Bridging Lexical Gaps between Queries and Questions on Large Online Q&A Collections with Compact Translation Models

Jung-Tae Lee[†] and **Sang-Bum Kim**[§] and **Young-In Song**[‡] and **Hae-Chang Rim**[†]
[†]Dept. of Computer & Radio Communications Engineering, Korea University, Seoul, Korea
[§]Search Business Team, SK Telecom, Seoul, Korea
[‡]Dept. of Computer Science & Engineering, Korea University, Seoul, Korea
{jtlee,sbkim,song,rim}@nlp.korea.ac.kr

Abstract

Lexical gaps between queries and questions (documents) have been a major issue in question retrieval on large online question and answer (Q&A) collections. Previous studies address the issue by *implicitly* expanding queries with the help of translation models pre-constructed using statistical techniques. However, since it is possible for *unimportant* words (e.g., non-topical words, common words) to be included in the translation models, a lack of noise control on the models can cause degradation of retrieval performance. This paper investigates a number of empirical methods for eliminating unimportant words in order to construct *compact translation models* for retrieval purposes. Experiments conducted on a real world Q&A collection show that substantial improvements in retrieval performance can be achieved by using compact translation models.

1 Introduction

Community-driven question answering services, such as Yahoo! Answers[1] and Live Search QnA[2], have been rapidly gaining popularity among Web users interested in sharing information online. By inducing users to collaboratively submit questions and answer questions posed by other users, large amounts of information have been collected in the form of question and answer (Q&A) pairs in recent years. This user-generated information is a valuable resource for many information seekers, because users can acquire information straightforwardly by searching through answered questions that satisfy their information need.

Retrieval models for such Q&A collections should manage to handle the lexical gaps or word mismatches between user questions (queries) and answered questions in the collection. Consider the two following examples of questions that are semantically similar to each other:

- *"Where can I get cheap airplane tickets?"*
- *"Any travel website for low airfares?"*

Conventional word-based retrieval models would fail to capture the similarity between the two, because they have no words in common. To bridge the query-question gap, prior work on Q&A retrieval by Jeon et al. (2005) implicitly expands queries with the use of pre-constructed translation models, which lets you generate query words not in a question by translation to alternate words that are related. In practice, these translation models are often constructed using statistical machine translation techniques that primarily rely on word co-occurrence statistics obtained from parallel strings (e.g., question-answer pairs).

A critical issue of the translation-based approaches is the quality of translation models constructed in advance. If no noise control is conducted during the construction, it is possible for translation models to contain "unnecessary" translations (i.e., translating a word into an unimportant word, such as a non-topical or common word). In the query expansion viewpoint, an attempt to identify and decrease

[1]http://answers.yahoo.com/
[2]http://qna.live.com/

Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, pages 410–418,
Honolulu, October 2008.

the proportion of unnecessary translations in a translation model may produce an effect of "selective" implicit query expansion and result in improved retrieval. However, prior work on translation-based Q&A retrieval does not recognize this issue and uses the translation model as it is; essentially no attention seems to have been paid to improving the performance of the translation-based approach by enhancing the quality of translation models.

In this paper, we explore a number of empirical methods for selecting and eliminating unimportant words from parallel strings to avoid unnecessary translations from being learned in translation models built for retrieval purposes. We use the term *compact translation models* to refer to the resulting models, since the total number of parameters for modeling translations would be minimized naturally. We also present experiments in which compact translation models are used in Q&A retrieval. The main goal of our study is to investigate if and how compact translation models can improve the performance of Q&A retrieval.

The rest of this paper is organized as follows. The next section introduces a translation-based retrieval model and accompanying techniques used to retrieve query-relevant questions. Section 3 presents a number of empirical ways to select and eliminate unimportant words from parallel strings for training compact translation models. Section 4 summarizes the compact translation models we built for retrieval experiments. Section 5 presents and discusses the results of retrieval experiments. Section 6 presents related works. Finally, the last section concludes the paper and discusses future directions.

2 Translation-based Retrieval Model

This section introduces the translation-based language modeling approach to retrieval that has been used to bridge the lexical gap between queries and already-answered questions in this paper.

