery
4	시	hour	hour	hour
5	스와치	swatch	트레이	tray
6	테크노	techno	유틸리티	utility
7	카시오	casio	아이콘	icon
8	알마니	almani	맞는지	correct
9	로렉스	rolex	변경	change
10	명품	luxury	clock	clock

Figure 5.1. Category Specific Word Relationships. Naver Collection.

Collection	**Wondir**			
Source Term	cheat	cheat	aspirin	theft
Target Term 1	cheat	cheat	aspirin	theft
Target Term 2	married	trust	pain	steal
Target Term 3	affair	forgive	asprin	charge
Target Term 4	boyfriend	cheater	tylenol	bank
Target Term 5	husband	relationship	headache	stolen
Target Term 6	another	dump	ring	game
Target Term 7	love	again	relieve	stole
Target Term 8	trust	leave	thin	felony
Target Term 9	relationship	deserve	arthritis	probation
Target Term 10	girlfriend	love	fever	item
Source Part	Answer	Question	Answer	Question
Target Part	Question	Answer	Question	Answer

Table 5.3. Word relationship examples. Wondir Collection. Each column shows top 10 target terms for a given source term. The last two rows show which parts are used for the source and the target in the training process.

Collection	**WebFAQ**			
Source Term	internet	pregnant	solar	tax
Target Term 1	internet	pregnant	solar	tax
Target Term 2	web	fetus	energy	irs
Target Term 3	online	baby	sun	revenue
Target Term 4	compute	physician	planet	income
Target Term 5	stolen	browse	miscarriage	power
Target Term 6	access	consult	system	taxpayer
Target Term 7	com	ovulation	panel	property
Target Term 8	site	conception	pv	taxation
Target Term 9	connect	prenatal	electric	state
Target Term 10	website	birth	battery	deduct
Source Part	Answer	Question	Answer	Question
Target Part	Question	Answer	Question	Answer

Table 5.4. Word relationship examples. WebFAQ collection. Each column shows top 10 target terms for a given source term. The last two rows show which parts are used for the source and the target in the training process.

Rank	bmp	format	music	intel	excel	font
1	bmp	format	music	pentium	excel	font
2	jpg	format*	file	4	korean	korean
3	gif	xp	tag	celeron	function	97
4	save	windows	sound	amd	novice	add
5	file	hard	background	intel	cell	download
6	picture	98	song	performance	disappear	control-panel
7	change	partition	play	support	convert	register
8	ms-paint	drive	mp3	question	if	install
9	convert	disk	cd	buy	xls	default
10	photo	C	source	cpu	record	photoshop

Table 5.5. Word relationship examples. Naver Collection. Learned from artificially generated training data. The first low shows source terms and each column shows top 10 terms that are most semantically similar to the source term. It is not hard to notice most of the words in the table have strong semantic relationships with the source words. (format and format* are different in Korean but both words are translated into 'format' in English) (Translated from Korean)

CHAPTER 6

ESTIMATING ANSWER QUALITY

Community-based question answering services are important sources of Q&A pairs. This chapter describes how we estimate the quality of answers collected from these services using non-textual features. The reasons for focusing on using non-textual features are two fold. First, they have strong correlation with the quality. Second, these features are abundant in many web services. Experimental results show that our approach reliably distinguishes good answers from bad and that even it can even boost retrieval performance. Our approach can be easily applied to other web services.

6.1 Introduction

New web services become available every day and these services accumulate new types of documents that have never before existed. Many service providers keep non-textual information related to their document collections such as click-through counts, or user recommendations. Depending on the service, the non-textual features of the documents may be numerous and diverse. For example, blog users often recommend or send interesting blogs to other people. Some blog services store this information for future use. Movie sites saves user reviews with symbolic representations rating the movie (such as **A** or $\star\star\star\star\star$).

This non-textual information has great potential for improving search quality. In the case of the homepage finding, link information has proved to be very helpful in estimating the authority or the quality of homepages [9, 33]. Usually textual

features are used to measure relevance of a document to a query and non-textual features can be utilized to estimate the quality of a document. While smart use of non-textual features is crucial in many web services, there has been little research to exploit them. In this chapter, we demonstrate how systematically process these non-textual features found in community-based question answering services to estimate the quality of answers submitted to the services.

Estimating answer quality is important in Q&A retrieval to improve user experience. Some users of community-based question answering services make fun of other users by answering nonsense. Sometimes irrelevant advertisements are given as answers. Figure 6.1 shows examples of bad quality Q&A pairs found from Yahoo Answers!. Returning these bad and irrelevant answers to users of a Q&A retrieval system should be avoided. The quality problem is also important when there are many duplicated questions, or many responses to a single question. Duplicated questions are generated because some users post their questions without carefully searching existing collections. These semantically duplicated questions have answers with varying quality levels and we need to rank these relevant answers by the order of their quality levels.

We use kernel density estimation [24] and the maximum entropy approach [7] to handle various types of non-textual features and build a stochastic process that can predict the quality of answers associated with the features. We do not use any service or collection specific heuristics, therefore our approach can be used in many other web services. The experimental results show the predictor has the ability to distinguish good answers from bad ones.

To evaluate the performance of the predictor with respect to retrieval, we performed simple retrieval experiments. We integrate the quality score into the query likelihood language model as a document prior. Experimental results show that we can effectively find relevant and high quality answers by combining the quality score

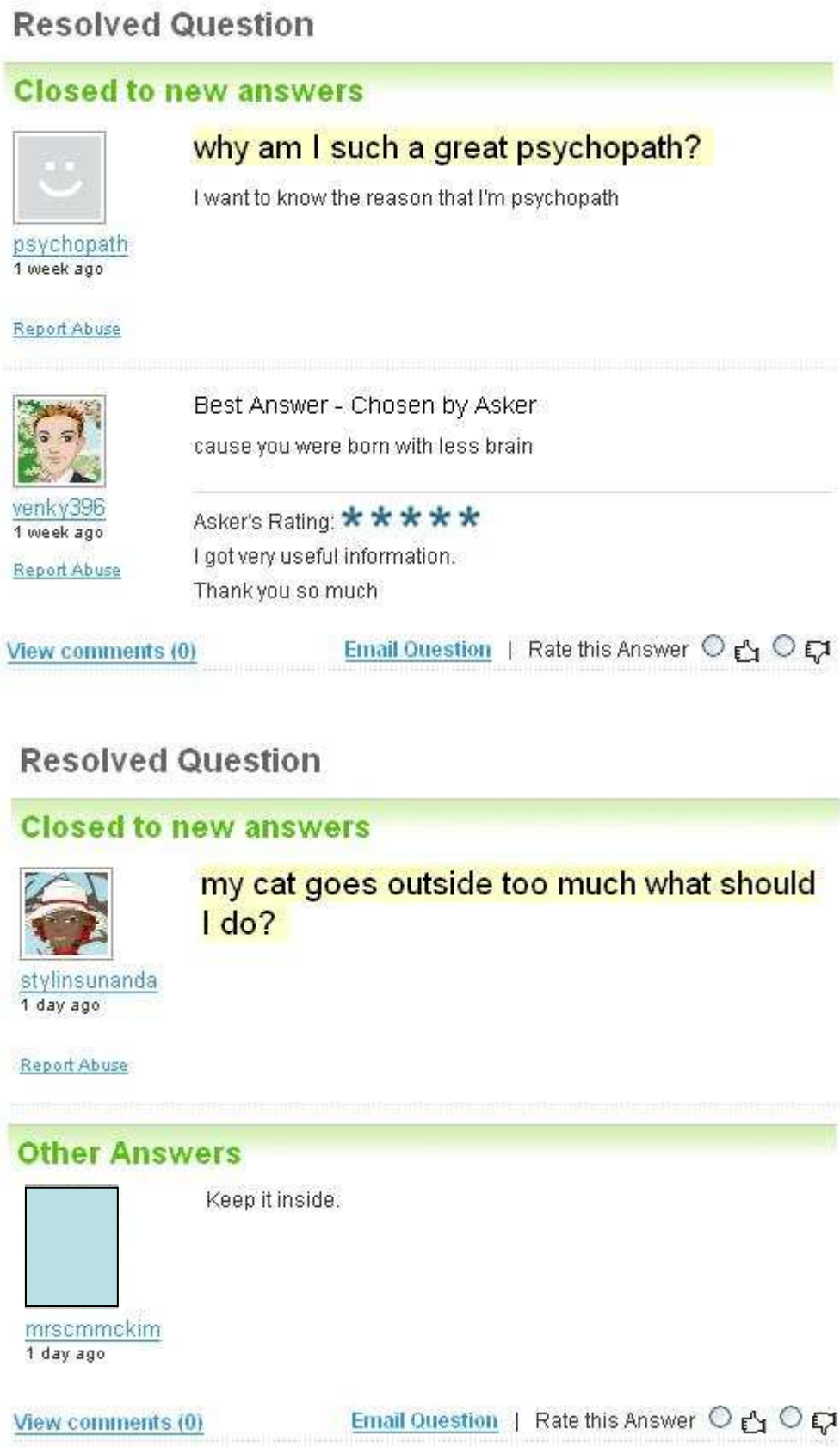


Figure 6.1. Examples of bad quality Q&A pairs found in Yahoo Answers!.

Quality of Answers, Test Samples

Bad	Medium	Good
208 (12.2%)	393 (23.1%)	1099 (64.7%)

Quality of Answers, Training Samples

Bad	Medium	Good
81 (9.1%)	212 (23.7%)	601 (67.2%)

Table 6.1. The relationships between questions and answers in Q&A pairs are manually judged. The test samples consist of 1700 Q&A pairs. The training samples have 894 Q&A pairs. Both training and test samples show similar statistics.

into the retrieval model. In chapter 7, we describe how to incorporate the quality measure into the translation-based retrieval model presented in chapter 4.

6.2 Training and Test Data

In general, good answers tend to be relevant, informative, objective, sincere and readable. We may separately measure these individual factors and combine scores to calculate overall the quality of the answer. However, this approach requires development of multiple estimators for each factor and the combination is not intuitive. Therefore, we use a holistic view to decide the quality of an answer. Our annotators read answers, consider all of the above factors and specify the quality of answers in just three levels: Bad, Medium and Good. This holistic approach shifts the burden of combining individual quality metrics to human annotators.

To build and evaluate our quality predictor, we use the Naver collection B since non-textual information is available for this collection. In section 3.3.2, we explained how we found 1700 relevant Q&A pairs to the 125 queries. For the 1,700 Q&A pairs, we manually judged the quality of answers. In this step, the query was ignored and only the relationships between questions and answers in Q&A pairs are considered.

The results of the quality judgment are in Table 6.1. Around one third of the answers have some sort of quality problems. Approximately one tenth of the answers are bad. Therefore, we need to properly handle these bad documents (Q&A pairs).

To build a machine learning based quality predictor, we need training samples. We randomly selected 894 new Q&A pairs from the Naver collection B and manually judged the quality of the answers in the same way. Table 6.1 shows that the test and the training samples have similar statistics.

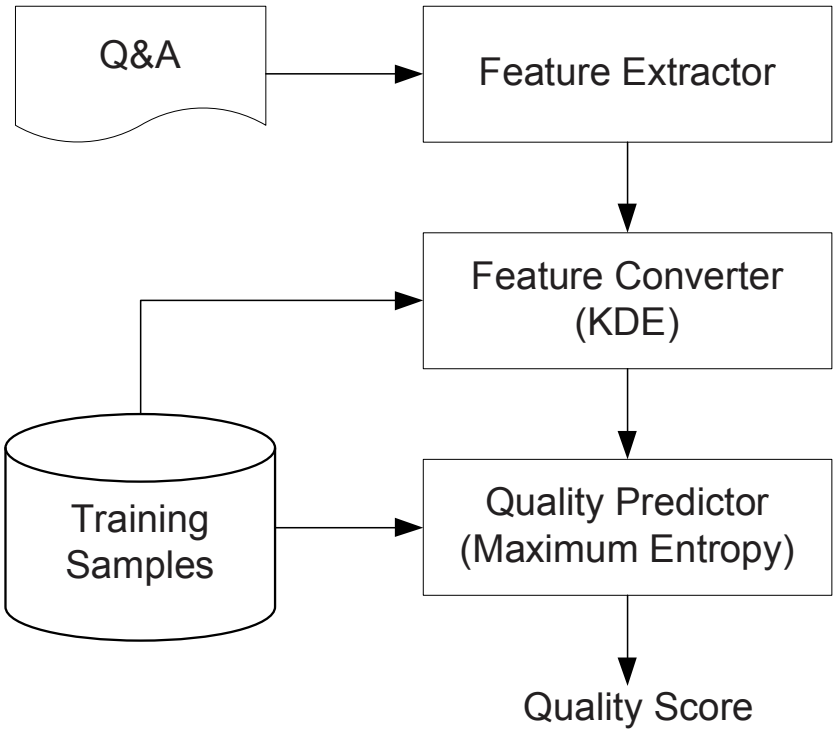


Figure 6.2. Architecture of the quality predictor.

6.3 Feature Extraction and Processing

Figure 6.2 shows the architecture of our quality prediction system. The input of the system is a Q&A pair and the output is the probability that the Q&A pair has a good answer. This section explains the feature extraction component and the feature conversion component and the next section describes our stochastic model based on maximum entropy approaches.

6.3.1 Non-Textual Features

First we need to extract feature vectors from a Q&A pair. We extract 13 non-textual features. Table 6.2 shows the list of the features. In the Naver Q&A service, multiple answers are possible for a single question and the questioner selects the best answer. Unless otherwise mentioned, we extract features only from the best answer. The following is a detailed explanation of each individual feature.

Answerer's Acceptance Ratio The ratio of best answers to all the answers that the answerer answered previously.

Answer Length The length of the answer. Depending on the points of view, this feature can be thought of as a textual feature. However, we add this feature because it can be easily extracted without a serious analysis of the content of the text and is known to be helpful in measuring the quality of online writings [37].

Questioner's Self Evaluation The questioner gives from one to five stars($\star$) to the answer when they select the answer.

Answerer's Activity Level If a user asks and answers many times in the service, the user gets a high activity score.

Answerer's Category Specialty If a user answers many questions in a category, the user gets a high category specialty score for that category.

Print Counts The number of times that users print the answer.

Copy Counts The number of times that users copy the answer to their blog.

Users' Recommendation The number of times the Q&A pair is recommended by other users.

Editor's Recommendation Sometimes editors of the service upload interesting Q&A pairs on the front page of the service.

Sponsor's Answer For some categories, there are approved answerers who are nominated as a 'sponsor' of the category.

Click Counts The number of times the Q&A pair is clicked by other users.

Number of Answers The number of answers for the given question.

Users' Dis-Recommendation The number of time the Q&A pair is dis-recommended by other users.

Although some features are specific to the Naver service, other features such as answer length, the number of answers and click counts are common in many Q&A services. Some features such as recommendation counts and evaluation scores using stars can be found in many other web services. As can be seen from table 6.2, various numerical types are used to represent diverse features.

6.3.2 Feature Analysis

We calculate the correlation coefficient (or Pearson's correlation) between individual features and the manual quality judgment scores (good answers have higher scores: Bad=0, Medium=1, Good=2). The third column in table 6.2 shows the coefficient values.

Surprisingly, "Questioner's Self Evaluation" is not the feature that has the strongest correlation with the quality of the answer. This means the questioner's self-evaluation

Features	Type	Corr
Answerer's Acceptance Ratio	Percentile	0.1837
Answer Length	Integer	0.1733
Questioner's Self Evaluation	1,...5	0.1675
Answerer's Activity Level	Integer	0.1430
Answerer's Category Specialty	Integer	0.1037
Print Counts	Integer	0.0528
Copy Counts	Integer	0.0469
Users' Recommendation	Integer	0.0351
Editor's Recommendation	Binary	0.0285
Sponsor's Answer	Binary	0.0232
Click Counts	Integer	-0.0085
Number of Answers	Integer	-0.0297
User's Dis-Recommendation	Integer	-0.0596

Table 6.2. List of features. The second column shows numerical types of the features. The last column shows the correlation coefficients between the feature values and the manually judged quality scores. Higher correlation means the feature is a better indicator to predict the quality of answers. Minus values means there are negative correlations.

is subjective and often does not agree with other users opinion about the answer. Many people simply appreciate getting answers from other people regardless of the quality of the answers, and give high scores for most of the answers. This user behavior may be related to the culture of Korean users. Performing similar analysis with other user groups, for example with North American users, may give an interesting comparison.

"Sponsor's Answer" and "Editor's Recommendation" are good features because they always guarantee the quality of answers but only small number of Q&A pairs are recommended by editors or written by sponsors. Therefore, these features have little impact on overall performance and the coefficient values are relatively small.

With the exception of the answer length, most of the important features are related to various properties of the answerer. This implies that knowing about the answerer is very important in estimating the quality of answers. This kind of user information is very hard to acquire by analyzing text of answers.

6.3.3 Feature Conversion using Kernel Density Estimation

We use the Maximum entropy approach to construct our quality predictor. Maximum entropy models require monotonic features that always represent stronger evidence with bigger values. For example, the number of recommendations is a monotonic feature since more recommendations means better quality. However, the length of an answer is not a monotonic feature because longer answers do not always mean better answers.

