Reranking Answers for Definitional QA Using Language Modeling

Yi Chen
School of Software Engineering
Chongqing University
Chongqing, China, 400044
126cy@126.com

Ming Zhou
Microsoft Research Asia
5F Sigma Center, No.49 Zhichun Road, Haidian
Bejing, China, 100080
mingzhou@microsoft.com

Shilong Wang
College of Mechanical Engineering
Chongqing University
Chongqing, China, 400044
slwang@cqu.edu.cn

Abstract[*]

Statistical ranking methods based on centroid vector (profile) extracted from external knowledge have become widely adopted in the top definitional QA systems in TREC 2003 and 2004. In these approaches, terms in the centroid vector are treated as a bag of words based on the independent assumption. To relax this assumption, this paper proposes a novel language model-based answer reranking method to improve the existing bag-of-words model approach by considering the dependence of the words in the centroid vector. Experiments have been conducted to evaluate the different dependence models. The results on the TREC 2003 test set show that the reranking approach with biterm language model, significantly outperforms the one with the bag-of-words model and unigram language model by 14.9% and 12.5% respectively in F-Measure(5).

1 Introduction

In recent years, QA systems in TREC (Text REtrieval Conference) have made remarkable progress (Voorhees, 2002). The task of TREC QA before 2003 has mainly focused on the *factoid* questions, in which the answer to the question is a number, a person name, or an organization name, or the like.

Questions like "Who is Colin Powell?" or "What is mold?" are *definitional* questions (Voorhees, 2003). Statistics from 2,516 Frequently Asked Questions (FAQ) extracted from *Internet FAQ Archives*[1] show that around 23.6% are definitional questions. This indicates that definitional questions occur frequently and are important question types. TREC started the evaluation for definitional QA in 2003. The definitional QA systems in TREC are required to extract definitional nuggets/sentences that contain the highly descriptive information about the question target from a given large corpus.

For definitional question, statistical ranking methods based on centroid vector (profile) extracted from external resources, such as the online encyclopedia, are widely adopted in the top systems in TREC 2003 and 2004 (Xu et al., 2003; Blair-Goldensohn et al., 2003; Wu et al., 2004). In these systems, for a given question, a vector is formed consisting of the most frequent co-occurring terms with the question target as the question profile. Candidate answers extracted from a given large corpus are ranked based on their similarity to the question profile. The similarity is normally the *TFIDF* score in which both the candidate answer and the question profile are treated as a bag of words in the framework of Vector Space Model (VSM).

VSM is based on an independence assumption, which assumes that terms in a vector are statistically independent from one another. Although this assumption makes the development of retrieval models easier and the retrieval operation tractable, it does not hold in textual data. For example, for question "Who is Bill Gates?" words "born" and "1955" in the candidate answer are not independent.

In this paper, we are interested in considering the term dependence to improve the answer reranking for definitional QA. Specifically, the

[*] This work was finished while the first author was visiting Microsoft Research Asia during March 2005-March 2006 as a component of the project of AskBill Chatbot led by Dr. Ming Zhou.

[1] http://www.faqs.org/faqs/

Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL, pages 1081–1088,
Sydney, July 2006.

language model is utilized to capture the term dependence. A language model is a probability distribution that captures the statistical regularities of natural language use. In a language model, key elements are the probabilities of word sequences, denoted as $P(w_1, w_2, ..., w_n)$ or $P\ (w_{1,n})$ for short. Recently, language model has been successfully used for information retrieval (IR) (Ponte and Croft, 1998; Song and Croft, 1998; Lafferty et al., 2001; Gao et al., 2004; Cao et al., 2005). Our natural thinking is to apply language model to rank the candidate answers as it has been applied to rank search results in IR task.

The basic idea of our research is that, given a definitional question *q*, an ordered centroid *OC* which is learned from the web and a language model *LM(OC)* which is trained with it. Candidate answers can be ranked by probability estimated by *LM(OC)*. A series of experiments on standard TREC 2003 collection have been conducted to evaluate bigram and biterm language models. Results show that both these two language models produce promising results by capturing the term dependence and biterm model achieves the best performance. Biterm language model interpolating with unigram model significantly improves the VSM and unigram model by 14.9% and 12.5% in F-Measure(5).

In the rest of this paper, Section 2 reviews related work. Section 3 presents details of the proposed method. Section 4 introduces the structure of our experimental system. We show the experimental results in Section 5, and conclude the paper in Section 6.