In the basic language modeling framework for retrieval (Ponte and Croft, 1998), the similarity between a query Q and a document D for ranking may be modeled as the probability of the document language model M_D built from D generating Q:

$$sim(Q, D) \approx P(Q|M_D) \tag{1}$$

Assuming that query words occur independently given a particular document language model, the query-likelihood $P(Q|M_D)$ is calculated as:

$$P(Q|M_D) = \prod_{q \in Q} P(q|M_D) \tag{2}$$

where q represents a query word.

To avoid zero probabilities in document language models, a mixture between a document-specific multinomial distribution and a multinomial distribution estimated from the entire document collection is widely used in practice:

$$P(Q|M_D) = \prod_{q \in Q} \Big[(1-\lambda) \cdot P(q|M_D) + \lambda \cdot P(q|M_C)\Big] \tag{3}$$

where $0 < \lambda < 1$ and M_C represents a language model built from the entire collection. The probabilities $P(w|M_D)$ and $P(w|M_C)$ are calculated using maximum likelihood estimation.

The basic language modeling framework does not address the issue of lexical gaps between queries and question. Berger and Lafferty (1999) viewed information retrieval as statistical document-query translation and introduced translation models to map query words to document words. Assuming that a translation model can be represented by a conditional probability distribution of translation $T(\cdot|\cdot)$ between words, we can model $P(q|M_D)$ in Equation 3 as:

$$P(q|M_D) = \sum_{w \in D} T(q|w) P(w|M_D) \tag{4}$$

where w represents a document word.[3]

The translation probability $T(q|w)$ virtually represents the degree of relationship between query word q and document word w captured in a different, machine translation setting. Then, in the traditional information retrieval viewpoint, the use of translation models produce an implicit query expansion effect, since query words not in a document are mapped to related words in the document. This implies that translation-based retrieval models would make positive contributions to retrieval performance only when the pre-constructed translation models have reliable translation probability distributions.

[3]The formulation of our retrieval model is basically equivalent to the approach of Jeon et al. (2005).

2.1 IBM Translation Model 1

Obviously, we need to build a translation model in advance. Usually the IBM Model 1, developed in the statistical machine translation field (Brown et al., 1993), is used to construct translation models for retrieval purposes in practice. Specifically, given a number of parallel strings, the IBM Model 1 learns the translation probability from a source word s to a target word t as:

$$T(t|s) = \lambda_s^{-1} \sum_{i}^{N} c(t|s; J_i) \quad (5)$$

where λ_s is a normalization factor to make the sum of translation probabilities for the word s equal to 1, N is the number of parallel string pairs, and J_i is the ith parallel string pair. $c(t|s; J_i)$ is calculated as:

$$c(t|s; J_i) = \left(\frac{P(t|s)}{P(t|s_1) + \cdots + P(t|s_n)} \right) \times freq_{t,J_i} \times freq_{s,J_i} \quad (6)$$

where $\{s_1, \ldots, s_n\}$ are words in the source text in J^i. $freq_{t,J^i}$ and $freq_{s,J^i}$ are the number of times that t and s occur in J_i, respectively.

Given the initial values of $T(t|s)$, Equations (5) and (6) are used to update $T(t|s)$ repeatedly until the probabilities converge, in an EM-based manner.

Note that the IBM Model 1 solely relies on word co-occurrence statistics obtained from parallel strings in order to learn translation probabilities. This implies that if parallel strings have unimportant words, a resulted translation model based on IBM Model 1 may contain unimportant words with non-zero translation probabilities.

We alleviate this drawback by eliminating unimportant words from parallel strings, avoiding them from being included in the conditional translation probability distribution. This naturally induces the construction of compact translation models.

2.2 Gathering Parallel Strings from Q&A Collections

The construction of statistical translation models previously discussed requires a corpus consisting of parallel strings. Since monolingual parallel texts are generally not available in real world, one must artificially generate a "synthetic" parallel corpus.

Question and answer as parallel pairs: The simplest approach is to directly employ questions and their answers in the collections by setting either as source strings and the other as target strings, with the assumption that a question and its corresponding answer are naturally parallel to each other. Formally, if we have a Q&A collection as $C = \{D_1, D_2, \ldots, D_n\}$, where D_i refers to an ith Q&A data consisting of a question q_i and its answer a_i, we can construct a parallel corpus C' as $\{(q_1, a_1), \ldots, (q_n, a_n)\} \cup \{(a_1, q_1), \ldots, (a_n, q_n)\} = C'$ where each element (s, t) refers to a parallel pair consisting of source string s and target string t. The number of parallel string samples would eventually be twice the size of the collections.