Most of the previous work [49, 53] on text classification using the maximum entropy approach used only monotonic features such as frequency of words or n-grams. Therefore, little attention was given to solve the problem of non-monotonic features. However, we have non-monotonic features and need to convert these features into monotonic features.

We propose using kernel density estimation (KDE) [24]. KDE is a nonparametric density estimation technique that overcomes the shortcomings of histograms. In KDE, neighboring data points are averaged to estimate the probability density of a given point. We use the Gaussian kernel to get more influence from closer data points. The probability of having a good answer given only the answer length, $P(good|AL)$, can be calculated from the density distributions.

$$P(good|AL) = \frac{P(good)F(good|AL)}{P(good)F(good|AL) + P(bad)F(bad|AL)} \tag{6.1}$$

where AL denotes the answer length and $F(\cdot)$ is the density function estimated using KDE. $P(good)$ is the prior probability of having a good quality answer estimated from the training data using the maximum likelihood estimator. $P(bad)$ is measured in the same way.

Figure 6.3 shows density distributions of good quality answers and bad quality answers according to the answer length. Good answers are usually longer than bad answers but very long and bad quality answers also exist. The graph shows $P(good|AL)$

Features	**Corr (Original)**	**Corr (KDE)**
Answer Length	0.1733	0.4285
Answerer's Activity Level	0.1430	0.1982
Answerer's Category Specialty	0.1037	0.2103

Table 6.3. Feature conversion results. The second column represents the correlation between the raw feature value and the quality scores. The third column shows the correlation coefficients after converting features using kernel density estimation. Much stronger correlations are observed after the conversion.

calculated from the density distributions. The probability initially increases as the answer length becomes longer but eventually starts decreasing. The probability that an answer is high quality is high for average-length answers, but low for very long answers. This accurately reflects what we see in practice in the Naver data.

We use $P(good|AL)$ as our feature value instead of using the answer length directly. This converted feature is monotonic since a bigger value always means stronger evidence. The 894 training samples are used to train the kernel density estimation module. Table 6.3 shows the power of this conversion. We calculate the correlation coefficients again after converting a few non-monotonic features. In the case of the answer length, the strength of the correlation is dramatically improved and it becomes the most significant feature.

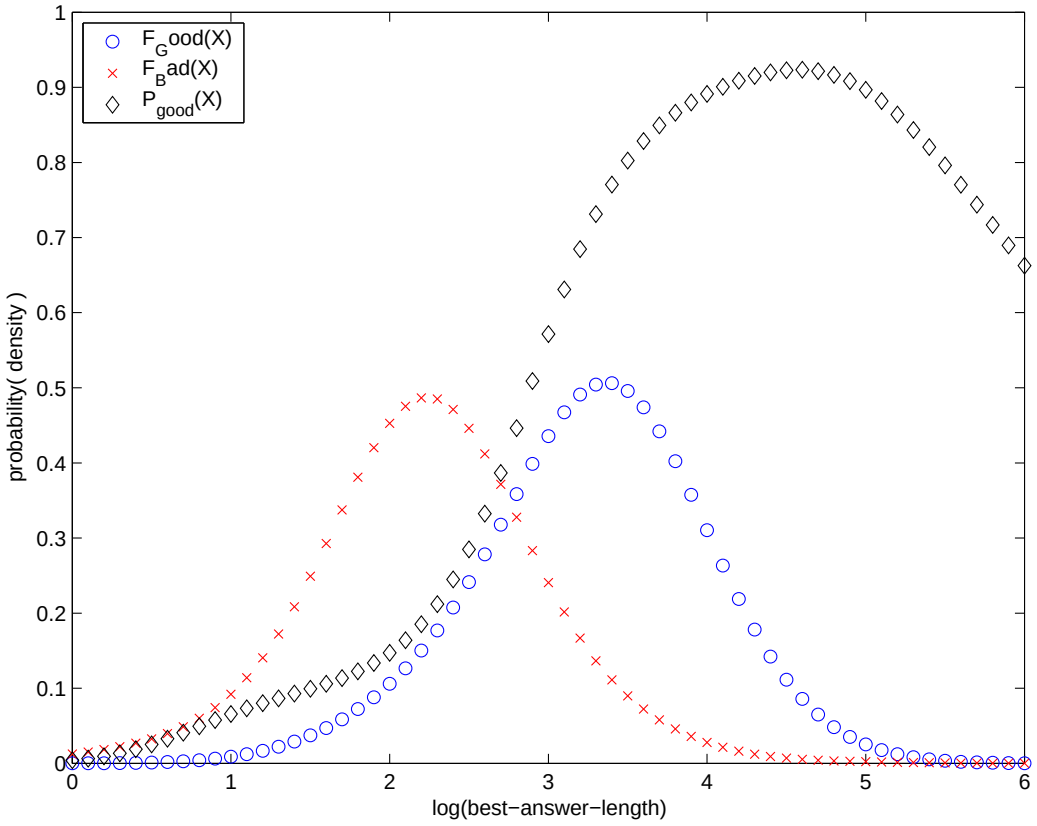


Figure 6.3. Density distributions of good answers and bad answers measured using KDE. The x axis is log(answer length) and the y axis is the density or the probability. The graph also shows the probability of having a good answer given the answer length.

6.4 Answer Quality Estimation using Maximum Entropy

We use the maximum entropy approach to build our quality predictor for the following reasons. First, the approach generates purely statistical models and the output of the models is a probability. The probability can be easily integrated into other statistical models. Our experimental results show the output can be seamlessly combined with statistical language models. Second, the model can handle a large number of features and it is easy to add or drop features. The models are also robust to noisy features.

We assume that there is a random process that observes a Q&A pair and generates a label y, an element of a finite set $Y = \{good, bad\}$. Our goal is to make a stochastic model that is close to the random process. We construct a training dataset by observing the behavior of the random process. The training dataset is $(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)$. x_i is a question and answer pair and y_i is a label that represents the quality of the answer. We make 894 training samples from the training data.

6.4.0.1 Predicate Functions and Constraints

We can extract many statistics from the training samples and the output of our stochastic model should match these statistics as much as possible. In the maximum entropy approach, any statistic is represented by the expected value of a feature function. To avoid confusion with the document features, we refer to the feature functions as predicates. We use 13 predicates. Each predicate corresponds to each document feature that we explained in the previous section.

$$f_i(x, y) = \begin{cases} kde(x_{fi}) & \text{if } i^{th}\text{feature is non-monotonic} \\ x_{fi} & \text{otherwise} \end{cases} \tag{6.2}$$

where $f_i(x, y)$ is the i^{th} predicate and x_{fi} is the actual value of the i^{th} feature in Q&A pair x.

The expected value of a predicate with respect to the training data is defined as follows,

$$\tilde{p}(f_i) = \sum_{x,y} \tilde{p}(x,y) f_i(x,y) \tag{6.3}$$

where $\tilde{p}(x,y)$ is a empirical probability distribution that can be easily calculated from the training data. The expected value of the predicate with respect to the output of the stochastic model should be the same with the expected value measured from the training data.

$$\sum_{x,y} \tilde{p}(x,y) f_i(x,y) = \sum_{x,y} \tilde{p}(x) p(y|x) f_i(x,y) \tag{6.4}$$

where $p(y|x)$ is the stochastic model that we want to construct. We call the equation (4) a constraint. We have to choose a model that satisfies these constraints for all predicates.

6.4.0.2 Finding Optimal Models

In many cases, there are infinite number of models that satisfy the constraints explained in the previous subsection. In the maximum entropy approach, we choose the model that has maximum conditional entropy

$$H(p) = -\sum_{x,y} \tilde{p}(x) p(y|x) \log p(y|x) \tag{6.5}$$

There are a few algorithms that can find optimal models. Optimal models must satisfy the constraints and maximize the entropy. Generalized Iterative Scaling and Improved Iterative Scaling have been widely used. We use Limited Memory Variable Metric method which is very effective for Maximum Entropy parameter estimation [45]. We use Zhang Le's maximum entropy toolkit[1] for the experiment.

[1] http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html

The model is represented by a set of parameters λ. Each predicate has a corresponding parameter and the following is the final equation to get the probability of having a good answer or bad answer.

$$p(y|x) = \frac{1}{Z(x)} \exp\left[\sum_{i=1}^{13} \lambda_i f_i(x, y)\right] \tag{6.6}$$

where $Z(x)$ is a normalization factor.

6.4.0.3 Predictor Performance

We build the predictor using the 894 training samples and test using the 1700 test samples. The output of the predictor is the probability that the answer of the given Q&A pair is good. The average output for good Q&A pairs is 0.9227 and the average output for bad Q&A pairs is 0.6558. In both cases, the averages are higher than 0.5 because the prior probability of having a good answer is high. As long as this difference is consistent, it is possible to build an accurate classifier using this probability estimate.

We rank 208 bad examples and 1099 good examples in the test collection together by the descending order of the output values. Figure 6.4 shows the quality of the ranking using the recall-precision graph. The predictor is significantly better than random ranking. In the top 100, all Q&A pairs are good. The top 250 pairs contain 2 bad pairs and the top 500 pairs contain 9 bad pairs. The results show that the predictor has the ability to discriminate good answers from bad answers. Increasing the size of the training samples may lead to improved performance. In the next section, we investigate the effectiveness of the predictor in the context of retrieval.

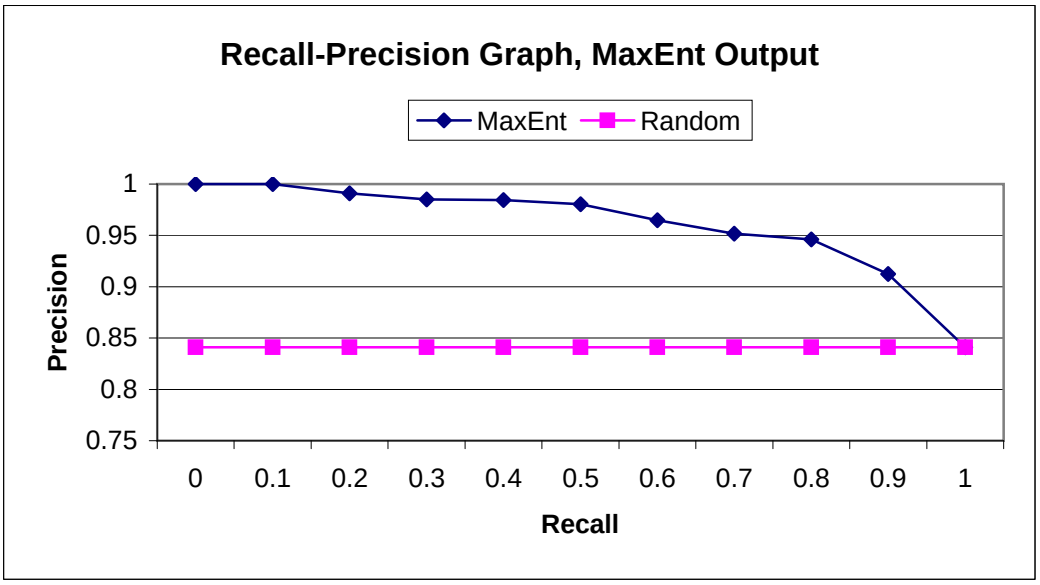


Figure 6.4. Performance of the quality predictor. 11pt recall-precision graph. Note that the y-axis scale starts from 0.75. 'Random' is the result of random ranking that positions Q&A pairs randomly.

6.5 Retrieval Experiments

In this section, we test whether the quality measure can improve user experience. The user experience is measured by employing new types of relevance judgment files that consider both the relevance and the quality of Q&A pairs. In this section, the quality score is integrated into the query likelihood model. In the next chapter, we incorporate the quality score into the translation-based retrieval model that we propose in chapter 4.

For the retrieval experiment, we use the Naver collection B described in section 3.3. Our baseline retrieval model is the query likelihood language model. We incorporate the quality measure into the baseline system and compare retrieval performance using different types of relevance judgment files. Here, we briefly explain the query likelihood model again to explain how we incorporate the quality score.

6.5.1 Retrieval Framework

In the query likelihood retrieval model, the similarity between a query and a document is given by the probability of the generating the query from the document language model.

$$sim(Q, D) = P(D|Q) = P(D)P(Q|D)/P(Q) \tag{6.7}$$

$P(Q)$ is independent of documents and does not affect the ranking. For the document model, usually, i.i.d sampling and unigram language models are used.

$$P(Q|D) = P(D) \prod_{w \in Q} P(w|D) \tag{6.8}$$

$P(D)$ is the prior probability of document D. Query independent prior information such as time, quality and popularity have been successfully integrated into the model using the prior probability [35, 78, 41]. Our quality score is given as a probability and it is also query independent. Therefore, the quality score can be plugged into the retrieval model as a document prior without any modification such as normalization. Therefore, in our approach, $P(D) = p(y|x = D)$. $p(y|x = D)$ is given as in equation 6.6.

To avoid zero probabilities and estimate more robust document language models, documents are smoothed using a background collection,

$$P(w|D) = (1 - \lambda)P_{ml}(w|D) + \lambda P_{ml}(w|C) \tag{6.9}$$

$P_{ml}(w|C)$ is the probability that the term w is generated from the collection C. $P_{ml}(w|C)$ is estimated using the maximum likelihood estimator. λ is the smoothing parameter. We use Dirichlet smoothing [76]. The optimal parameter value is found

by exhaustive search of the parameter space. We use the implementation of the query likelihood retrieval model in the Lemur toolkit[2].

6.5.2 Evaluation Method

In order to automatically evaluate retrieval performance, usually a relevance judgment file is created. This file contains lists of relevant documents to queries and an evaluation system looks up this file to automatically assess the performance of search engines. We created three different relevance judgment files. The first one (Rel_1) considers only the relevance between the query and the question. If the question part of a Q&A pair addresses the same information need as the query, the Q&A pair is considered to be relevant to the query. The second file (Rel_2) considers both the relevance and the quality of Q&A pairs. If the quality of the the answer is judged as 'bad', then the Q&A pair is removed from the relevance judgment file even if the question part is judged as relevant to the query. The last judgment file (Rel_3) requires a stronger requirement of quality. If the quality of the answer is judged 'bad' or 'medium', then the Q&A pair is removed from the file and only relevant and good quality Q&A pairs remain in the file.

Rel_2 is a subset of Rel_1 and Rel_3 is a subset of Rel_2. From table 6.1, Rel_1 contains 1700 Q&A pairs, Rel_2 has 1492 pairs and Rel_3 includes 1099 pairs. Most previous research in FAQ retrieval considered only the relevance of the question (Rel_1). We believe the performance measured using Rel_2 or Rel_3 is closer to real user satisfaction, since they take into account both relevance and quality.

---

[2]http://www.lemurproject.org/

Mean Average Precisions

	Rel_1	Rel_2	Rel_3
Without Quality	0.294	0.267	0.222
With Quality	0.322*	0.316*	0.290*
P-value	0.007	1.97E-06	2.96E-11

Precisions at Rank 10

	Rel_1	Rel_2	Rel_3
Without Quality	0.366	0.313	0.236
With Quality	0.427*	0.404*	0.338*
P-value	3.59E-05	5.81E-09	1.18E-12

Table 6.4. Comparison of retrieval performance. The upper table shows mean average precisions and the lower table shows precisions at rank 10. Asterisks (*) denote the score is statistically significantly better than the score of the baseline system.

6.5.3 Experimental Results

We measure retrieval performance using various standard evaluation metrics such as the mean average precision, precision at rank 10 and 11pt recall-precision graphs. Table 6.4 and Figure 6.5 show the retrieval results.

Table 6.4 shows that the retrieval performance is significantly improved after adding the quality measure. Surprisingly, the retrieval performance is significantly improved even when we use the relevance judgment file that does not consider quality. This implies bad quality Q&A pairs tend not to be relevant to any query and incorporating the quality measure pulls down these useless Q&A pairs to lower ranks and improves the retrieval results overall.

Because Rel_2 has smaller number of relevant Q&A pairs and Rel_3 contains even smaller number of the pairs, the retrieval performance is lower. However, the performance drop becomes much less dramatic when we integrate the quality measure. The retrieval performance evaluated by Rel_2 is better than the performance evaluated by Rel_1, if we incorporate the quality measure.

The third rows in Table 6.4 show the P-values of the statistical significant test[3]. The results show all the improvements are significant. The significance of the improvement is higher when we use stricter requirements for the correct Q&A pairs. Figure 6.5 shows 11pt recall-precision graphs. In all recall levels, we get improved precisions by adding the quality measure. The improvement becomes bigger when we use Rel_3 instead of Rel_1.

6.6 Summary

In this chapter, we showed how we could systematically and statistically use non-textual features that are commonly recorded by web services, to improve search quality. We estimated the quality of answers reliably using the maximum entropy approach and kernel density estimation. The predicted quality scores were successfully incorporated into the query likelihood language model. In the next chapter, we integrate the quality scores into TransLM.

[3]Wilcoxon signed rank test.