2 Related Work

Web information has been widely used for answer reranking and validation. For factoid QA task, *AskMSR* (Brill et al., 2001) ranks the answers by counting the occurrences of candidate answers returned from a search engine. Similarly, *DIOGENE* (Magnini et al., 2002) applies search engines to validate candidate answers.

For definitional QA task, Lin (2002) presented an approach in which web-based answer reranking is combined with dictionary-based (e.g., WordNet) reranking, which leads to a 25% increase in mean reciprocal rank (MRR). Xu et al. (2003) proposed a statistical ranking method based on centroid vector (i.e., vector of words and frequencies) learned from the online encyclopedia (i.e., *Wikipedia*[2]) and the web. Candidate answers were reranked based on their similarity (*TFIDF* score) to the centroid vector. Similar techniques were explored in (Blair-Goldensohn et al., 2003). In this paper, we explore the dependence among terms in centroid vector for improving the answer reranking for definitional QA.

In recent years, language modeling has been widely employed in IR (Ponte and Croft, 1998; Song and Croft, 1998; Miller and Zhai, 1999; Lafferty and Zhai, 2001). The basic idea is to compute the conditional probability $P(Q|D)$, i.e., the probability of generating a query Q given the observation of a document D. The searched documents are ranked in descending order of this probability.

Song and Croft (1998) proposed a general language model to incorporate word dependence by using bigrams. Srikanth and Srihari (2002) introduced biterm language models similar to the bigram model except that the constraint of order in terms is relaxed and improved performance was observed. Gao et al. (2004) presented a new method of capturing word dependencies, in which they extended state-of-the-art language modeling approaches to information retrieval by introducing a dependence structure that learned from training data. Cao et al. (2005) proposed a novel dependence model to incorporate both relationships of WordNet and co-occurrence with the language modeling framework for IR. In our approach, we propose bigram and biterm models to capture the term dependence in centroid vector.

Applying language modeling for the QA task has not been widely researched. Zhang D. and Lee (2003) proposed a method using language model for passage retrieval for the factoid QA. They trained two language models, in which one was the question-topic language model and the other was passage language model. They utilized the divergence between the two language models to rank passages. In this paper, we focus on reranking answers for definitional questions.

As other ranking approaches, Xu, et al. (2005) formalized ranking definitions as classification problems, and Cui et al. (2004) proposed soft patterns to rank answers for definitional QA.

3 Reranking Answers Using Language Model

3.1 Model background

In practice, language model is often approximated by N-gram models.

Unigram:

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

$$P(w_{1,n}) = P(w_1)P(w_2)...P(w_n) \quad (1)$$

Bigram:

$$P(w_{1,n}) = P(w_1)P(w_2|w_1)...P(w_n|w_{n-1}) \quad (2)$$

The unigram model makes a strong assumption that each word occurs independently. The bigram model takes the local context into consideration. It has been proved to work better than the unigram language model in IR (e.g., Song and Croft, 1998).

Biterm language models are similar to bigram language models except that the constraint of order in terms is relaxed. Therefore, a document containing *information retrieval* and a document containing *retrieval (of) information* will be assigned the same generation probability. The biterm probabilities can be approximated using the frequency of occurrence of terms.

Three approximation methods were proposed in Srikanth and Srihari (2002). The so-called min-Adhoc approximation truly relaxes the constraint of word order and outperformed other two approximation methods in their experiments.

$$P_{BT}(w_i \mid w_{i-1}) \approx \frac{C(w_{i-1}, w_i) + C(w_i, w_{i-1})}{\min\{C(w_{i-1}), C(w_i)\}} \quad (3)$$

Equation (3) is the min-Adhoc approximation. Where *C(X)* gives the occurrences of the string *X*.

3.2 Reranking based on language model

In our approach, we adopt bigram and biterm language models. As a smoothing approach, linear interpolation of unigrams and bigrams is employed.