Similar questions as parallel pairs: Jeon et al. (2005) proposed an alternative way of automatically collecting a relatively larger set of parallel strings from Q&A collections. Motivated by the observation that many semantically identical questions can be found in typical Q&A collections, they used similarities between answers calculated by conventional word-based retrieval models to automatically group questions in a Q&A collection as pairs. Formally, two question strings q_i and q_j would be included in a parallel corpus C' as $\{(q_i, q_j), (q_j, q_i)\} \subset C'$ only if their answer strings a_i and a_j have a similarity higher than a pre-defined threshold value. The similarity is calculated as the reverse of the harmonic mean of ranks as $sim(a_i, a_j) = \frac{1}{2}(\frac{1}{r_j} + \frac{1}{r_i})$, where r_j and r_i refer to the rank of the a_j and a_i when a_i and a_j are given as queries, respectively. This approach may artificially produce much more parallel string pairs for training the IBM Model 1 than the former approach, depending on the threshold value.[4]

To our knowledge, there has not been any study comparing the effectiveness of the two approaches yet. In this paper, we try both approaches and compare the effectiveness in retrieval performance.

3 Eliminating Unimportant Words

We adopt a term weight ranking approach to identify and eliminate unimportant words from parallel strings, assuming that a word in a string is unim-

[4] We have empirically set the threshold (0.05) for our experiments.

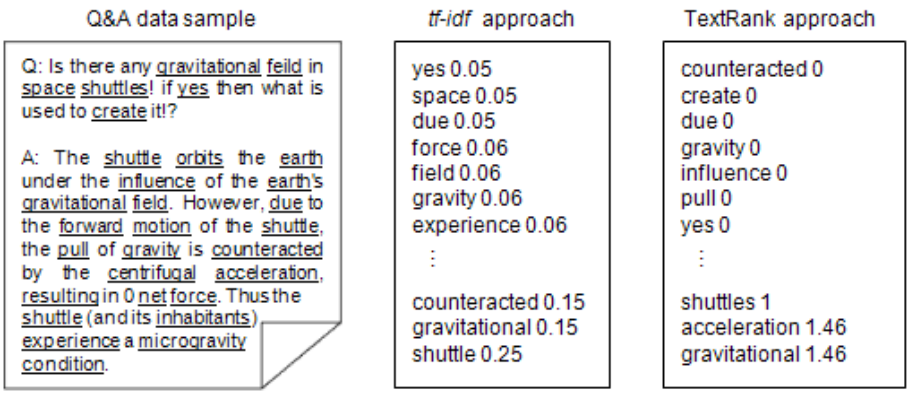

Figure 1: Term weighting results of tf-idf and TextRank (window=3). Weighting is done on underlined words only.

portant if it holds a relatively low significance in the document (Q&A pair) of which the string is originally taken from. Some issues may arise:

- How to assign a weight to each word in a document for term ranking?
- How much to remove as unimportant words from the ranked list?

The following subsections discuss strategies we use to handle each of the issues above.

3.1 Assigning Term Weights

In this section, the two different term weighting strategies are introduced.

tf-idf: The use of *tf-idf* weighting on evaluating how unimportant a word is to a document seems to be a good idea to begin with. We have used the following formulas to calculate the weight of word w in document D:

$$tf\text{-}idf_{w,D} = tf_{w,D} \times idf_w \tag{7}$$

$$tf_{w,D} = \frac{freq_{w,D}}{|D|}, \quad idf_w = \log \frac{|C|}{df_w}$$

where $freq_{w,D}$ refers to the number of times w occurs in D, $|D|$ refers to the size of D (in words), $|C|$ refers to the size of the document collection, and df_w refers to the number of documents where w appears. Eventually, words with low *tf-idf* weights may be considered as unimportant.

TextRank: The task of term weighting, in fact, has been often applied to the keyword extraction task in natural language processing studies. As an alternative term weighting approach, we have used a variant of Mihalcea and Tarau (2004)'s TextRank, a graph-based ranking model for keyword extraction which achieves state-of-the-art accuracy without the need of deep linguistic knowledge or domain-specific corpora.