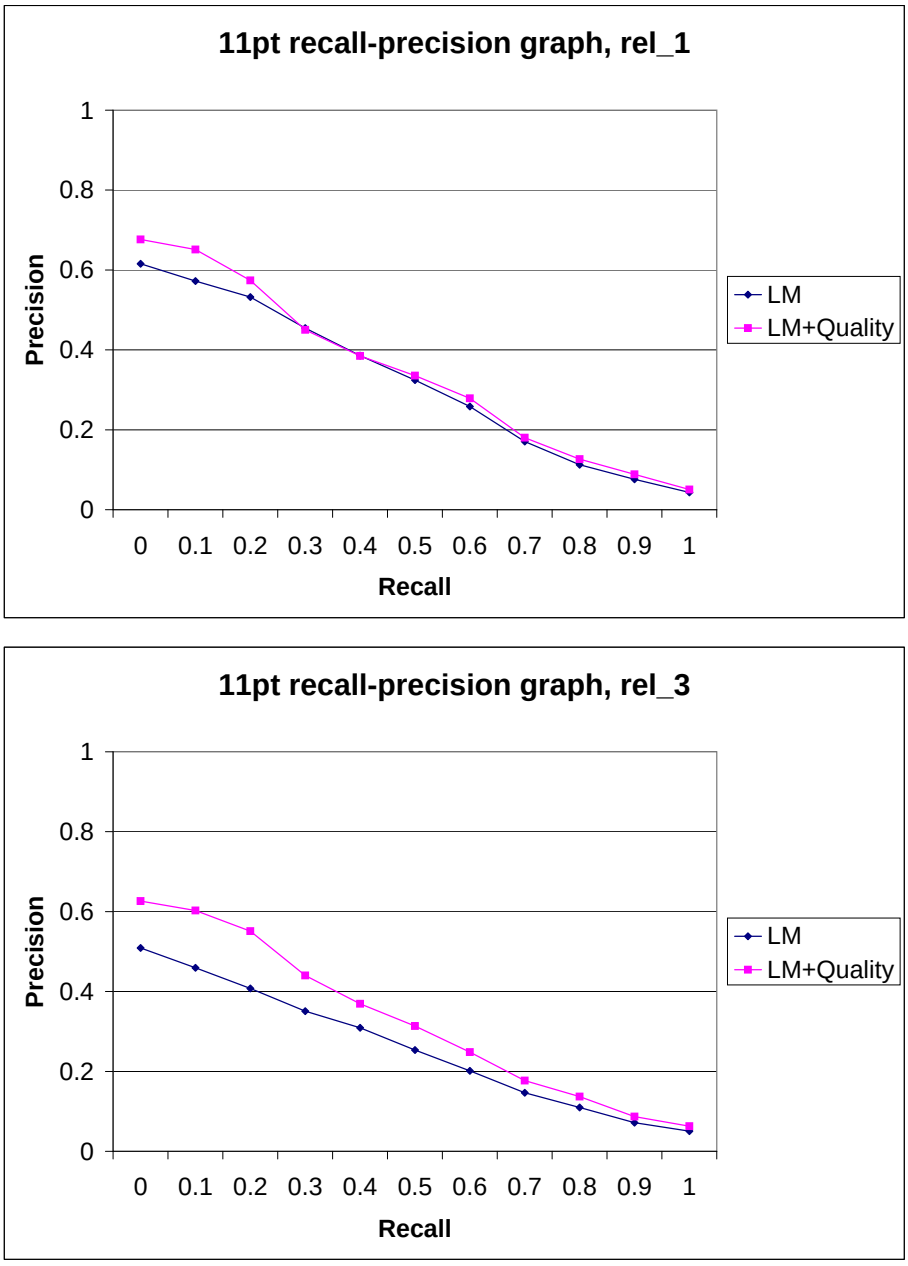


Figure 6.5. 11pt recall precision graphs. LM is the result of using the query likelihood retrieval model. LM+Quality is the result after incorporating the quality measure into the same retrieval model.

CHAPTER 7

EXPERIMENTS

In this chapter, we describe experiments we conducted to evaluate the effectiveness of our retrieval system. We compare it to other state of the art retrieval systems such as query likelihood, Okapi BM25, relevance models and other translation-based retrieval models proposed by Murdock [48] and Jeon [25].

The main task is to find semantically identical questions to the user query. All three test collections introduced in chapter 3 are used to evaluate the performance of our system. We focus mainly on long question-type queries but we also test our system with short keyword queries. The utility of category-specific word translations is tested. We also evaluate the effectiveness of combining the quality scores into our retrieval model.

Two additional experiments were performed to verify whether our retrieval model and the word relationships learned from Q&A collections can be used for other information retrieval tasks. The first task is finding relevant answer passages for a given question. The second one is finding relevant news articles for a given keyword query.

In all experiments, our approach outperforms other models. Retrieval examples show that our model has ability to address the word mismatch problem. After incorporating the quality scores in the retrieval model, our system can effectively find relevant and high quality answers.

We begin this chapter by introducing our evaluation metrics and baseline retrieval models.

7.1 Evaluation Method

In this section, we explain how we evaluate and compare our system with respect to other baseline systems. Our evaluation metrics and methods for statistical significance test are presented.

7.1.1 Evaluation Metrics

Precision and recall are the basic measures used in evaluating search systems. **Recall** is the ratio of retrieved relevant items to all relevant items in the collection. **Precision** is the ratio of retrieved relevant items to items retrieved at a given rank. In many cases, including ours, accurate estimation of recall is hard because it is difficult to know the total number of relevant items in the collection.

Most commercial search services emphasize returning relevant documents earlier especially in the first page of the search results, since most users browse only the first page. Therefore, they are mostly interested in precisions of top n (the number of documents that the first page can display) documents. Anther advantage of using precisions at fixed ranks is that it is intuitive and easy to understand. In this dissertation, we also emphasize returning relevant Q&A pair earlier and use precisions at rank 10 and 20 to evaluate our system. We use **P@n** to denote the precision at rank n. For example, P@10 is the precision at rank 10.

Mean Average Precision (MAP) is another evaluation metric that we use in this work. MAP is the most widely used evaluation measure in information retrieval. Average precision is the average of precisions at the ranks of relevant documents. MAP is the mean of average precisions over multiple queries. It has been shown that MAP can reliably identify performance gaps among retrieval systems [12]. MAP is less intuitive than P@n but it takes into account both recall and precision.

7.1.2 Significance Test

A statistical significance test is a mathematical test for determining if the performance gap between two systems is significant. In this thesis, we use the Wilcoxon signed rank test. The following is a brief explanation about the Wilcoxon test from MathWorld[1].

> "A nonparametric alternative to the paired t-test which is similar to the Fisher sign test. This test assumes that there is information in the magnitudes of the differences between paired observations, as well as the signs. Take the paired observations, calculate the differences, and rank them from smallest to largest by absolute value. Add all the ranks associated with positive differences, giving the T+ statistic. Finally, the P-value associated with this statistic is found from an appropriate table. The Wilcoxon test is an R-estimate."

A good property of the Wilcoxon test is that both the signs and the magnitude of differences are considered together, therefore it is more reliable than typical sign tests. We decide a system is better than the other when the P-value is less than 0.05 (confidence level of 95%).

[1]Weisstein, Eric W. "Wilcoxon Signed Rank Test." From MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/WilcoxonSignedRankTest.html

7.2 Baseline Retrieval Models

We choose three state of the art information retrieval models as our baselines. They are query likelihood language models (**LM**), relevance models (**RM**) and Okapi BM25 (**Okapi**). The query likelihood language model is described in section 4.3.1 and this section explains relevance models and Okapi BM25. We also compare our system with two other translation-based retrieval models. These models are introduced in section 4.3.

7.2.1 Okapi BM25

Okapi retrieval models [57, 31] are based on the probabilistic retrieval framework developed by Stephen E. Robertson and others in 1980s and 1990s. Okapi BM25 is the most well known and widely used retrieval function among a family of Okapi functions. It has shown good performance in many information retrieval tasks. The scoring function is given as follows:

For a query $Q = \{q_1, q_2, ..., q_n\}$ and a document D,

$$Sim(D, Q) = \sum_{i=1}^{n} \frac{\#(q_i, D) \cdot (k_1 + 1)}{\#(q_i, D) + k_1 \cdot (1 - b + b\frac{|D|}{avgDL})} \cdot idf(q_i) \tag{7.1}$$

$$idf(q_i) = log\frac{N - \#(q_i) + 0.5}{\#(q_i) + 0.5} \tag{7.2}$$

where $\#(q_1, D)$ is the term frequency of q_i in D, $|D|$ is the length of the document D, $avgDL$ is the average length of documents, N is the total number of documents and $\#(q_i)$ is the number of documents containing q_i. k_1 and b are free parameters.

A typical (standard) parameter setting for Okapi BM25 is $k_1 = 1.2$ and $b = 0.75$. However, we found better performance could often be achieved with non-standard parameter settings in our experiments. Therefore, we always exhaustively search its parameter space to find optimal parameter values for every experiment reported in this thesis. This approach always set the baseline as high as possible.

7.2.2 Relevance Models

The concept of relevance models was proposed by Lavrenko and Croft [38]. They assumed that the query and the relevant documents are sampled from the same underlying relevance model. In this retrieval framework, the document retrieval task is measuring distances between document language models and the relevance model of the query. Usually, Kullback Leibler divergence is used to measure the distance. The relevance model $P(w|R)$ is estimated using a joint probability of sampling the word w together with the query.

$$P(w|R) \approx P(w|Q) = \frac{P(w, q_1, q_2, ..., q_n)}{P(q_1, q_2, ..., q_n)} \tag{7.3}$$

$$P(w, q_1, q_2, ..., q_n) = \sum_{D \in C} P(D) P(w|D) \prod_{i=1}^{n} P(q_i|D) \tag{7.4}$$

where D is a document in a collection C and $P(w|D)$ is the probability that the term w is sampled from the document language model of D. A constant document prior $P(D)$ is assumed and the smoothed maximum likelihood estimator is used to build document language models. Typically, top m documents are used instead of all the documents in the collection.

When there are a small number of relevant documents or the quality of the initial retrieval is poor, the relevance model can loose original query terms. To avoid this situation, the query language model is often added to the relevance model. This approach, namely RM3 empirically gives better performance than original relevance models. In this thesis, we use RM3.

The relevance model has a few hidden parameters such as the number of feedback documents and the number of expanded terms. In this thesis, we use default settings implemented in the Lemur toolkit. The only parameters that we tune are the Dirichlet prior for the initial ranking and the mixing parameter that controls the amounts of the query language model in the updated relevance model.

7.3 Q&A Retrieval Experiments

In this section, we evaluate the performance of our system for the task of finding semantically identical questions from a large collection of questions. All three test collections introduced in chapter 3 are used to evaluate the performance of our system: (subsection 7.3.1 - 7.3.3). We focus mainly on long question-type queries but we also test our system with short keyword queries: (subsection 7.3.4). The utility of category-specific word translations is also tested: (subsection 7.3.5). The final experiment is designed to evaluate the effectiveness of combining the quality scores into our retrieval model: (subsection 7.3.6).

7.3.1 Experiments on Wondir collection

For this experiment, we used the Wondir test collection described in section 3.1. The test collection consists of 1 million Q&A pairs, 50 questions and a set of relevance judgments (220 relevant Q&A pairs). We used K-stemmer [36] and did not remove stops words. Word relationships learned from the Wondir collection were used because they reflect Wondir community's interests and word usages better than any other word relationships learned from other Q&A collections.

7.3.1.1 Comparison of Retrieval Models

Table 7.1 is the summary of the retrieval results. The reported results are the best scores that the models achieve after an exhaustive search on their parameter spaces. The last column is the number of parameters that we optimized for each model. TransLM (our method) outperforms all other retrieval models. Relevance model outperforms both the query likelihood language model and Okapi BM25 but it is worse compared to TransLM and Jeon's model. Murdock's model is no better than baseline models in this experiment. TransLM achieves statistically significant improvements in all evaluation measures over all baseline models. Statistical significance was measured by Wilcoxon signed rank test at a confidence level of 95%.

Baseline Retrieval Models

Model	**MAP**	**P at 10**	**P at 20**	#param
LM	0.3024	0.2158	0.1447	1
Okapi	0.2994	0.2184	0.1421	2
RM	0.3458	0.2158	0.1553	2

Translation-based Retrieval Models

Model	**MAP**	**P at 10**	**P at 20**	#param
Murdock	0.2999	0.2053	0.1421	1
Jeon	0.3576	0.2500*	0.1697	1
TransLM	**0.3816***	**0.2789***	**0.1803***	2

Table 7.1. Summary of Question Retrieval Results - Wondir Collection. Word relationships: $P(A|Q)$. Asterisks* denote the score is statistically significantly better than the scores of all baseline models.

Figure 7.1 shows the retrieval effectiveness of TransLM depending on the mixing parameter. The upper figure shows results evaluated by MAP. The lower figure presents the results measured by P@20. The x-axis is the mixing parameter. Bigger value means more impact from the translation component in the model. The relevance model (RM) that we use in our experiments linearly combines the query language model and the original relevance model to build an updated relevance model. The mixing parameter controls the proportion of the relevance model scores in the final scores. Higher value for the mixing parameter means higher contributions from the relevance model. The graphs show that the performance of TransLM increases as the mixing parameter becomes bigger and reaches its peak when the mixing parameter is around 0.7.

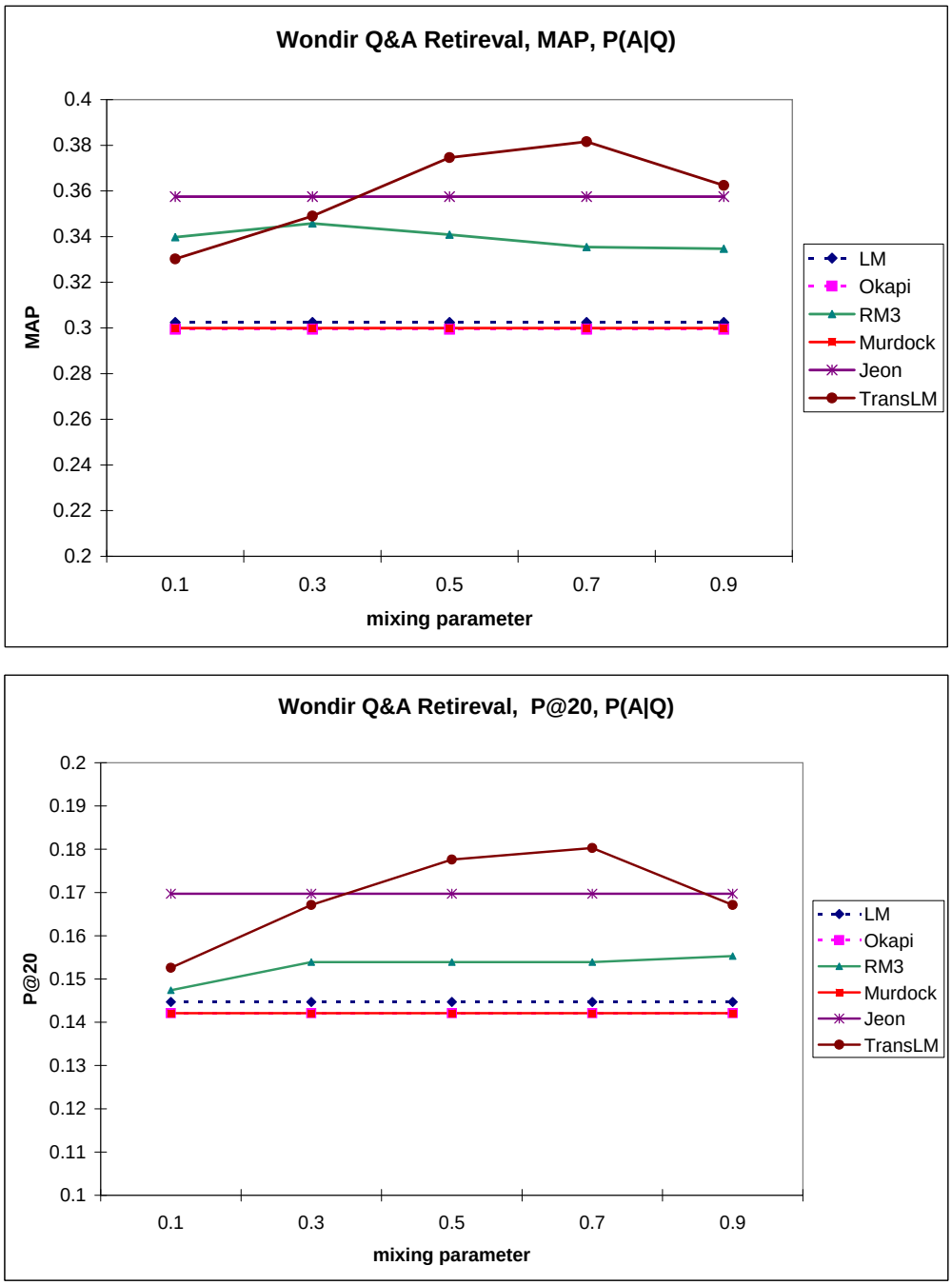


Figure 7.1. Question Retrieval Results. Wondir Collection. Comparison of retrieval models. The X axis is the mixing parameter β in equation 4.12. The upper figure shows results evaluated by MAP. The lower figure presents the results measured by P@20.