Given a candidate answer $A=t_1t_2...t_i...t_n$ and a bigram or biterm back-off language model *OC* trained with the ordered centroid, the probability of generating *A* can be estimated by Equation (4).

$$\begin{aligned} P(A \mid OC) &= P(t_1, ..., t_n \mid OC) \quad (4) \\ &= P(t_1 \mid OC)\prod_{i=2}^{n}[\lambda P(t_i \mid OC) + (1-\lambda)P(t_i \mid t_{i-1}, OC)] \end{aligned}$$

where *OC* stands for the language model of the ordered centroid and λ is the mixture weight combining the unigram and bigram (or biterm) probabilities. After taking logarithm and exponential for Equation (4), we get Equation (5).

$$Score(A) = \exp\begin{pmatrix} \log P(t_1 \mid OC) + \\ \sum_{i=2}^{n} \log[\lambda P(t_i \mid OC) + (1-\lambda)P(t_i \mid t_{i-1}, OC)] \end{pmatrix} \quad (5)$$

We observe that this formula penalizes verbose candidate answers. This can be alleviated by adding a brevity penalty, *BP*, which is inspired by machine translation evaluation (Papineni et al., 2001).

$$BP = \exp\left(\min\left(1 - \frac{L_{ref}}{L_A}, 1\right)\right) \quad (6)$$

where L_{ref} is a constant standing for the length of reference answer (i.e., centroid vector). L_A is the length of the candidate answer. By combining Equation (5) and (6), we get the final scoring function.

$$\begin{aligned} FinalScore(A) &= BP \times Score(A) \quad (7) \\ &= \exp\left(\min\left(1 - \frac{L_{ref}}{L_A}, 1\right)\right) \times \exp\begin{pmatrix} \log P(t_1 \mid OC) + \\ \sum_{i=2}^{n} \log[\lambda P(t_i \mid OC) + (1-\lambda)P(t_i \mid t_{i-1}, OC)] \end{pmatrix} \end{aligned}$$

3.3 Parameter estimation

In Equation (7), we need to estimate three parameters: $P(t_i|OC)$, $P(t_i|t_{i-1}, OC)$ and λ.

For $P(t_i|OC)$, $P(t_i|t_{i-1}, OC)$, maximum likelihood estimation (MLE) is employed.

$$P(t_i \mid OC) = \frac{Count_{OC}(t_i)}{N_{OC}} \quad (8)$$

$$P(t_i \mid t_{i-1}, OC) = \frac{Count_{OC}(t_{i-1}, t_i)}{Count_{OC}(t_{i-1})} \quad (9)$$

where $Count_{OC}(X)$ is the occurrences of the string *X* in the ordered centroid and N_{OC} stands for the total number of tokens in the ordered centroid.

For biterm language model, we use the above mentioned min-Adhoc approximation (Srikanth and Srihari, 2002).

$$P_{BT}(t_i \mid t_{i-1}, OC) = \frac{Count_{OC}(t_{i-1}, t_i) + Count_{OC}(t_i, t_{i-1})}{\min\{Count_{OC}(t_{i-1}), Count_{OC}(t_i)\}} \quad (10)$$

For unigram, we do not need smoothing because we only concern terms in the centroid vector. Recall that bigram and biterm probabilities have already been smoothed by interpolation.

The λ can be learned from a training corpus using an Expectation Maximization (EM) algorithm. Specifically, we estimate λ by maximizing the likelihood of all training instances, given the bigram or biterm model:

$$\begin{aligned} \lambda^* &= \arg\max_{\lambda} \sum_{j=1}^{|INS|} P(t_1^{(j)}...t_{l(j)}^{(j)} \mid OC) \quad (11) \\ &= \arg\max_{\lambda} \sum_{j=1}^{|INS|} \left\{ \sum_{i=2}^{l_j} \log\left[\lambda P(t_i^{(j)}) + (1-\lambda)P(t_i^{(j)} \mid t_{i-1}^{(j)})\right] \right\} \end{aligned}$$

BP and $P(t_1)$ are ignored because they do not affect λ. λ can be estimated using EM iterative procedure:

1) Initialize λ to a random estimate between 0 and 1, i.e., 0.5;
2) Update λ using:

$$\lambda^{(r+1)} = \frac{1}{|INS|} \times \sum_{j=1}^{|INS|} \frac{1}{l_j - 1} \sum_{i=2}^{l_j} \frac{\lambda^{(r)} P(t_i^{(j)})}{\lambda^{(r)} P(t_i^{(j)}) + (1-\lambda^{(r)})P(t_i^{(j)} \mid t_{i-1}^{(j)})} \quad (12)$$

where *INS* denotes all training instances and |*INS*| gives the number of training instances which is used as a normalization factor. l_j gives the number of tokens in the j^{th} instance in the training data;

3) Repeat Step 2 until λ converges.