Specifically, the ranking algorithm proceeds as follows. First, words in a given document are added as vertices in a graph G. Then, edges are added between words (vertices) if the words co-occur in a fixed-sized window. The number of co-occurrences becomes the weight of an edge. When the graph is constructed, the score of each vertex is initialized as 1, and the PageRank-based ranking algorithm is run on the graph iteratively until convergence. The TextRank score of a word w in document D at kth iteration is defined as follows:

$$R^k_{w,D} = (1-d) + d \cdot \sum_{\forall j:(i,j)\in G} \frac{e_{i,j}}{\sum_{\forall l:(j,l)\in G} e_{j,l}} R^{k-1}_{w,D} \tag{8}$$

where d is a damping factor usually set to 0.85, and $e_{i,j}$ is an edge weight between i and j.

The assumption behind the use of the variant of TextRank is that a word is likely to be an important word in a document if it co-occurs frequently with other important words in the document. Eventually, words with low TextRank scores may be considered as unimportant. The main differences of TextRank compared to *tf-idf* is that it utilizes the context information of words to assign term weights.

Figure 1 demonstrates that term weighting results of TextRank and *tf-idf* are greatly different. Notice that TextRank assigns low scores to words that co-

Corpus: (Q‖A)	**Vocabulary Size (%chg)**		**Average Translations (%chg)**	
	tf-idf	*TextRank*	*tf-idf*	*TextRank*
Initial	90,441		73	
25%Removal	90,326 (∇0.1%)	73,021 (∇19.3%)	73 (∇0.0%)	44 (∇39.7%)
50%Removal	90,230 (∇0.2%)	72,225 (∇20.1%)	72 (∇1.4%)	43 (∇41.1%)
75%Removal	88,763 (∇1.9%)	65,268 (∇27.8%)	53 (∇27.4%)	38 (∇47.9%)
Avg.Score	66,412 (∇26.6%)	31,849 (∇64.8%)	14 (∇80.8%)	18 (∇75.3%)

Table 1: Impact of various word elimination strategies on translation model construction using (Q‖A) corpus.

Corpus: (Q‖Q)	**Vocabulary Size (%chg)**		**Average Translations (%chg)**	
	tf-idf	*TextRank*	*tf-idf*	*TextRank*
Initial	34,485		442	
25%Removal	34,374 (∇0.3%)	26,900 (∇22.0%)	437 (∇1.1%)	282 (∇36.2%)
50%Removal	34,262 (∇0.6%)	26,421 (∇23.4%)	423 (∇4.3%)	274 (∇38.0%)
75%Removal	32,813 (∇4.8%)	23,354 (∇32.3%)	288 (∇34.8%)	213 (∇51.8%)
Avg.Score	28,613 (∇17.0%)	16,492 (∇52.2%)	163 (∇63.1%)	164 (∇62.9%)

Table 2: Impact of various word elimination strategies on translation model construction using (Q‖Q) corpus.

occur only with stopwords. This implies that TextRank weighs terms more "strictly" than the *tf-idf* approach, with use of contexts of words.

3.2 Deciding the Quantity to be Removed from Ranked List

Once a final score (either *tf-idf* or TextRank score) is obtained for each word, we create a list of words ranked in decreasing order of their scores and eliminate the ones at lower ranks as unimportant words. The question here is how to decide the proportion or quantity to be removed from the ranked list.

Removing a fixed proportion: The first approach we have used is to decide the number of unimportant words based on the size of the original string. For our experiments, we manually vary the proportion to be removed as 25%, 50%, and 75%. For instance, if the proportion is set to 50% and an original string consists of ten words, at most five words would be remained as important words.

Using average score as threshold: We also have used an alternate approach to deciding the quantity. Instead of eliminating a *fixed* proportion, words are removed if their score is lower than the average score of all words in a document. This approach decides the proportion to be removed more flexibly than the former approach.

4 Building Compact Translation Models

We have initially built two parallel corpora from a Q&A collection[5], denoted as (Q‖A) corpus and (Q‖Q) corpus henceforth, by varying the methods in which parallel strings are gathered (described in Section 2.2). The (Q‖A) corpus consists of 85,938 parallel string pairs, and the (Q‖Q) corpus contains 575,649 parallel string pairs.

In order to build compact translation models, we have preprocessed the parallel corpus using different word elimination strategies so that unimportant words would be removed from parallel strings. We have also used a stoplist[6] consisting of 429 words to remove stopwords. The out-of-the-box GIZA++[7] (Och and Ney, 2004) has been used to learn translation models using the pre-processed parallel corpus for our retrieval experiments. We have also trained initial translation models, using a parallel corpus from which only the stopwords are removed, to compare with the compact translation models.