7.3.1.2 Comparison of Translation Tables

Figure 7.2 shows the performance of TransLM depending on different types of word relationships discussed in subsection 5.3.1. The best performance is achieved using $P(A|Q)$. $P(Q|A)$ is the worst. Our task is sampling questions (queries) from questions. To explicitly simulate this task, we built two different types of word relationships: $P(Q|Q)$ and $P(Q \leftrightarrow Q)$. $P(Q|Q)$ is generated by convoluting $P(Q|A)$ and $P(A|Q)$. $P(Q|Q)$ shows small improvements over $P(Q|A)$ but it is worse compared to $P(A|Q)$. The convolution of the good translation table and the bad translation table seems to make a mediocre quality table. $P(Q \leftrightarrow Q)$ is learned from the question pairs that are automatically collected using the similarities between answers. $P(Q \leftrightarrow Q)$ shows overall good performance although it is worse when compared to $P(A|Q)$.

Figure 7.3 compares the performance of three translation-based retrieval models depending on different types of word relationships. The graphs show that TransLM is better than the other models regardless of the choice of word relationship types. Jeon's model is consistently better than Murdock's model in this collection.

Table 7.2 summarizes all experimental results presented in this subsection. The second column shows the type of word relationships used with translation-based retrieval models. TransLM with $P(A|Q)$ achieves the best performance in all measures and the improvement is statistically significant.

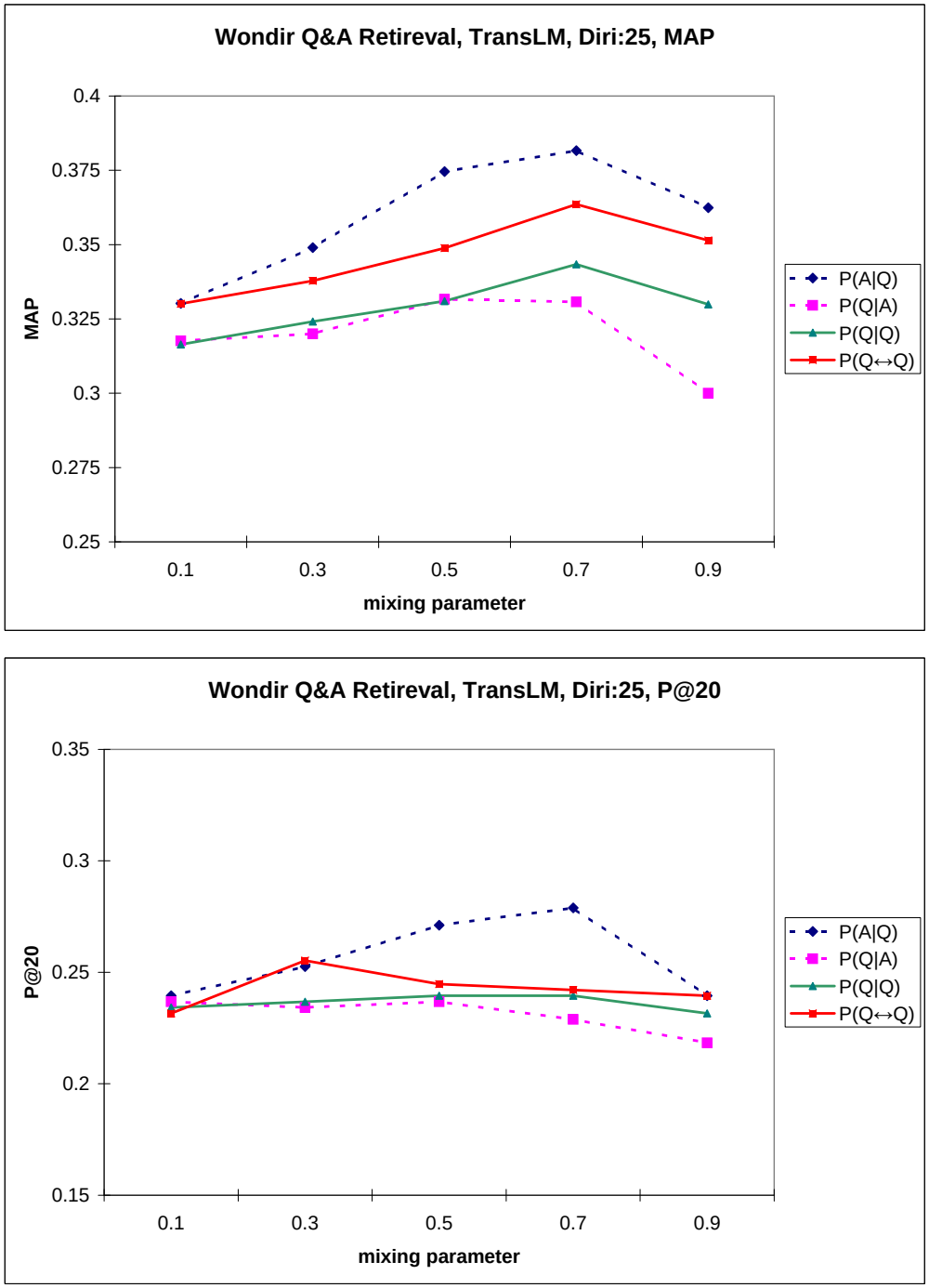


Figure 7.2. Comparison of Translation Tables. Question Retrieval Task. Wondir collection. The X axis is the mixing parameter β in equation 4.12. The upper figure shows results evaluated by MAP. The lower figure presents the results measured by P@20.

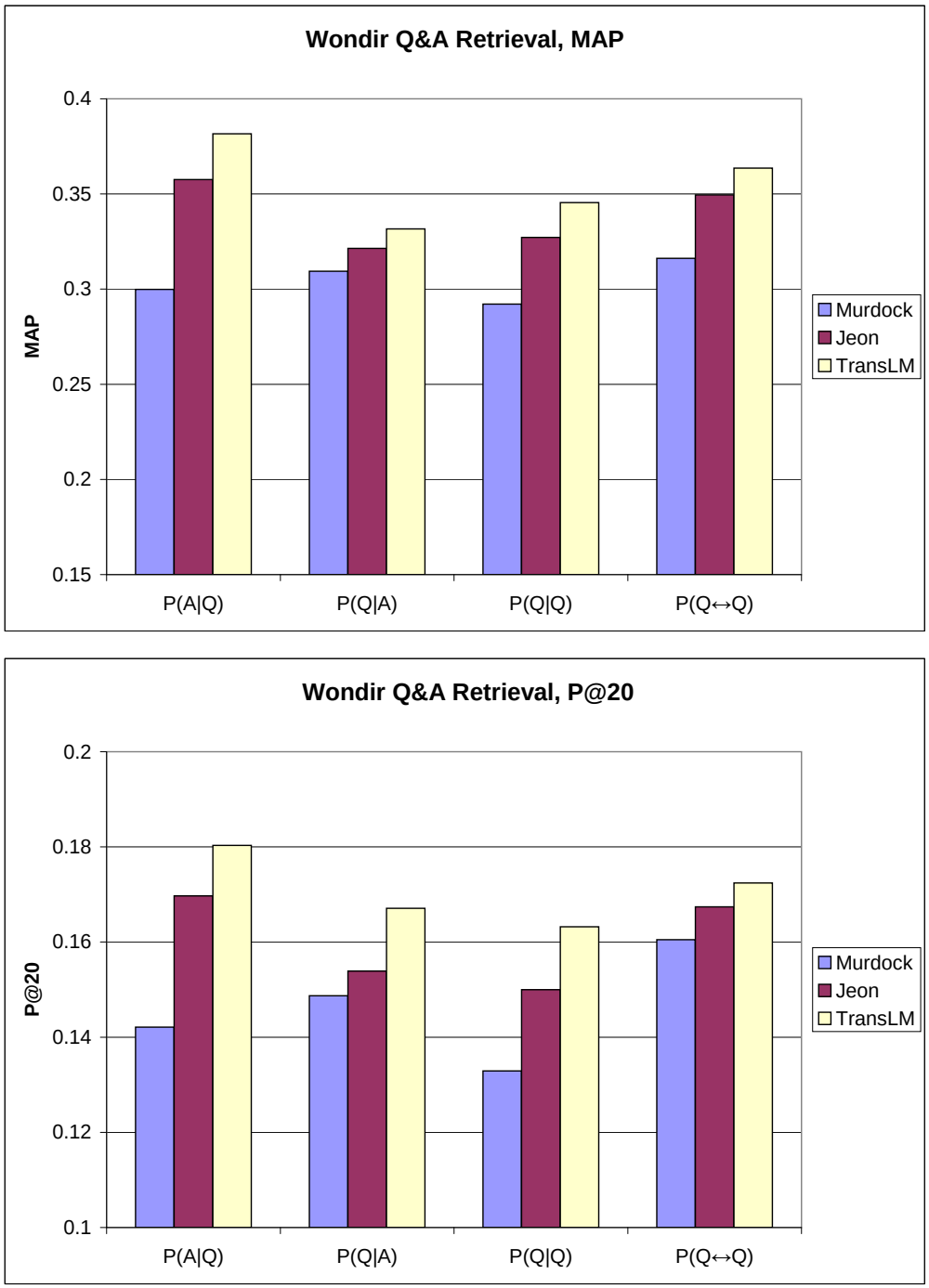


Figure 7.3. Comparison of Translation-Based Retrieval Models. Question Retrieval Task. Wondir collection. The upper figure shows results evaluated by MAP. The lower figure presents the results measured by P@20.

Baseline Retrieval Models					
Model	TTable	MAP	P at 10	P at 20	#
LM		0.3024	0.2158	0.1447	1
Okapi		0.2994	0.2184	0.1421	2
RM		0.3458	0.2158	0.1553	2

Translation-based Retrieval Models					
Model	TTable	MAP	P at 10	P at 20	#
Murdock	$P(A\|Q)$	0.2999	0.2053	0.1421	1
Jeon	$P(A\|Q)$	0.3576	0.2500*	0.1697	1
TransLM	$P(A\|Q)$	**0.3816***	**0.2789***	**0.1803***	2
Murdock	$P(Q\|A)$	0.3094	0.2184	0.1487	1
Jeon	$P(Q\|A)$	0.3214	0.2184	0.1539	1
TransLM	$P(Q\|A)$	0.3316	0.2368	0.1671	2
Murdock	$P(Q\|Q)$	0.2921	0.1789	0.1329	1
Jeon	$P(Q\|Q)$	0.3272	0.2289	0.1500	1
TransLM	$P(Q\|Q)$	0.3455	0.2342	0.1632*	2
Murdock	$P(Q \leftrightarrow Q)$	0.3162	0.2289	0.1605	1
Jeon	$P(Q \leftrightarrow Q)$	0.3495	0.2184	0.1671	1
TransLM	$P(Q \leftrightarrow Q)$	0.3636	0.2421*	0.1724*	2

Table 7.2. Summary of Question Retrieval Results. Wondir Collection. Asterisks* denote the score is statistically significantly better than all baseline models.

Model	MAP	P at 10	P at 20
LM	0.1924	0.1320	0.098
Okapi	0.1948	0.1300	0.086
RM	0.1949	0.1300	0.098
Murdock	0.1948	0.1280	0.1010
Jeon	0.1920	0.1320	0.1130*
TransLM	**0.2307***	**0.148***	**0.1230***

Table 7.3. Summary of Question Retrieval Results - WebFAQ Collection. Word relationships: $P(A|Q)$. Asterisks* denote the score is statistically significantly better than the scores of all baseline models.

7.3.2 Experiments on WebFAQ collection

For this experiment, we used the WebFAQ test collection described in section 3.2. The test collection consists of 3 million FAQs, 50 questions and a set of relevance judgments (262 relevant FAQs). We used K-stemmer and did not remove stops words. Word relationships learned from the WebFAQ collection were used.

Table 7.1 is the summary of the retrieval results. The reported results are the best scores that the models achieve after an exhaustive search on their parameter spaces. TransLM outperforms all other retrieval models. Except TransLM, all other models show overall similar performance. In precisions at top 20, translation-based models outperform baseline models.

Figure 7.4 shows the retrieval effectiveness of TransLM depending on the mixing parameter. The performance increases as the mixing parameter becomes bigger and reaches its peak again when the parameter value in around 0.7. This is the identical pattern that we observed in the experiments with the Wondir collection.

Figure 7.5 provides a comparison of the translation-based retrieval models depending on various types of word relationships. TransLM is better than other translation-based models regardless of the word relationship types and evaluation measures. This time, Jeon's model is no better than Murdock's model in MAP but it is better when P@20 is used as an evaluation metric.

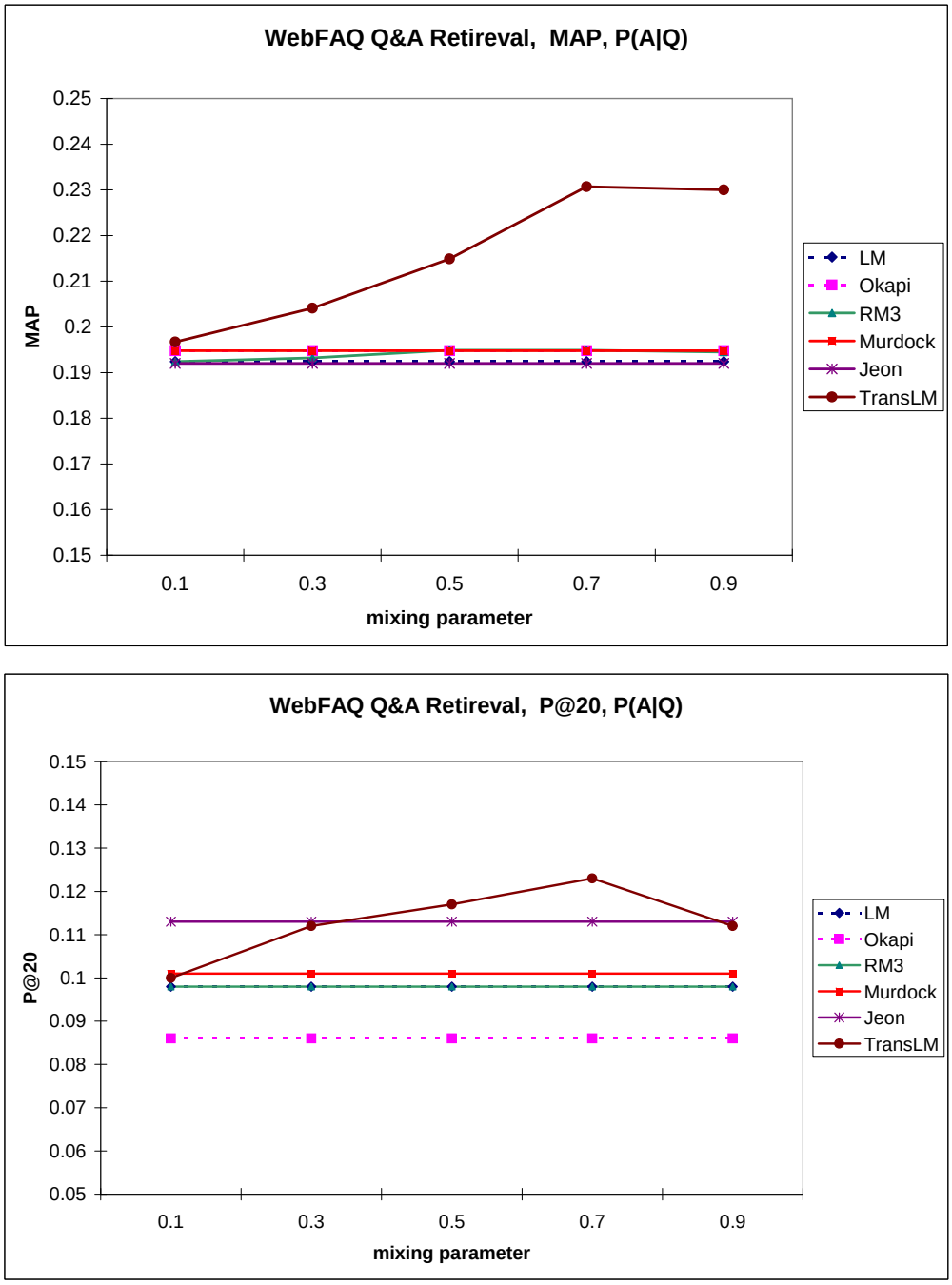


Figure 7.4. Question Retrieval Results. WebFAQ Collection. Comparison of retrieval models. The X axis is the mixing parameter β in equation 4.12. The upper figure shows results evaluated by MAP. The lower figure presents the results measured by P@20.

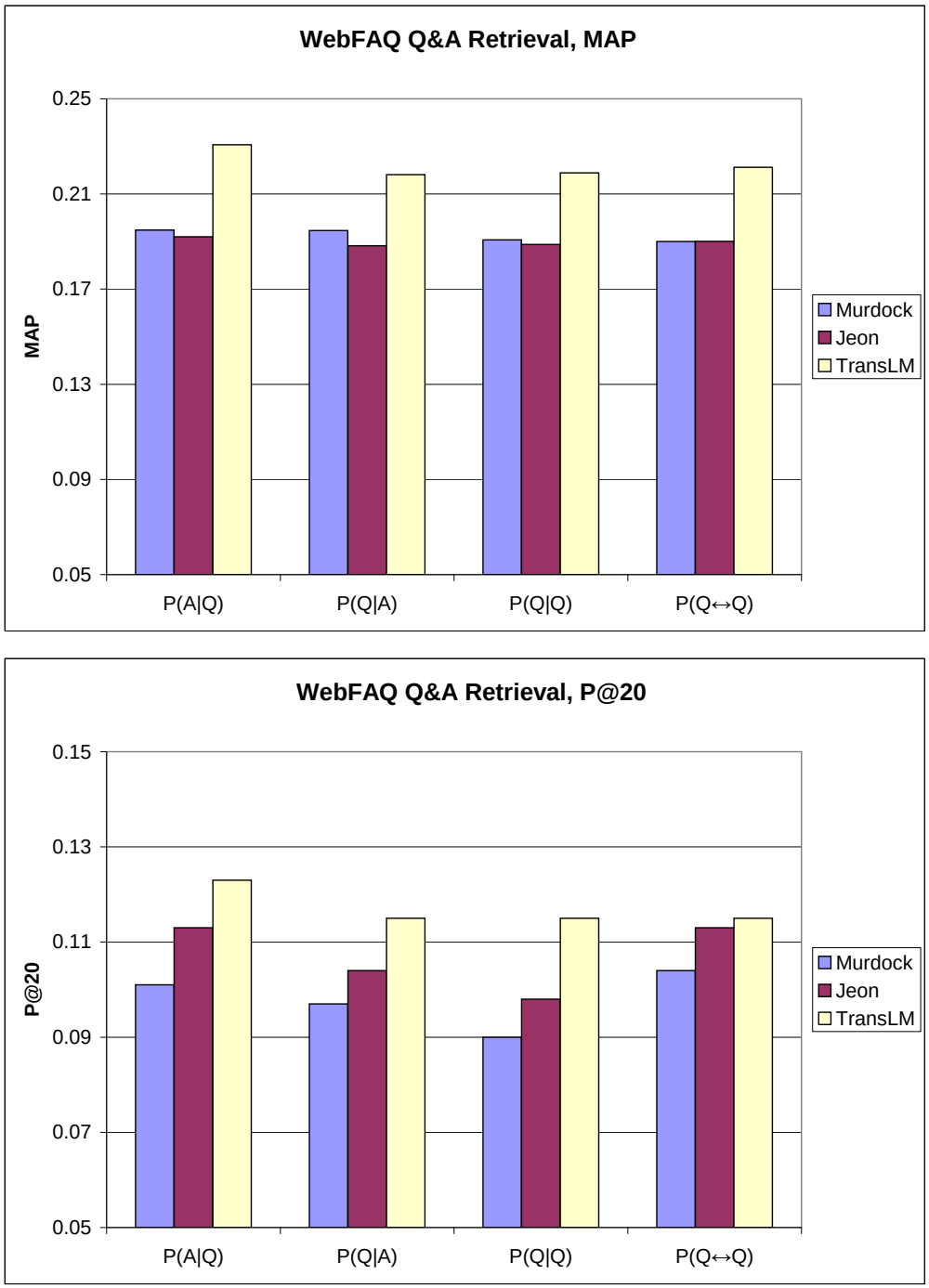


Figure 7.5. Comparison of Translation-Based Retrieval Models. Question Retrieval Task. WebFAQ collection. The upper figure shows results evaluated by MAP. The lower figure presents the results measured by P@20.

Baseline Retrieval Models				
Model	TTable	MAP	P at 10	P at 20
LM		0.1924	0.1320	0.098
Okapi		0.1948	0.1300	0.086
RM		0.1949	0.1300	0.098

Translation-based Retrieval Models				
Model	TTable	MAP	P at 10	P at 20
Murdock	$P(A\|Q)$	0.1948	0.1280	0.1010
Jeon	$P(A\|Q)$	0.1920	0.1320	0.1130*
TransLM	$P(A\|Q)$	**0.2307***	**0.148***	**0.123***
Murdock	$P(Q\|A)$	0.1946	0.1280	0.0970
Jeon	$P(Q\|A)$	0.1882	0.1300	0.1040
TransLM	$P(Q\|A)$	0.2181*	0.1440*	0.1150*
Murdock	$P(Q\|Q)$	0.1907	0.1320	0.0900
Jeon	$P(Q\|Q)$	0.1888	0.1360	0.0980
TransLM	$P(Q\|Q)$	0.2189*	0.1420*	0.1150*
Murdock	$P(Q \leftrightarrow Q)$	0.1900	0.1280	0.1040
Jeon	$P(Q \leftrightarrow Q)$	0.1901	0.1300	0.1130*
TransLM	$P(Q \leftrightarrow Q)$	0.2212*	0.1400	0.1150*

Table 7.4. Summary of Question Retrieval Results. WebFAQ Collection. Asterisks* denote the score is statistically significantly better than all baseline models.

Table 7.4 summarizes all experimental results presented in this subsection. Overall, we observe very similar results to the results on the Wondir collection. The second column shows the type of word relationships used with translation-based retrieval models. TransLM with $P(A|Q)$ achieves the best performance in all measures and the improvement is statistically significant.

Baseline Retrieval Models

Model	**MAP**	**P at 10**	**P at 20**
LM	0.3488	0.4304	0.3652
Okapi	0.3533	0.4391	0.3674
RM	0.2719	0.3652	0.2717

Translation-based Retrieval Models

Model	**MAP**	**P at 10**	**P at 20**
Murdock	0.3754	0.4478	0.4087*
Jeon	0.3870	0.4261	0.4043*
TransLM	**0.4046***	**0.4609**	**0.4109***

Table 7.5. Summary of Question Retrieval Results - Naver Collection A. Word relationships: $P(A|Q)$. Asterisks* denote the score is statistically significantly better than the scores of all baseline models.

7.3.3 Experiments on Naver collection

For this experiment, we used Naver test collection A described in section 3.3.1. The test collection consists of 8 million Q&A pairs, 50 questions and a set of relevance judgments (815 relevant FAQs). Word relationships learned from Naver collection A were used.

Table 7.5 is the summary of the retrieval results. The reported results are the best scores that the models achieve after an exhaustive search on their parameter spaces. In MAP, TransLM outperforms all other models and translation-based retrieval models are better than baseline models. In P@20, all three translation-based models show similar performance. In this collection, relevance model shows poor performance in both measures.

The best performance of TransLM is achieved again when the mixing parameter is 0.7. Therefore, in all collections, TransLM shows the best performance when the mixing parameter is 0.7. This result indicates that TransLM is stable with respect to the mixing parameter. In all the following experiments, we used 0.7 as a default value for the mixing parameter instead of tuning it.

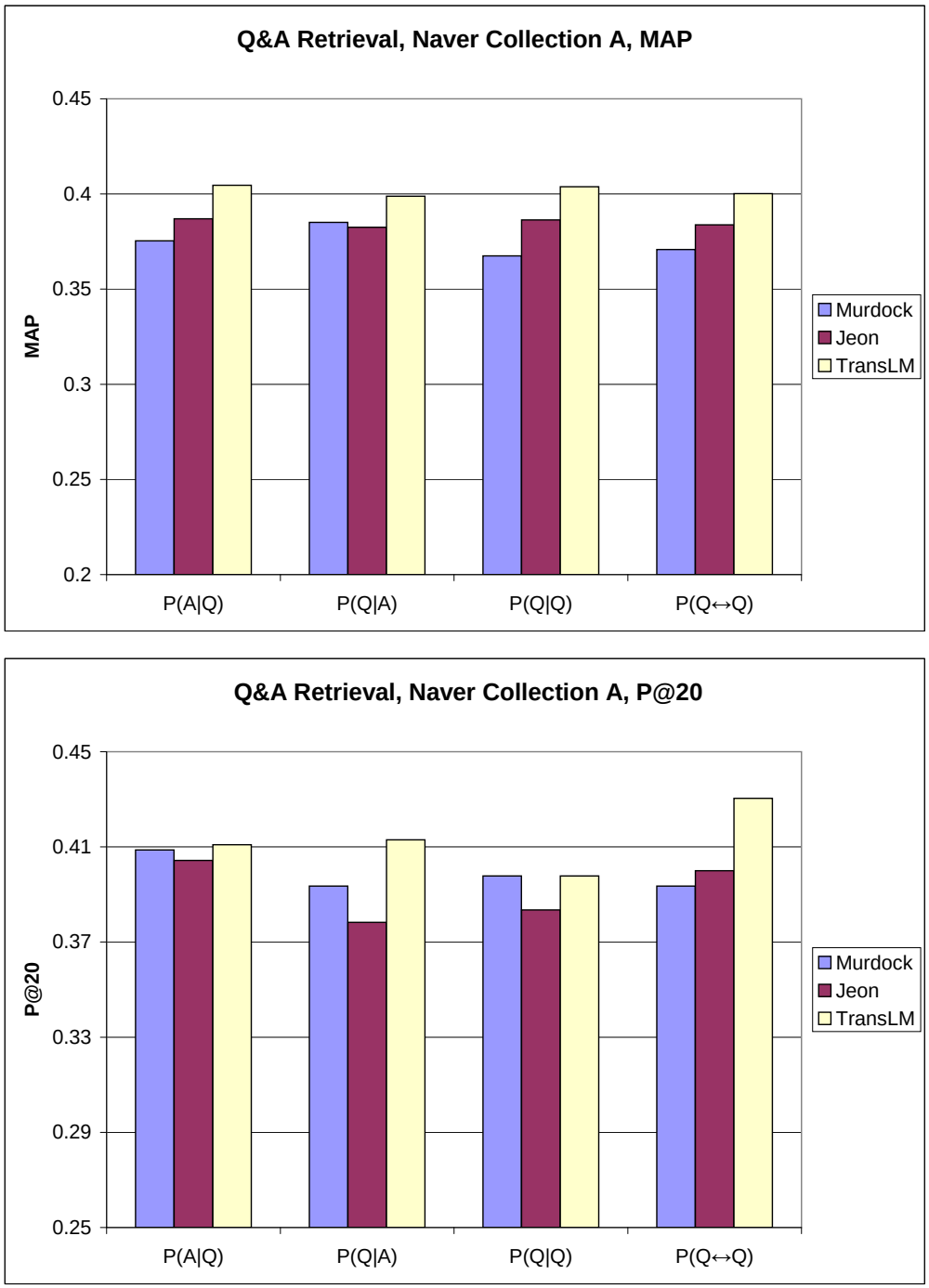


Figure 7.6. Comparison of Translation Tables. Naver Retrieval Task. Naver collection. The X axis is the mixing parameter β in equation 4.12. The upper figure shows results evaluated by MAP. The lower figure presents the results measured by P@20.

Baseline Retrieval Models				
Model	TTable	MAP	P at 10	P at 20
LM		0.3488	0.4304	0.3652
Okapi		0.3533	0.4391	0.3674
RM		0.2719	0.3652	0.2717

Translation-based Retrieval Models				
Model	TTable	MAP	P at 10	P at 20
Murdock	$P(A\|Q)$	0.3754	0.4478	0.4087*
Jeon	$P(A\|Q)$	0.387	0.4261	0.4043*
TransLM	$P(A\|Q)$	**0.4046***	0.4609	0.4109*
Murdock	$P(Q\|A)$	0.3851*	0.4565	0.3935
Jeon	$P(Q\|A)$	0.3825*	0.4348	0.3783
TransLM	$P(Q\|A)$	0.3988*	**0.4739***	0.4130*
Murdock	$P(Q\|Q)$	0.3675	0.4391	0.3978
Jeon	$P(Q\|Q)$	0.3864*	0.4174	0.3935
TransLM	$P(Q\|Q)$	0.4038*	0.4435	0.3978
Murdock	$P(Q \leftrightarrow Q)$	0.3708	0.4348	0.3935
Jeon	$P(Q \leftrightarrow Q)$	0.3838*	0.4435	0.4000*
TransLM	$P(Q \leftrightarrow Q)$	0.4002*	0.4652	**0.4304***

Table 7.6. Summary of effectiveness of retrieval models on the Naver collection A. Summary of Question Retrieval Results. Naver collection. Asterisks* denote significant improvement over all baseline models.

Figure 7.6 provides a comparison of the translation-based retrieval models depending on various types of word relationships. TransLM is better than the other models in all measures with all types of word relationships. In this collection, the performance gap between different types of word relationships is relatively small compared to other collections. All types of word relationships show similar performance in MAP.

One interesting point is that we get the best P@20 score with $P(Q \leftrightarrow Q)$ instead of $P(A|Q)$. This demonstrates the potential of the proposed approach to automatically construct training samples using similarities between answers. The Naver collection seems to have more duplicated answers than other collections because of the bigger collection size. Therefore, better training samples could be extracted.

Table 7.6 summarizes all experimental results presented in this subsection.

7.3.4 Category Specific Word Translation

In the Naver Q&A service, users manually designate categories of their questions. Therefore, Q&A pairs under the same category can be clustered into a group. If we learn word relationships using Q&A pairs in a specific cluster, we can make a category specific word-to-word translation table. These category specific word relationships have potential to disambiguate terms with multiple meanings and improve retrieval performance. In this section, we test whether category specific word translations can improve retrieval performance. In subsection 5.4.1, we show examples of category specific word-to-word translations.

For the experiment, we used the Naver test collection **A** because category information is available for this collection. Every Q&A pair in the collection is assigned to one of 11 categories. For each category, a category specific word translation table was constructed.

We performed exactly the same experiment described in subsection 7.3.3. The only difference is that category specific word relationships are used instead of global word relationships learned from whole collection. For example, when we score questions in a category, say 'Computer' category, we use word relationships learned from the training samples in the corresponding category.

Figure 7.7 presents experimental results. 'Global' in the graphs denotes the use of using word relationships learned from all Q&A pairs in the Naver collection. All translation-based retrieval models achieve better results with category specific translations in all measures. In many cases, the improvement is statistically significant. Table 7.7 is the summary of the retrieval results. Experimental results show that Category specific translations deliver more accurate context information into the model and improve the retrieval performance further.

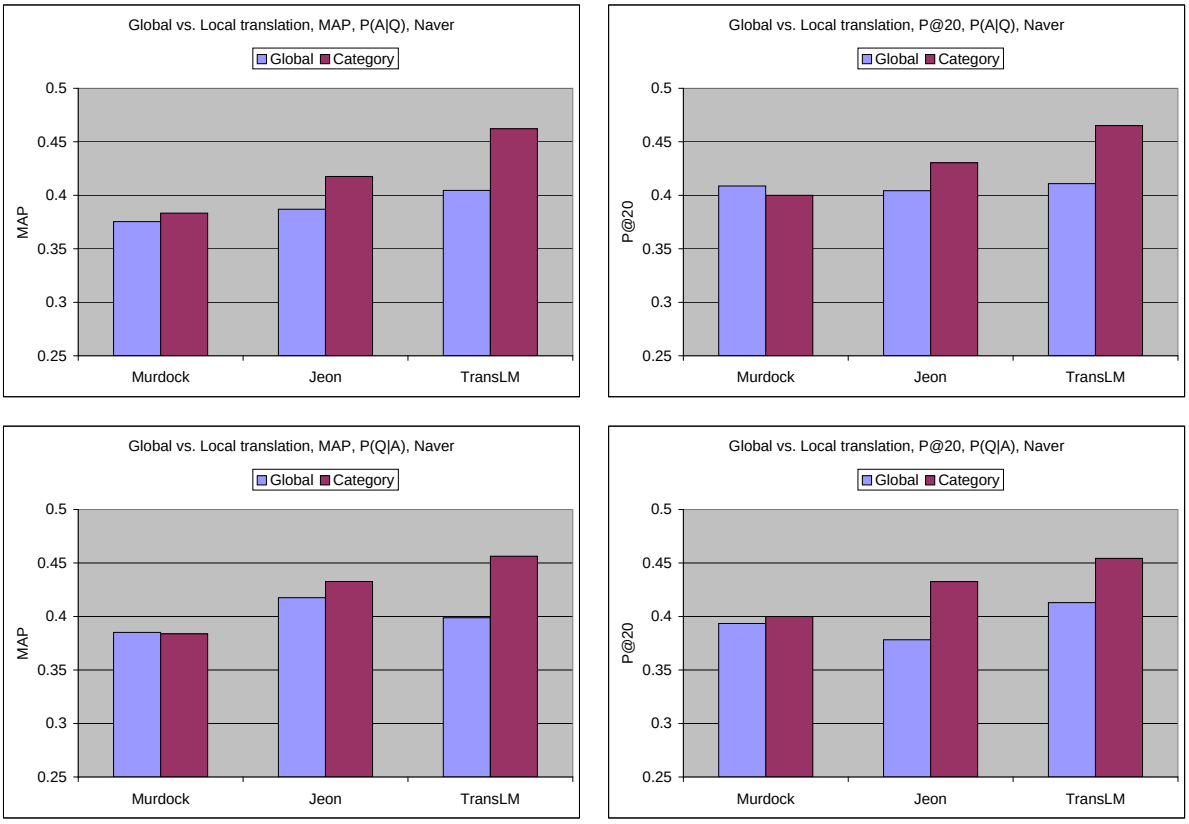


Figure 7.7. Retrieval Results with category specific word relationships. 'Global' in the graphs denotes the results are using word relationships learned from all Q&A pairs in the Naver collection. Graphs on the left show results evaluated by MAP and the graphs on the right show results measured by P@20. Upper graphs use $P(A|Q)$ and lower graphs use $P(Q|A)$. The category specific word translations boost retrieval performance in all cases.