Eventually, the number of parameters needed for modeling translations would be minimized if unimportant words are eliminated with different ap-

[5]Details on this data will be introduced in the next section.
[6]http://truereader.com/manuals/onix/stopwords1.html
[7]http://www.fjoch.com/GIZA++.html

proaches. Table 1 and 2 shows the impact of various word elimination strategies on the construction of compact translation models using the (Q‖A) corpus and the (Q‖Q) corpus, respectively. The two tables report the size of the vocabulary contained and the average number of translations per word in the resulting compact translation models, along with percentage decreases with respect to the initial translation models in which only stopwords are removed. We make these observations:

- The translation models learned from the (Q‖Q) corpus have *less* vocabularies but *more* average translations per word than the ones learned from the (Q‖A) corpus. This result implies that a large amount of noise may have been created inevitably when a large number of parallel strings (pairs of similar questions) were artificially gathered from the Q&A collection.

- The TextRank strategy tends to eliminate larger sets of words as unimportant words than the *tf-idf* strategy when a fixed proportion is removed, regardless of the corpus type. Recall that the TextRank approach assigns weights to words more strictly by using contexts of words.

- The approach to remove words according to the average weight of a document (denoted as *Avg.Score*) tends to eliminate relatively larger portions of words as unimportant words than any of the fixed-proportion strategies, regardless of either the corpus type or the ranking strategy.

5 Retrieval Experiments

Experiments have been conducted on a real world Q&A collection to demonstrate the effectiveness of compact translation models on Q&A retrieval.

5.1 Experimental Settings

In this section, four experimental settings for the Q&A retrieval experiments are described in detail.

Data: For the experiments, Q&A data have been collected from the *Science* domain of Yahoo! Answers, one of the most popular community-based question answering service on the Web. We have obtained a total of 43,001 questions with a best answer (selected either by the questioner or by votes of other users) by recursively traversing subcategories of the *Science* domain, with up to 1,000 question pages retrieved.[8]

Among the obtained Q&A pairs, 32 Q&A pairs have been randomly selected as the test set, and the remaining 42,969 questions have been the reference set to be retrieved. Each Q&A pair has three text fields: question title, question content, and answer.[9] The fields of each Q&A pair in the test set are considered as various test queries; the question title, the question content, and the answer are regarded as a short query, a long query, and a supplementary query, respectively. We have used long queries and supplementary queries only in the relevance judgment procedure. All retrieval experiments have been conducted using short queries only.

Relevance judgments: To find relevant Q&A pairs given a short query, we have employed a pooling technique used in the TREC conference series. We have pooled the top 40 Q&A pairs from each retrieval results generated by varying the retrieval algorithms, the search field, and the query type. Popular word-based models, including the Okapi BM25, query-likelihood language model, and previous translation-based models (Jeon et al., 2005), have been used.[10]

Relevance judgments have been done by two student volunteers (both fluent in English). Since many community-based question answering services present their search results in a hierarchical fashion (*i.e.* a list of relevant questions is shown first, and then the user chooses a specific question from the list to see its answers), a Q&A pair has been judged as relevant if its question is semantically similar to the query; neither quality nor rightness of the answer has not been considered. When a disagreement has been made between two volunteers, one of the authors has made the final judgment. As a result, 177 relevant Q&A pairs have been found in total for the 32 short queries.

Baseline retrieval models: The proposed ap-

[8] Yahoo! Answers did not expose additional question pages to external requests at the time of collecting the data.

[9] When collecting parallel strings from the Q&A collection, we have put together the question title and the question content as one question string.

[10] The retrieval model using compact translation models has not been used in the pooling procedure.

proach to Q&A retrieval using compact translation models (denoted as CTLM henceforth) is compared to three baselines:

QLM: Query-likelihood language model for retrieval (equivalent to Equation 3, without use of translation models). This model represents word-based retrieval models widely used in practice.

TLM(Q‖Q): Translation-based language model for question retrieval (Jeon et al., 2005). This model uses IBM Model 1 learned from the (Q‖Q) corpus of which stopwords are removed.

TLM(Q‖A): A variant of the translation-based approach. This model uses IBM model 1 learned from the (Q‖A) corpus.

Evaluation metrics: We have reported the retrieval performance in terms of Mean Average Precision (MAP) and Mean R-Precision (R-Prec).