Baseline Retrieval Models					
Model	TTable	Context	MAP	P at 10	P at 20
LM			0.3488	0.4304	0.3652
Okapi			0.3533	0.4391	0.3674
RM			0.2719	0.3652	0.2717

Global Translation					
Model	TTable	Context	MAP	P at 10	P at 20
Murdock	$P(A\|Q)$	global	0.3754	0.4478	0.4087
Jeon	$P(A\|Q)$	global	0.387	0.4261	0.4043
TransLM	$P(A\|Q)$	global	0.4046	0.4609	0.4109
Murdock	$P(Q\|A)$	global	0.3851	0.4565	0.3935
Jeon	$P(Q\|A)$	global	0.3825	0.4348	0.3783
TransLM	$P(Q\|A)$	global	0.3988	0.4739	0.4130

Category Specific Translation					
Model	TTable	Context	MAP	P at 10	P at 20
Murdock	$P(A\|Q)$	category	0.3833	0.4391	0.4000
Jeon	$P(A\|Q)$	category	0.4176	0.4609	0.4304
TransLM	$P(A\|Q)$	category	**0.4622***	**0.4870**	**0.4652***
Murdock	$P(Q\|A)$	category	0.3838	0.4435	0.4000
Jeon	$P(Q\|A)$	category	0.4327*	0.4609	0.4326*
TransLM	$P(Q\|A)$	category	0.4564*	0.4870	0.4543*

Table 7.7. Global vs. Category Specific word relationships. The top table shows baseline retrieval performance. The middle table presents the performance using global translation table. The bottom table is the performance using category specific translations. Asterisks (*) in the bottom table denote the score is statistically significantly better than the corresponding score in the middle table.

7.3.5 Experiments with Short Queries

In this subsection, we describe experiments conducted to evaluate our system with short keyword queries. We used Naver test collection B which was used to develop the quality prediction system in chapter 6. The test collection consists of 6.8 million Q&A pairs, 125 keyword queries and a set of relevance judgments (1700 relevant Q&A pairs). We calculated word relationships of type $P(A|Q)$ from Naver collection B and used them for TransLM. In this experiment, we tune only one parameter (Dirichlet smoothing parameter) and set the mixing parameter to 0.7 as this parameter setting achieved good performance in previous experiments.

Retrieval results are presented in Table 7.8. The query likelihood language model and Okapi BM25 show almost identical performance and the relevance model results in the worst performance. The relevance model consistently works poorly with the Naver collection. If we do fine-tune all other parameters in the relevance model, we may get little improved performance. TransLM is better than all baseline models in all measures. The improvements are statistically significant. Experimental results show our approach works well with both long and short queries.

7.3.6 Integrating Quality Scores

In this section, we integrate the quality scores calculated in chapter 6 into our retrieval model. We used Naver test collection B because the quality scores were calculated only for this collection. In the previous subsection, we show the retrieval

	MAP	P10	P20
LM	0.294	0.366	0.298
Okapi	0.296	0.370	0.295
RM	0.259	0.316	0.254
TransLM, $P(A\|Q)$	**0.327***	**0.436***	**0.373***

Table 7.8. Retrieval results with keyword Queries. Naver collection. Asterisks* denote the model is statistically significantly better than all baseline models.

performance can be significantly improved using TransLM. Now we integrate the quality score into TransLM. TransLM is based on language modeling framework and it contains the document prior component ($P(D)$ in equation 4.9). Instead of assuming constant document priors, we use the quality scores as document priors.

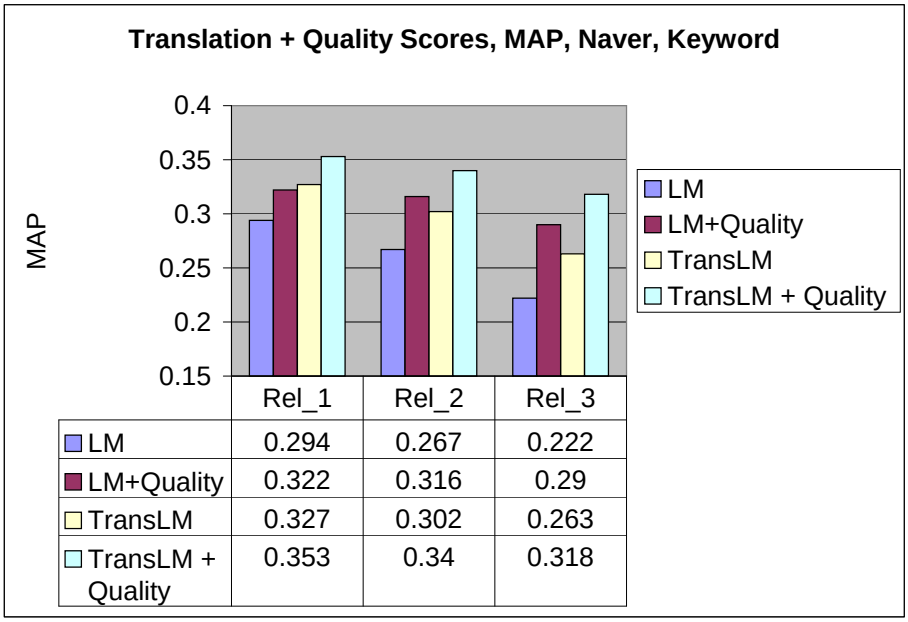
We use the same relevance judgment files used in chapter 6. The judgement file 'Rel 1' takes into account only relevance. 'Rel 2' and 'Rel 3' consider both relevance and quality together. 'Rel 3' requires stricter conditions than 'Rel 2' to be a good quality Q&A pair. A detailed explanation about the relevance judgment is in chapter 6.

Figure 7.8 shows experimental results. As shown in chapter 6, incorporating the quality scores can boost the retrieval performance of the query likelihood language model. The graphs in the figure show that similar improvement can be achieved with TransLM as well. Considering TransLM is already better than the query likelihood language model, further improvement is impressive.

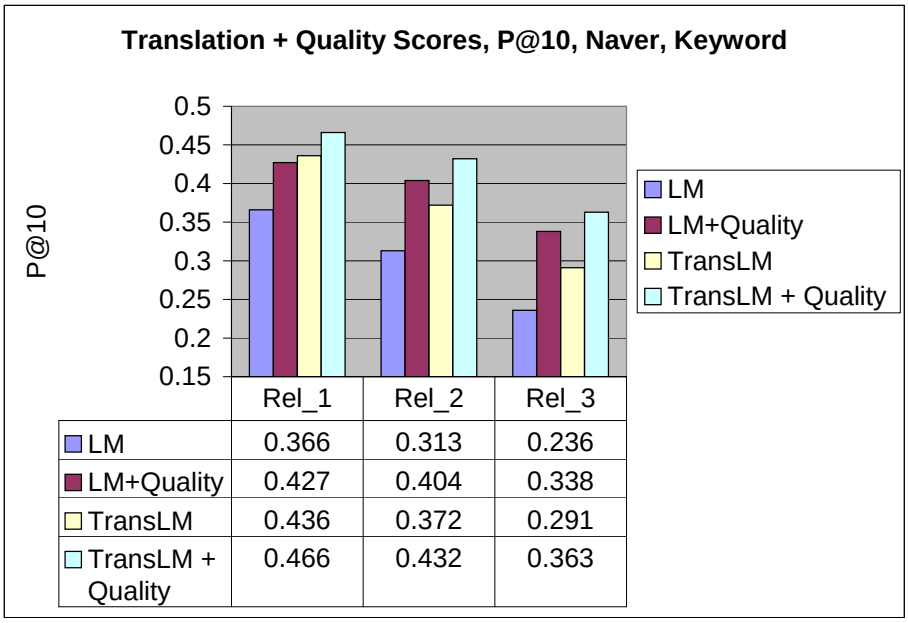
With regular relevance judgment file (Rel 1), TransLM and the quality measure improve the retrieval performance almost by equal amounts. The advantage of using the quality scores becomes bigger when we require stricter conditions for a good quality Q&A pair. TransLM consistently adds more improvement on top of the improved results regardless of the relevance judgement files. This means that TransLM and the quality scores improve the retrieval performance in different ways and they can be combined to provide additional improvements.

7.3.7 Retrieval Examples

Table 7.9 shows how our approach overcomes the word mismatch problem. In the first example, the question has "mooon" instead of "moon", but TransLM can retrieve the question in the top 10 ranks because our model knows "moon" and "mooon" are related. One thing to notice here is that P(mooon,A|moon,Q) is much smaller than



	Rel_1	Rel_2	Rel_3
LM	0.294	0.267	0.222
LM+Quality	0.322	0.316	0.29
TransLM	0.327	0.302	0.263
TransLM + Quality	0.353	0.34	0.318



	Rel_1	Rel_2	Rel_3
LM	0.366	0.313	0.236
LM+Quality	0.427	0.404	0.338
TransLM	0.436	0.372	0.291
TransLM + Quality	0.466	0.432	0.363

Figure 7.8. Integrating quality scores into TransLM. The upper table shows mean average precisions and the lower table shows precisions at rank 10.

Query	how far away is the moon?
Question	how far is it to the mooon?
Analysis	P(moon,A | mooon,Q) = 0.605

Query	Who invented television?
Question	by who was invent the first tv? who invent the telavision? who invent the televsion? when was the tv imbente?
Analysis	P(television,A | tv,Q) = 0.029 P(television,A |telavision,Q) = 0.271 P(television,A | televsion,Q) = 0.350 P(invent,A| imbente,Q) = 0.230

Query	what is primary language of the Philippines?
Question	what languge do the phillipinoe speak?
Analysis	P(philippines,A | phillipinoe,Q) = 0.231

Table 7.9. Analysis of the retrieval results. All the questions are retrieved in top 10 by TransLM.

P(moon,A|mooon,Q). This means we have high chance of seeing "moon" in the answer when we see "mooon" in the question but the chance of observing "mooon" in the answer when we see "moon" in the question is very low.

In the second example, questions from the Wondir collection have "tv", "telavision" and "televsion" to denote "television". Our approach successfully retrieves these questions in the top 10 ranks. The third example shows that our model exploits the word relationship between "phillipinoe" and "philippines" to successfully find the semantically identical question. None of the questions in the examples can be retrieved using conventional retrieval algorithms. These examples show our method can handle mis-spelled words and synonyms.

Table 7.10 compares the top 10 questions retrieved by the query likelihood language model and TransLM for a sample query. Overall, TransLM returns many topically related questions to the query although not all of them are semantically

identical to the original query. Our translation-based approach successfully relates 'Francis Scott Key' with 'Star Spangled Banner' or 'national anthem' because of good word-to-word translations. The following are few other example questions that are retrieved only by translation-based retrieval models in the top 10 ranks.

Query: what is a caldera?
Question: what is the open at the top of a volcano call?

Query: what did Vasco da Gama discover?
Question: why was portugal able to take an early lead in the exploration of the indian ocean?

Query: What was the name of the famous battle in 1836 between Texas and Mexico?
Question: how did the battle of the alamo start?

The above examples show our approach can connect semantically related questions even thought there is no word overlap between them. The knowledge in Q&A collections seem to be encoded into the word translation relationships and it is reused in the retrieval procedure to capture the semantic relationship between the query and the document.

7.4 Experiments on Other IR Tasks

In this section, we explore the utility of word relationships learned from our Q&A collections for other information retrieval tasks. We consider two IR tasks: a)answer passage retrieval, b)finding relevant news articles for a given title query. For the second task, we used 2005 Robust Track data. Experimental results show the potential of our approach as a general-purpose information retrieval framework.

Query: What is Francis Scott Key best known for? (Answer: The Star Spangled Banner)

Query Likelihood Language Model

Rank	Retrieved Questions
1	did Francis Scott Key rename "Anacrean to Heaven" to "The Star Spangled Banner"?
2	What is Magic Johnson best known for?
3	who is Boniface and what is he best known for
4	country best known for it's neutrality in world affars?
5	What was Albert Einstein best known for?
6	What document is James Monroe best known for and what does this document say?
7	what is francis ford coppolas proffesion
8	What is Francis Ford Coppola's profession
9	He is best known for his work Gitagovinda?
10	What is Edwin Land's best known invention?

TransLM

Rank	Retrieved Questions
1	did Francis Scott Key rename "Anacrean to Heaven" to "The Star Spangled Banner"?
2	what is ireland national athem?
3	What is the best herbicide for clover?
4	what is a banner?
5	what is the national athem of the us
6	who wrote THE STAR SPANGLE BANNER and what year?
7	What is Magic Johnson best known for?
8	original name before The Star Spangled Banner?
9	who is Boniface and what is he best known for
10	In "Star-Spangled Banner" over what does the flag wave

Table 7.10. Question Retrieval Examples. Wondir Collection.

Wondir Collection

Model	TTable	MAP	P at 10	P at 20
LM		0.1201	0.1303	0.0955
Okapi		0.1151	0.1242	0.0894
RM		0.1356	0.1333	0.1152
TransLM	$P(A\|Q)$	0.1242	0.1273	0.1030
TransLM	$P(Q\|A)$	**0.2315***	**0.2000***	**0.1576***

WebFAQ Collection

Model	TTable	MAP	P at 10	P at 20
LM		0.0671	0.0480	0.0340
Okapi		0.0661	0.0460	0.0330
RM		0.0646	0.0460	0.0390
TransLM	$P(A\|Q)$	0.0884*	0.0600*	0.0470*
TransLM	$P(Q\|A)$	**0.090***	**0.0640***	**0.0490***

Table 7.11. Answer Passage Retrieval Results. The upper table shows retrieval results on the Wondir collection and the lower table shows retrieval results on the WebFAQ collection. Asterisks (*) next to scores denote the score is statistically significantly better than all baseline models.

7.4.1 Answer Passage Retrieval

Answer passage retrieval is typically used as the first step in automated question answering systems. Retrieved answer passages are further processed to extract exact answers. Therefore, the effectiveness of QA systems heavily depends on the initial retrieval.

To simulate answer passage retrieval tasks, we removed all questions from the Q&A collections and built collections of answers. We used the same query set that was used for the Q&A retrieval tasks in the previous chapter. We assume that if the question part is semantically identical to the query then the corresponding answer must be relevant to the query. This allow us to reuse the relevance judgement files built for the question retrieval tasks. The Wondir and the WebFAQ collections were used for this experiment.

Experimental results presented in Table 7.11 show that TransLM significantly outperforms baseline retrieval models on both collections in all evaluation metrics. Reported scores are the best performance that each model can achieve after tuning their parameters. The query likelihood model has only one parameter and all other models have two parameters to be tuned.

One interesting point is that we get the best performance with $P(Q|A)$ instead of $P(A|Q)$ on both test collections in direct opposite to the result that we observed in Q&A retrieval experiments. $P(Q|A)$ is aimed to simulate the task of sampling questions from answers and this corresponds to the language modeling approach for answer passage retrieval.

Surprisingly, our model can retrieve correct answers even when the answers have no word overlap with the query. The following are a few example answers retrieved in the top 10 ranks for the given query.

Query: Who was Whitcomb Judson? (A: Inventor of Zipper)
Answer: the guy who invent the modern zipper

Query: Name a flying mammal? (A: Bat)
Answer: bumblee bat

Query: What does laser stand for?
Answer: light amplify by stimulate emission of radiate

Such retrieval is possible when similar questions to the query have been previously asked and answered in the Q&A collection. The model seems to capture some knowledge about the topic in the form of word translation relationships and exploits this knowledge to find relevant answers. This suggests that if we have enough question

and answer pairs for a given domain, we can build a high quality answer passage retrieval system for that domain.

7.4.2 Robust Track Experiments

To demonstrate the potential of our approach as a general purpose IR framework we test our model on 2005 Robust track data. The Robust track [70] uses the ACQUAINT corpus that consists of about one million news articles. We used 50 title queries and the relevance judgement file provided by NIST[2]. Since the Robust track focuses on poorly performing topics, our expectation is that our approach may be useful for these hard queries that conventional retrieval algorithms cannot handle well.

Table 7.12 shows our approach outperforms state of the art retrieval models and achieves statistically significant improvements. The Wondir and the WebFAQ collection are very different from the news collection in many aspects such as topics, length and writing quality. However, the results show that the word relationships learned from Q&A collections can be used for other collections.

Better performance than ours was previously reported [18]. They used external news corpora to expand original queries terms. We believe that if we can get large collections of Q&A pairs that discuss news topics, then we will be able to further improve our retrieval performance on this collection.

Zinxi et al. [75] showed that combining multiple translation tables built from multiple sources can improve the performance of cross lingual information retrieval systems. We linearly combined translation tables generated from the Wondir and the WebFAQ collections: $P_{comb}(w, Q|t, A) = 0.5P_{wondir}(w, Q|t, A) + 0.5P_{wondir}(w, Q|t, A)$. The results show that some improvement can be obtained from the combination.

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

Model	TTable	MAP	P at 10	P at 20
LM		0.2076	0.4460	0.4050
Okapi		0.2049	0.4620	0.4020
TransLM	Wondir	0.2272*	**0.4760***	0.4200*
TransLM	WebFAQ	0.2299*	0.4740*	0.4220*
TransLM	Comb	**0.2334***	0.4600	**0.4280***

Table 7.12. Experimental Results on 2005 Robust Track Data.

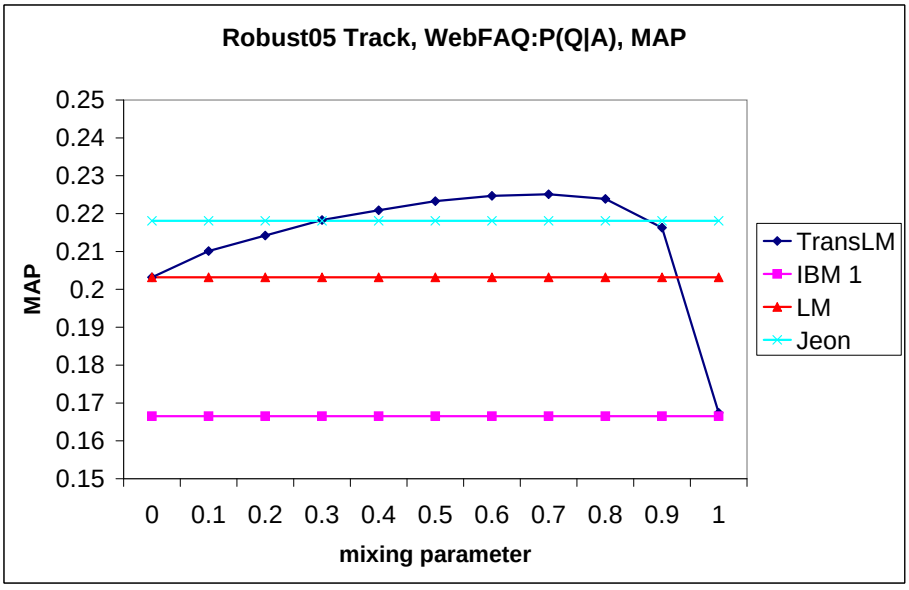


Figure 7.9. Comparison of Retrieval Models. 2005 Robust Track Data.

7.5 Summary

In this chapter, we highlighted the utility of our approach with multiple real world collections and diverse set of queries: short, long, English, Korean, queries submitted to web search engines and questions submitted to Q&A services. Experimental results show that the proposed approach consistently and significantly outperforms state of the art retrieval models for all experiments. TransLM also outperforms other translation-based retrieval models when we use the same word relationships. Table 7.13 is the list of experiments that we described in this chapter.

Retrieval examples presented in section 7.3.7 show that TransLM can successfully address the word mismatch problem and has the ability to capture semantic relationships between questions.

We also successfully and seamlessly incorporated the quality measure into our retrieval model to find relevant and high quality Q&A pairs for a given query. The integration improved the retrieval performance even further.

We learn word-to-word translation probabilities from the target Q&A collection for the retrieval task. This strategy is successful because we can build collection specific word relationships that accurately reflect unique interests and word usages of the collection. Category specific translations bring more context information into the model and improve the retrieval performance further.

Finally, we evaluated our system with two other information retrieval tasks: answer passage retrieval and adhoc retrieval. Experimental results showed the promising potential of our approach as general information retrieval framework.

#Sec	Task	Collection	Language	Query	Etc.
7.3.1	Q&A Retrieval	Wondir	English	Question	Global
7.3.2	Q&A Retrieval	WebFAQ	English	Question	Global
7.3.3	Q&A Retrieval	Naver A	Korean	Question	Global
7.3.4	Q&A Retrieval	Naver A	Korean	Question	Category
7.3.5	Q&A Retrieval	Naver B	Korean	Keywords	Global
7.3.6	Q&A Retrieval	Naver B	Korean	Keywords	Global, Quality
7.4.1	Answer Retrieval	Wondir	English	Question	Global
7.4.1	Answer Retrieval	WebFAQ	English	Question	Global
7.4.2	Adhoc Retrieval	Robust05	English	Keywords	Global

Table 7.13. Summary of Experimental Settings. The first column is the section number that the corresponding experiment is described.

CHAPTER 8

RELATED WORK

Research related to our work spans many areas such as question answering, FAQ retrieval and statistical machine translation. In this chapter, we aim to place our work in context of previous research in related areas.

- **Translation-based Information Retrieval**

  Since the introduction of the idea of using machine translation techniques for information retrieval by Berger and Lafferty [6], many researchers have applied the idea for diverse applications such as summarization, sentence retrieval, FAQ retrieval and semantic smoothing [3, 48, 5, 77].

  There are two main issues in applying translation methods. The first one is that the pure translation model is not suitable for information retrieval and proper modification is needed. In chapter 4, we reviewed some previous attempts [25, 28, 48] to address the problem and propose a new solution.

  The second issue is that estimating high quality word translation probabilities. Berger and Lafferty artificially generated queries from documents using mutual information and used the query-document pairs to learn word relationships. However, mutual information does not generate high quality queries and subsequent research [48] showed poor performance using this approach.

  Cao et al. [14] used co-occurrence statistics of two terms in a given window size to measure similarities between terms. However, synonyms usually do not occur together and therefore, this approach may not effectively handle the word mismatch problem caused by synonyms.

Jin et al. [28] used title-document pairs to train their title language models but the vocabulary for titles is much smaller than the vocabulary for documents. Because of the unbalanced vocabulary sizes, target terms may not converge to the correct source terms. If the vocabulary for queries is much bigger than the vocabulary for the titles, then only a limited number of the queries can benefit from the approach. Our Q&A collections contain millions of different words both in questions and answers and avoid the imbalance problem caused by different vocabulary sizes.

Other approaches include using dictionaries and thesauri but limited coverage, biased term distributions and outdated contents make them less useful. Murdock and Croft [48] used English-Arabic and Arabic-English dictionaries to automatically generate a probabilistic English-English thesaurus but could not produce statistically significant improvement. Berger et al. [5] used pure IBM model 1 to find the relevant answers among multiple candidate answers for call center users. Their experiments were done with small data sets that consisted of only a few thousand Q&A pairs. Our system is built using millions of Q&A pairs that encompass broad topics. We showed our system could handle general questions sampled from real user queries.

- **FAQ Retrieval**

  The most similar work to ours has been done in FAQ retrieval research. FAQ Finder [13] heuristically combines statistical similarities and semantic similarities between questions to rank FAQs. Conventional vector space models are used to calculate the statistical similarity while WordNet [19] is used to estimate semantic similarity. FAQ Finder has been modified by adding other components such as a question type classifier [44] and a word sense disambiguator [43]. Sneiders [61] manually attached four types of annotations (Required keywords, Optional keywords, Irrelevant words, Forbidden words) to FAQs and

proposed a heuristic-based search technique. Lens et al. [39] applied case-based reasoning techniques for domain specific FAQ retrieval. They manually defined domain specific keywords and attributes to build cases. Wu et al. [73] proposed domain specific FAQ retrieval techniques based on aspect models. They used domain dependent ontology to represent semantic representations of the aspects. In their approach, queries and FAQs were interpreted as mixtures of independent aspects. Kim and Seo [32] clustered query logs and smoothed a FAQ using the closest cluster. All these previous approaches were tested on relatively small sized FAQ collections. They were also hard to scale because they are based on specific knowledge databases or handcrafted rules. More recently, Jijkoun and Rijke [27] automatically collected a few million FAQs from the web and implemented a search system for their FAQ collection. Their search system used traditional document retrieval algorithms.

- **Question Answering**

  While extensive research has been done in the field of question answering [69, 54, 46], our work is different from traditional question answering. In question answering, short answers for a relatively limited class of question types are automatically extracted from document collections. In Q&A retrieval, answers for an unlimited range of questions are retrieved by focusing on finding semantically similar questions in the archive.

  Some of the research related to our work includes the approach that uses FAQ collections as training data to discover relationships between user questions and candidate answer passages. Ramakrishnan [56] et al. automatically predicted answer types and keywords for a given question using FAQs as training data.

  FAQ collections have been used to train question-answering systems. Soricut and Brill [62] used one million FAQs collected from the web to train their

answer passage retrieval system. They used the original IBM model 1 without any modification. Interestingly they calculated the probability of generating answers from questions to select answer passages which is exactly opposite to our approach. In our experiments, we got better performance following the generative language modeling framework that forces us to sample questions from answers for this task. Agichtein at al. [1] used FAQ collections to learn relationships between question types and words in answers. They transformed a user question by adding words that are related to the type of the question. They could handle only four types (how, what, where, who) of simple questions.

- **Document Quality Estimation**

  Many factors decide the quality of documents (or answers). Strong et al. [65] listed 15 factors and classified those factors into 4 categories: contextual, intrinsic, representational and accessibility. Zhu and Gauch [79] came up with 6 factors to define the quality of web pages. However, so far, there is no standard metric to measure and represent the quality of documents.

  As far as we know, there has been almost no research to estimate the quality of answers in FAQ or Q&A collections. However, there has been extensive research to estimate the quality of web pages. Most of the work [9, 33] is based on link analysis. Zhu and Gauch [79] studied various content-based metrics to predict the quality of web documents. Zhou and Croft [78] proposed a document quality model that uses content based features such as the information-noise ratio and the distance between the document language model and the collection model.

  The language modeling framework provides a natural way for combining prior knowledge in the form of prior probability. Prior information such as time, quality and popularity have been successfully integrated using the prior prob-

ability [35, 78, 41]. We also incorporated our quality scores into the retrieval framework as document priors.

Berger et al. [7] proposed the use of the maximum entropy approach for various natural language processing tasks in mid 1990's and after that many researchers have applied this method successfully to a number of other tasks including text classification [49, 53] and image annotation [26]. We used this approach to build our answer quality predictor.

- **Query Clustering**

  We found semantically similar questions using answers. The idea of finding similar queries using click logs or retrieval results has been proposed previously [72, 66, 4]. They assumed that if two different queries have similar click logs or similar retrieval results, then the queries are semantically similar, and the query similarities obtained using this approach would be superior to comparing the text of the queries directly. We also make a similar assumption that if two answers are similar enough then the corresponding questions should be semantically similar.

- **Paraphrase Generation**

  In this thesis, we described a method to acquire semantically similar questions using the similarity between answers. These question pairs can be thought of as paraphrases. There have been a number of studies to generate or find paraphrases in the area of natural language processing. Lepage and Denoual [40] generated paraphrases of a given sentence using previously detected paraphrases. Shinyama and Sekine [60] automatically find paraphrases in Japanese news articles by finding overlap of certain kinds of noun phrases such as names, dates and numbers.

- **Query Expansion**

  In this thesis, we addressed the lexical chasm problem between questions. Various query expansion techniques have been studied to solve word mismatch problems between queries and documents, including relevance feedback [58], thesaurus-based expansion (e.g. [67] [20], [59]), dimensionality reduction (e.g. [17], [23]), and techniques based on modifying the query based on the top retrieved documents (e.g. [74], [38]). The model proposed here implicitly expands queries using translation probabilities. We generate these translation probabilities from Q&A collections. These translation probabilities are then used in a retrieval model to rank the questions in the Q&A collections for a new user-generated question.

CHAPTER 9

CONCLUSION AND FUTURE DIRECTIONS

9.1 Conclusion

In this thesis, we proposed a new type of information system that behaves like an intelligent question answering system. This system searches collections of previously answered questions instead of using sophisticated and expensive semantic and contextual analysis to generate answers.

To this end, we defined a new information retrieval task, namely Q&A retrieval. The goal of Q&A retrieval is finding answers but the actual task is finding questions that are semantically identical to the user question. In this task, the word mismatch problem becomes very serious because of the short lengths of questions. To solve this problem, we designed a translation-based retrieval model that combines advantages of both machine translation and information retrieval. We showed that this model was successful for the task of Q&A retrieval and it has good potential as general purpose IR framework.

The success of translation-based retrieval models depends on the quality of word translation. We built good quality word-to-word translation tables from Q&A collections. A collection specific word translation table captures various interesting properties of the collection such as topical interests and word usages. In our experiments with category specific word translation, we could integrate more context information into the translation table. Retrieval examples showed our retrieval model with collection and category specific word translations can reduce the word mismatch problem and significantly improves retrieval performance.

Another important challenge is measuring the quality of answers to return high quality answers to users. In this thesis, we showed how to systematically use various non-textual features to predict the quality of answers in community-based question answering services. The proposed method is general enough to be applied to many other web-based information retrieval services. The advantage of our approach is that the quality score is given as a probability value and it can be easily integrated into other statistical models. We successfully combined this quality measure into our retrieval system.

In this thesis, we demonstrated how to approach a new information retrieval problem using recent advances in related fields like information retrieval, machine learning and natural language processing. All experiments in this work are based on real world settings including collections and queries. Since we relied on only statistical approaches and avoided using any collection and/or task dependent heuristics, our methodology can be easily applied to solve other information retrieval problems.

One another contribution of this work is that it provide insights on the real power of the language modeling framework in real world environments. Most commercial information retrieval systems rely on heuristic based information retrieval models and the language modeling framework has been rarely used. We showed the translation model and the quality predictor could be seamlessly combined under the language modeling framework to build a powerful and practical retrieval system. The flexibility and adaptability came from the solid mathematical background of the language modeling framework.

9.2 Directions for Future Research

Every component in our system can be improved in multiple ways. In this section, we briefly discuss the directions that we think are most promising.

- **Phrase-based Translation**

  Phrase-based machine translation [34, 15, 51] has shown superior performance compared to word-to-word translation models. Integrating phrase to phrase and phrase to word relationships into our model may further improve retrieval performance.

- **Combining Word Relationships**

  Combining multiple word relationships learned from diverse sources may be useful. If we smooth category or topic specific word relationships with the word relationships acquired from the whole collection, then more accurate translation-based language models may be estimated leading to improved performance.

- **Combining Multiple Fields**

  In this work, we focused on the similarity measure between the query and the question part of Q&A pairs because the question part is much more important than the answer part in our task. However, the answer part has good potential to improve the accuracy of our retrieval system. In the preliminary experiments not described in this thesis, we could verify careful combinations of both parts can improve retrieval performance for short keyword queries. In the case of long queries, because of the big performance gap between two parts, we could not see the advantages of combinations. Actually, careless combinations can hurt retrieval performance. However, we believe there may be a combination method that can consistently improve performance regardless of query types.

- **Category Smoothing**

  A great deal of effort has been devoted toward improving retrieval performance and efficiency using clusters. Previous work [22, 66] have claimed that better retrieval performance can be achieved if high quality clusters can be obtained. Most Q&A services request users to select a category when they post questions. Therefore, almost all questions have natural clusters. Since we have access to almost perfect clusters, we expect improved retrieval performance using these clusters. One possible way [42] is smoothing document language models with cluster language models before smoothing them with background collections.

- **Paraphrase Finding Using Near Duplicates Detection**

  In this thesis, we proposed a rank based similarity measure between answers to find semantically related questions pairs. This measure is expensive because all pair-wise ranks must be calculated. We found near duplicates detections algorithms [16, 8] can do similar work more efficiently. Many people copy other users' answers to answer similar questions. If we can find these near duplicated answers, clusters of semantically related questions can be easily built. These clusters can be further used for diverse purposes.

- **Quality Estimation using Non-Textual Features**

  Our annotators who manually judged the quality of answers reported that word usages also have close connections to the answer quality. However, our quality predictor exploits only non-textual features. If we can modify our framework to handle both non-textual and textual features, it will be much more useful. In a preliminary experiment not presented in this thesis, we were able to reliably predict the answer quality with non-textual features with very similar machine learning framework that we proposed in this thesis.

APPENDIX A

Q&A RETRIEVAL SYSTEM ARCHITECTURE

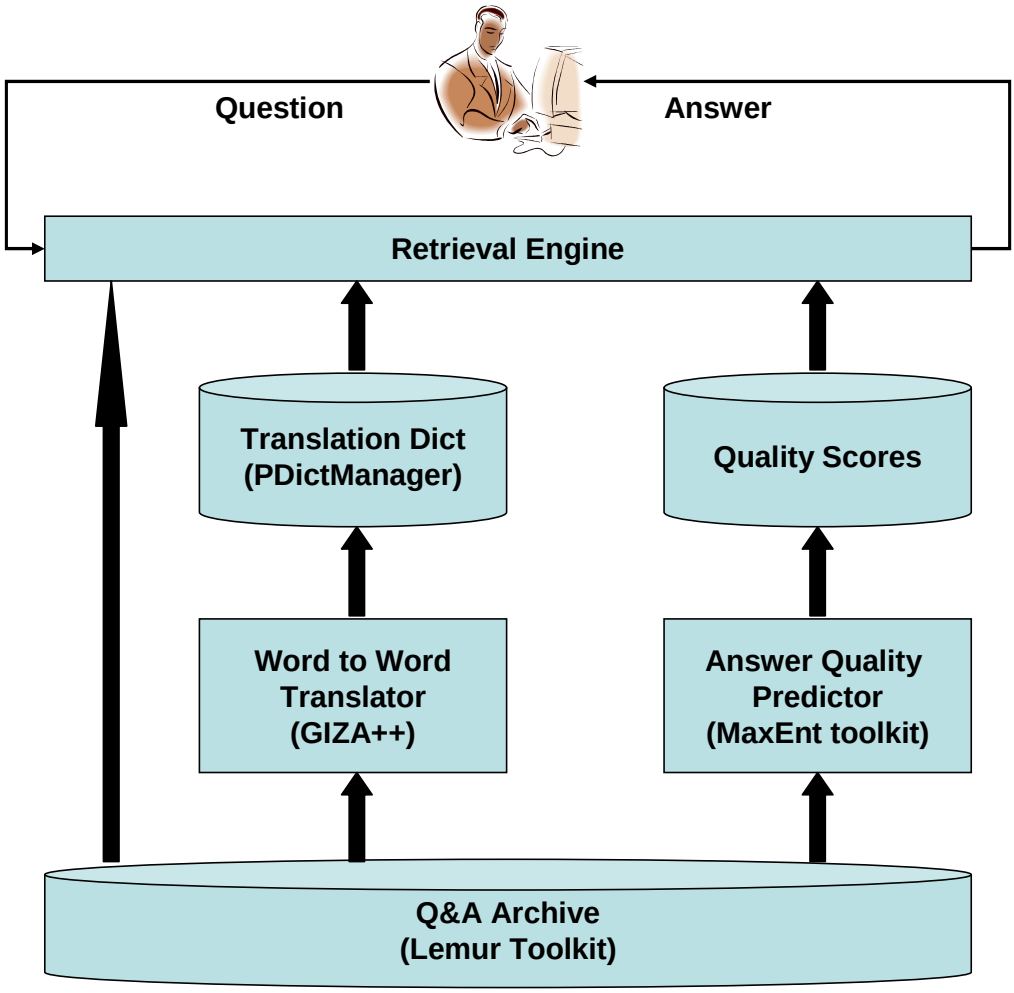


Figure A.1. Q&A Retrieval System Architecture.