Average Precision can be computed based on the precision at each relevant document in the ranking. Mean Average Precision is defined as the mean of the Average Precision values across the set of all queries:

$$MAP(Q) = \frac{1}{|Q|} \sum_{q \in Q} \frac{1}{m_q} \sum_{k=1}^{m_q} Precision(R_k) \quad (9)$$

where Q is the set of test queries, m_q is the number of relevant documents for a query q, R_k is the set of ranked retrieval results from the top until rank position k, and $Precision(R_k)$ is the fraction of relevant documents in R_k (Manning et al., 2008).

R-Precision is defined as the precision after R documents have been retrieved where R is the number of relevant documents for the current query (Buckley and Voorhees, 2000). Mean R-Precision is the mean of the R-Precisions across the set of all queries.

We take MAP as our primary evaluation metric.

5.2 Experimental Results

Preliminary retrieval experiments have been conducted using the baseline QLM and different fields of Q&A data as retrieval unit. Table 3 shows the effectiveness of each field.

The results imply that the question title field is the most important field in our Yahoo! Answers collection; this also supports the observation presented by Jeon et al. (2005). Based on the preliminary observations, all retrieval models tested in this paper have ranked Q&A pairs according to the similarity scores between queries and question titles.

Retrieval unit	**MAP**	**R-Prec**
Question title	**0.1031**	**0.2396**
Question content	0.0422	0.0999
Answer	0.0566	0.1062

Table 3: Preliminary retrieval results.

Model	**MAP (%chg)**	**R-Prec (%chg)**
QLM	0.1031	0.2396
TLM(Q‖Q)*	0.1121 (△9%)	0.2251 (▽6%)
CTLM(Q‖Q)	**0.1415** (△37%)	**0.2425** (△1%)
TLM(Q‖A)	0.1935 (△88%)	0.3135 (△31%)
CTLM(Q‖A)	**0.2095** (△103%)	**0.3585** (△50%)

Table 4: Comparisons with three baseline retrieval models. * indicates that it is equivalent to Jeon et al. (2005)'s approach. MAP improvements of CTLMs have been tested to be statistically significant using paired t-test.

Table 4 presents the comparison results of three baseline retrieval models and the proposed CTLMs. For each method, the best performance after empirical λ parameter tuning according to MAP is presented.

Notice that both the TLMs and CTLMs have outperformed the word-based QLM. This implies that word-based models that do not address the issue of lexical gaps between queries and questions often fail to retrieve relevant Q&A data that have little word overlap with queries, as noted by Jeon et al. (2005).

Moreover, notice that the proposed CTLMs have achieved significantly better performances in all evaluation metrics than both QLM and TLMs, regardless of the parallel corpus in which the incorporated translation models are trained from. This is a clear indication that the use of compact translation models built with appropriate word elimination strategies is effective in closing the query-question lexical gaps

(Q‖Q)	MAP (%chg)	
	tf-idf	*TextRank*
Initial	0.1121	
25%Rmv	0.1141 (△1.8)	0.1308 (△16.7)
50%Rmv	0.1261 (△12.5)	0.1334 (△19.00)
75%Rmv	0.1115 (▽0.5)	0.1160 (△3.5)
Avg.Score	0.1056 (▽5.8)	**0.1415 (△26.2)**

Table 5: Contributions of various word elimination strategies on MAP performance of CTLM(Q‖Q).

(Q‖A)	MAP (%chg)	
	tf-idf	*TextRank*
Initial	0.1935	
25%Rmv	**0.2095 (△8.3)**	0.1733 (▽10.4)
50%Rmv	0.2085 (△7.8)	0.1623 (▽16.1)
75%Rmv	0.1449 (▽25.1)	0.1515 (▽21.7)
Avg.Score	0.1168 (▽39.6)	0.1124 (▽41.9)

Table 6: Contributions of various word elimination strategies on MAP performance of CTLM(Q‖A).

for improving the performance of question retrieval in the context of language modeling framework.

Note that the retrieval performance varies by the type of training corpus; CTLM(Q‖A) has outperformed CTLM(Q‖Q) significantly. This proves the statement we made earlier that the (Q‖Q) corpus would contain much noise since the translation models learned from the (Q‖Q) corpus tend to have smaller vocabulary sizes but significantly more average translations per word than the ones learned from the (Q‖A) corpus.

Table 5 and 6 show the effect of various word elimination strategies on the retrieval performance of CTLMs in which the incorporated compact translation models are trained from the (Q‖Q) corpus and the (Q‖A) corpus, respectively. It is interesting to note that the importance of modifications in word elimination strategies also varies by the type of training corpus.

The retrieval results indicate that when the translation model is trained from the "less noisy" (Q‖A) corpus, eliminating a relatively large proportions of words may hurt the retrieval performance of CTLM. In the case when the translation model is trained from the "noisy" (Q‖Q) corpus, a better retrieval performance may be achieved if words are eliminated appropriately to a certain extent.

In terms of weighting scheme, the TextRank approach, which is more "strict" than *tf-idf* in eliminating unimportant words, has led comparatively higher retrieval performances on all levels of removal quantity when the translation model has been trained from the "noisy" (Q‖Q) corpus. On the contrary, the "less strict" *tf-idf* approach has led better performances when the translation model has been trained from the "less noisy" (Q‖A) corpus.

In summary, the results imply that the performance of translation-based retrieval models can be significantly improved when strategies for building of compact translation models are chosen properly, regarding the expected noise level of the parallel corpus for training the translation models. In a case where a noisy parallel corpus is given for training of translation models, it is better to get rid of noise as much as possible by using "strict" term weighting algorithms; when a less noisy parallel corpus is given for building the translation models, a tolerant approach would yield better retrieval performance.

6 Related Works

Our work is most closely related to Jeon et al. (2005)'s work, which addresses the issue of word mismatch between queries and questions in large online Q&A collections by using translation-based methods. Apart from their work, there have been some related works on applying translation-based methods for retrieving FAQ data. Berger et al. (2000) report some of the earliest work on FAQ retrieval using statistical retrieval models, including translation-based approaches, with a small set of FAQ data. Soricut and Brill (2004) present an answer passage retrieval system that is trained from 1 million FAQs collected from the Web using translation methods. Riezler et al. (2007) demonstrate the advantages of translation-based approach to answer retrieval by utilizing a more complex translation model also trained from a large amount of data extracted from FAQs on the Web. Although all of these translation-based approaches are based on the statistical translation models, including the IBM Model 1, none of them focus on addressing the noise issues in translation models.

7 Conclusion and Future Work

Bridging the query-question gap has been a major issue in retrieval models for large online Q&A collections. In this paper, we have shown that the performance of translation-based retrieval on real online Q&A collections can be significantly improved by using compact translation models of which the noise (unimportant word translations) is properly reduced. We have also observed that the performance enhancement may be achieved by choosing the appropriate strategies regarding the strictness of various term weighting algorithms and the expected noise level of the parallel data for learning such translation models.

Future work will focus on testing the effectiveness of the proposed method on a larger set of Q&A collections with broader domains. Since the proposed approach cannot handle many-to-one or one-to-many word transformations, we also plan to investigate the effectiveness of phrase-based translation models in closing gaps between queries and questions for further enhancement of Q&A retrieval.

Acknowledgments

This work was supported by Microsoft Research Asia. Any opinions, findings, and conclusions or recommendations expressed above are those of the authors and do not necessarily reflect the views of the sponsor.

References

Adam Berger, Rich Caruana, David Cohn, Dayne Freitag, and Vibhu Mittal. 2000. 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*, pages 192–199.

Adam Berger and John Lafferty. 1999. Information Retrieval as Statistical Translation. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 222–229.

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263–311.

Chris Buckley and Ellen M. Voorhees. 2000. Evaluating Evaluation Measure Stability. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 33–40.

Jiwoon Jeon, W. Bruce Croft, and Joon Ho Lee. 2005. Finding Similar Questions in Large Question and Answer Archives. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, pages 84–90.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing Order into Text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411.

Franz J. Och and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51.

Jay M. Ponte and W. Bruce Croft. 1998. A Language Modeling Approach to Information Retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 275–281.

Stefan Riezler, Alexander Vasserman, Ioannis Tsochantaridis, Vibhu Mittal, and Yi Liu. 2007. Statistical Machine Translation for Query Expansion in Answer Retrieval. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 464–471.

Radu Soricut and Eric Brill. 2004. Automatic Question Answering: Beyond the Factoid. In *Proceedings of the 2004 Human Language Technology and Conference of the North American Chapter of the Association for Computational Linguistics*, pages 57–64.