APPENDIX B

PUBLICATION LIST

- Jiwoon Jeon and W. Bruce Croft. Translation-Based Language Models. submitted to the ACM Seventeenth Conference on Information and Knowledge Management (CIKM), 2007.

- Ron Bekkerman and Jiwoon Jeon. Multi-Modal Clustering for Multimedia Collection, to appear in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2007.

- Jiwoon Jeon, W. Bruce Croft, Joon Ho Lee and Soyeon Park. A Framework to Predict the Quality of Answers with Non-Textual Features. In Proceedings of the 29th International ACM SIGIR Conference, pp. 228-235, 2006.

- Soyeon Park, Joon Ho Lee and Jiwoon Jeon. Evaluation of the Documents from the Web-based Question and Answer Service. Journal of the Korean Society for Library and Information Science,40(2), pp.299-314, 2006.

- Jiwoon Jeon, W. Bruce Croft and Joon Ho Lee. Finding Similar Questions in Large Question and Answer Archives. In Proceedings of the ACM Fourteenth Conference on Information and Knowledge Management (CIKM), pp. 84-90, 2005.

- Jiwoon Jeon, W. Bruce Croft and Joon Ho Lee. Finding Semantically Similar Questions Based on Their Answers. In Proceedings of the 28th International ACM SIGIR Conference, pp. 617-618, 2005.

- Jiwoon Jeon and R. Manmatha. Using Maximum Entropy for Automatic Image Annotation. In Proceedings of the 3rd International Conference on Image and Video Retrieval, (CIVR) pp. 24-32, 2004.

- Jiwoon Jeon. Automatic Image Annotation of News Images with Large Vocabularies and Low Quality Training Data. Master Thesis, CIIR Technical Report MM-368, 2004.

- V. Lavrenko, R. Manmatha and Jiwoon Jeon. A Model for Learning the Semantics of Pictures. In Proceedings of the 17th Annual Conference on Neural Information Processing Systems, (NIPS) 2003.

- Jiwoon Jeon, V. Lavrenko and R. Manmatha. Automatic Image Annotation and Retrieval using Cross-Media Relevance Models. In Proceedings of the 26th international ACM SIGIR Conference, pp. 119-126, 2003.

BIBLIOGRAPHY

[1] Agichtein, Eugene, Lawrence, Steve, and Gravano, Luis. Learning to find answers to questions on the web. *ACM Transactions on Internet Technology 4*, 2 (2004), 129–162.

[2] Bahl, Lalit R., Jelinek, Frederick, and Mercer, Robert L. A maximum likelihood approach to continuous speech recognition. 308–319.

[3] Banko, Michele, Mittal, Vibhu O., and Witbrock, Michael J. Headline generation based on statistical translation. In *ACL '00: Proceedings of the 38th Annual Meeting on Association for Computational Linguistics* (Morristown, NJ, USA, 2000), Association for Computational Linguistics, pp. 318–325.

[4] Beeferman, Doug, and Berger, Adam. Agglomerative clustering of a search engine query log. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining* (2000), pp. 407–416.

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

[6] Berger, Adam, and Lafferty, John. Information retrieval as statistical translation. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (1999), pp. 222–229.

[7] Berger, Adam, Pietra, Stephen Della, and Pietra, Vincent Della. A maximum entropy approach to natural language processing. *Computational Linguistics 22*, 1 (1996), 39–71.

[8] Bernstein, Yaniv, and Zobel, Justin. Accurate discovery of co-derivative documents via duplicate text detection. *Inf. Syst. 31*, 7 (2006), 595–609.

[9] Brin, Sergey, and Page, Lawrence. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems 30*, 1–7 (1998), 107–117.

[10] Brown, Eric W. Fast evaluation of structured queries for information retrieval. In *SIGIR '95: Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval* (New York, NY, USA, 1995), ACM Press, pp. 30–38.

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

[12] Buckley, Chris, and Voorhees, Ellen M. Evaluating evaluation measure stability. In *SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval* (New York, NY, USA, 2000), ACM Press, pp. 33–40.

[13] Burke, Robin D., Hammond, Kristian J., Kulyukin, Vladimir A., Lytinen, Steven L., Tomuro, N., and Schoenberg, S. Question answering from frequently asked question files: Experiences with the faq finder system. *AI Magazine 18*, 2 (1997), 57–66.

[14] Cao, Guihong, Nie, Jian-Yun, and Bai, Jing. Integrating word relationships into language models. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval* (New York, NY, USA, 2005), ACM Press, pp. 298–305.

[15] Chiang, David. A hierarchical phrase-based model for statistical machine translation. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics* (Morristown, NJ, USA, 2005), Association for Computational Linguistics, pp. 263–270.

[16] Chowdhury, Abdur, Frieder, Ophir, Grossman, David, and McCabe, Mary Catherine. Collection statistics for fast duplicate document detection. *ACM Trans. Inf. Syst. 20*, 2 (2002), 171–191.

[17] Deerwester, Scott C., Dumais, Susan T., Landauer, Thomas K., Furnas, George W., and Harshman, Richard A. Indexing by latent semantic analysis. *Journal of the American Society of Information Science 41*, 6 (1990), 391–407.

[18] Diaz, Fernando, and Metzler, Donald. Improving the estimation of relevance models using large external corpora. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval* (New York, NY, USA, 2006), ACM Press, pp. 154–161.

[19] Fellbaum, Christiane. *WordNet – An electronic lexical database*. MIT Press, 1998.

[20] Grefenstette, Gregory. Use of syntactic context to produce term association lists for text retrieval. In *SIGIR '92: Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval* (New York, NY, USA, 1992), ACM Press, pp. 89–97.

[21] Harman, Donna. Overview of the first trec conference. In *SIGIR '93: Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval* (New York, NY, USA, 1993), ACM Press, pp. 36–47.

[22] Hearst, Marti A., and Pedersen, Jan O. Reexamining the cluster hypothesis: Scatter/gather on retrieval results. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (1996), pp. 76–84.

[23] Hofmann, Thomas. Probabilistic latent semantic analysis. In *Proceedings of Uncertainty in Artificial Intelligence* (1999), pp. 289–296.

[24] Hwang, J.N., Lay, S.R., and Lippman, A. Nonparametric multivariate density estimation: A comparative study. *IEEE Transactions of Signal Processing 42*, 10 (1994), 2795–2810.

[25] Jeon, Jiwoon, Croft, W. Bruce, and Lee, Joon Ho. Finding similar questions in large question and answer archives. In *Proceedings of the ACM Fourteenth Conference on Information and Knowledge Management* (2005), pp. 76–83.

[26] Jeon, Jiwoon, and Manmatha, R. Using maximum entropy for automatic image annotation. *Image and Video Retrieval Third International Conference, CIVR 2004, Proceedings Series: Lecture Notes in Computer Science 3115* (2004), 24–32.

[27] Jijkoun, Valentin, and de Rijke, Maarten. Retrieving answers from frequently asked questions pages on the web. In *Proceedings of the ACM Fourteenth Conference on Information and Knowledge Management* (2005), pp. 76–83.

[28] Jin, Rong, Hauptmann, Alex G., and Zhai, Cheng Xiang. Title language model for information retrieval. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval* (New York, NY, USA, 2002), ACM Press, pp. 42–48.

[29] Jones, K. Sparck, and Rijsbergen, C. J. Van. Report on the need for and provision of an ideal test collection. *Technical report, University Computer Laboratory* (1975).

[30] Jones, K. Sparck, and Rijsbergen, C. J. Van. Information retireval test collections. *Journal of Documentation 32*, 1 (1976), 59–72.

[31] Jones, Karen Sparck, Walker, Steve, and Robertson, Stephen E. A probabilistic model of information retrieval: development and comparative experiments - part 2. *Information Processing and Management 36*, 6 (2000), 809–840.

[32] Kim, Harksoo, and Seo, Jungyun. High-performance faq retrieval using an automatic clustering method of query logs. *Information Processing and Management 42*, 3 (2006), 650–661.

[33] Kleinberg, Jon M. Authoritative sources in a hyperlinked environment. *Journal of the ACM 46*, 5 (1999), 604–632.

[34] Koehn, Philipp, Och, Franz Josef, and Marcu, Daniel. Statistical phrase-based translation. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology* (Morristown, NJ, USA, 2003), Association for Computational Linguistics, pp. 48–54.

[35] Kraaij, Wessel, Westerveld, Thijs, and Hiemstra, Djoerd. The importance of prior probabilities for entry page search. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (2002), pp. 27–34.

[36] Krovetz, Robert. Viewing morphology as an inference process. In *SIGIR '93: Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval* (New York, NY, USA, 1993), ACM Press, pp. 191–202.

[37] Larkey, Leah S. Automatic essay grading using text categorization techniques. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (1998), pp. 90–95.

[38] Lavrenko, Victor, and Croft, W. Bruce. Relevance based language models. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (2001), pp. 120–127.

[39] Lenz, Mario, Hubner, Andre, and Kunze, Mirjam. Question answering with textual cbr. In *Proceedings of the Third International Conference on Flexible Query Answering Systems* (1998), pp. 236–247.

[40] Lepage, Yves, and Denoual, Etienne. Automatic generation of paraphrases to be used as translation references in objective evaluation measures of machine translation. In *Proceedings of the third International Workshop on Paraphrasing* (2005), pp. 57–64.

[41] Li, Xiaoyan, and Croft, W. Bruce. Time-based language models. In *Proceedings of the Twelfth ACM International Conference on Information and knowledge management* (2003), pp. 469–475.

[42] Liu, Xiaoyong, and Croft, W. Bruce. Cluster-based retrieval using language models. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (2004), pp. 186–193.

[43] Lytinen, Steven L., and Tomuro, Noriko. The use of question types to match questions in faqfinder. In *Proceedings of the 2002 AAAI Spring Symposium on Mining Answers from Texts and Knowledge Bases* (2002), pp. 46–53.

[44] Lytinen, Steven L., Tomuro, Noriko, and Repede, Tom. The use of wordnet sense tagging in faqfinder. In *Proceedings of the AAAI-2000 workshop on AI and Web Search* (2000).

[45] Malouf, Robert. A comparison of algorithms for maximum entropy parameter estimation. In *Proceedings of Conference on Computational Natural Language Learning* (2002), pp. 49–55.

[46] Metzler, Donald, and Croft, W. Bruce. Analysis of statistical question classification for fact-based questions. *Information Retrieval 8*, 3 (2005), 481–504.

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

[48] Murdock, Vanessa, and Croft, W. Bruce. A translation model for sentence retrieval. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing* (Vancouver, British Columbia, Canada, October 2005), Association for Computational Linguistics, pp. 684–691.

[49] Nigam, K., Lafferty, J., and McCallum, A. Using maximum entropy for text classification. In *Proceedings of IJCAI-99 Workshop on Machine Learning for Information Filtering* (1999), pp. 61–67.

[50] Och, Franz Josef, and Ney, Hermann. A comparison of alignment models for statistical machine translation. In *Proceedings of the 18th conference on Computational linguistics* (Morristown, NJ, USA, 2000), Association for Computational Linguistics, pp. 1086–1090.

[51] Och, Franz Josef, and Ney, Hermann. The alignment template approach to statistical machine translation. *Comput. Linguist. 30*, 4 (2004), 417–449.

[52] Ogilvie, Paul, and Callan, Jamie. Language models and structured document retrieval. In *Proceedings of the first INEX workshop* (2003).

[53] Pang, Bo, Lee, Lillian, and Vaithyanathan, Shivakumar. Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (2002).

[54] Pasca, Marius A., and Harabagiu, Sandra M. High performance question/answering. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (2001), pp. 366–374.

[55] Ponte, Jay M., and Croft, W. Bruce. A language modeling approach to information retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (1998), pp. 275–281.

[56] Ramakrishnan, Ganesh, Chakrabarti, Soumen, Paranjpe, Deepa, and Bhattacharya, Pushpak. Is question answering an acquired skill? In *Proceedings of the 13th international conference on World Wide Web* (2004), pp. 111–120.

[57] Robertson, Stephen E., Walker, Steve, and Hancock-Beaulieu, Micheline. Okapi at trec-7: Automatic ad hoc, filtering, vlc and interactive. In *Proceedings of the the 7th Text Retrieval Conference* (1998), pp. 199–210.

[58] Rocchio, J. J. Relevance feedback in information retrieval. In *The SMART Retrieval System: Experiments in Automatic Indexing.* Prentice Hall, 1971, pp. 324–336.

[59] Schutze, Hinrich, and Pedersen, Jan O. A cooccurrence-based thesaurus and two applications to information retrieval. *Inf. Process. Manage. 33*, 3 (1997), 307–318.

[60] Shinyama, Yusuke, and Sekine, Satoshi. Paraphrase acquisition for information extraction. In *Proceedings of the Second International Workshop on Paraphrasing* (2003), pp. 65–71.

[61] Sneiders, Eriks. Automated faq answering: Continued experience with shallow language understanding. In *Proceedings for the 1999 AAAI Fall Symposium on Question Answering Systems* (1999).

[62] Soricut, Radu, and Brill, Eric. Automatic question answering: Beyond the factoid. In *Proceedings of the Human Language Technology conference / North American chapter of the Association for Computational Linguistics annual meeting* (2004), pp. 57–64.

[63] Soyeon Park, Joon Ho Lee, and Jeon, Jiwoon. Evaluation of the documents from the web-based question and answer service. *Journal of the Korean Society for Library and Information Science 40*, 2 (2006), 299–314.

[64] Strohman, Trevor, Turtle, Howard, and Croft, W. Bruce. Optimization strategies for complex queries. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval* (New York, NY, USA, 2005), ACM Press, pp. 219–225.

[65] Strong, Diane M., Lee, Yang W., and Wang, Richard Y. Data quality in context. *Communications of the ACM 40*, 5 (1997), 103–110.

[66] Tombros, Anastasios, Villa, Robert, and Rijsbergen, C. J. Van. The effectiveness of query-specific hierarchic clustering in information retrieval. *Information Processing and Management 38*, 4 (2002), 559–582.

[67] Voorhees, Ellen M. Query expansion using lexical-semantic relations. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (1994), pp. 61–69.

[68] Voorhees, Ellen M. Overview of the TREC 2001 question answering track. In *Text REtrieval Conference* (2001).

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

[70] Voorhees, Ellen M. The trec 2005 robust track. *SIGIR Forum 40*, 1 (2006), 41–48.

[71] Weaver, Warren. Translation. *Machine Translation of Languages* (1949).

[72] Wen, Ji Rong, Nie, Jian Yun, and Zhang, HongJiang. Query clustering using user logs. *ACM Transactions on Information Systems 20*, 1 (2002), 59–81.

[73] Wu, Chung-Hsien, Yeh, Jui-Feng, and Chen, Ming-Jun. Domain-specific faq retrieval using independent aspects. *ACM Transactions on Asian Language Information Processing 4*, 1 (2005), 1–17.

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

[75] Xu, Jinxi, Fraser, Alexander, and Weischedel, Ralph M. TREC 2001 cross-lingual retrieval at BBN. In *Text REtrieval Conference* (2001).

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

[77] Zhou, Xiaohua, Hu, Xiaohua, Zhang, Xiaodan, Lin, Xia, and Song, Il-Yeol. Context-sensitive semantic smoothing for the language modeling approach to genomic ir. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval* (New York, NY, USA, 2006), ACM Press, pp. 170–177.

[78] Zhou, Yun, and Croft, W. Bruce. Document quality models for web ad hoc retrieval. In *Proceedings of the ACM Fourteenth Conference on Information and Knowledge Management* (2005), pp. 331–332.

[79] Zhu, Xiaolan, and Gauch, Susan. Incorporating quality metrics in centralized/distributed information retrieval on the world wide web. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (2000), pp. 288–295.

[80] Zobel, Justin. How reliable are the results of large-scale information retrieval experiments? In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval* (New York, NY, USA, 1998), ACM Press, pp. 307–314.