We use the TREC 2004 test set[3] as our training data and we set λ as 0.4 for bigram model and 0.6 for biterm model according to the experimental results.

4 System Architecture

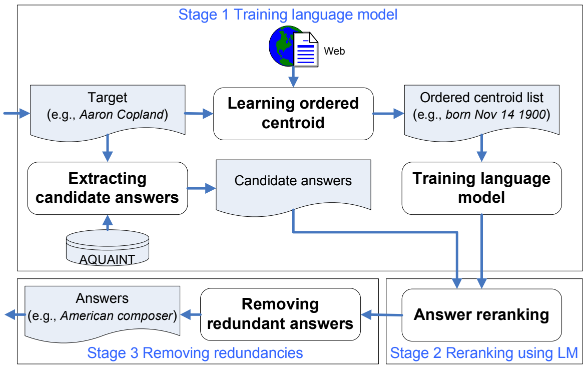
Stage 1 Training language model
Web
Target (e.g., *Aaron Copland*)
Learning ordered centroid
Ordered centroid list (e.g., *born Nov 14 1900*)
Extracting candidate answers
Candidate answers
Training language model
AQUAINT
Answers (e.g., *American composer*)
Removing redundant answers
Answer reranking
Stage 3 Removing redundancies
Stage 2 Reranking using LM

Figure 1. System architecture.

We propose a three-stage approach for answer extraction. It involves: 1) learning a language model from the web; 2) adopting the language model to rerank candidate answers; 3) removing redundancies. Figure 1 shows five main modules.

Learning ordered centroid:

1) Query expansion. Definitional questions are normally short (i.e., who is Bill Gates?). Query expansion is used to refine the query intention. First, reformulate query via simply adding clue words to the questions. i.e., for "Who is ...?" question, we add the word "biography"; and for "What is ...?" question, we add the word "is usually", "refers to", etc. We learn these clue words using the similar method proposed in (Ravichandran and Hovy, 2002). Second, query a web search engine (i.e., *Google*[4]) with reformulated query and learn top-*R* (we empirically set *R*=5) most frequent co-occurring terms with the target from returned snippets as query *expansion terms*;

2) Learning centroid vector (profile). We query *Google* again with the target and expanded terms learned in the previous step, download top-*N* (we empirically set *N*=500 based on the tradeoff between the snippet number and the time complexity) snippets, and split snippets into sentences. Then, we retain the generated sentences that contain the target, denoted as *W*. Finally, learn top-*M* (We empirically set *M*=350) most frequent co-occurring terms (stemmed) from *W* using Equation (15) (Cui et al., 2004) as the centroid vector.

$$Weight(t) = \frac{\log(Co(t,T)+1)}{\log(Count(t)+1)+\log(Count(T)+1)} \times idf(t) \qquad (13)$$

where *Co(t, T)* denotes the number of sentences in which *t* co-occurs with the target *T*, and *Count(t)* gives the number of sentences containing the word *t*. We also use the inverse document frequency of *t*, *idf(t)*[5], as a measurement of the global importance of the word;

3) Extracting ordered centroid. For each sentence in *W*, we retain the terms in the centroid vector as the ordered centroid list. Words not contained in the centroid vector will be treated as the "stop words" and ignored.

E.g., "Who is Aaron Copland?", the ordered centroid list is shown below(where italics are extracted and put in the ordered centroid list):

1. Today's Highlight in History: On *November 14*, *1900*, *Aaron Copland*, one of *America's* leading 20th century *composers*, was *born* in *New York City*. ⇒ *November 14 1900 Aaron Copland America composer born New York City*
2. ...

Extracting candidate answers: We extract candidates from AQUAINT corpus.

1) Querying AQUAINT corpus with the target and retrieve relevant documents;

2) Splitting documents into sentences and extracting the sentences containing the target. Here in order to improve recall, simple heuristics rules are used to handle the problem of coreference resolution. If a sentence is deemed to contain the target and its next sentence starts with "he", "she", "it", or "they", then the next sentence is retained.

Training language models: As mentioned above, we train language models using the obtained ordered centroid for each question.

Answer reranking: Once the language models and the candidate answers are ready for a given question, candidate answers are reranked based on the probabilities of the language models generating candidate answers.

Removing redundancies: Repetitive and similar candidate sentences will be removed. Given a reranked candidate answer set *CA*, redundancy removing is conducted as follows:

[3] The test data for TREC-13 includes 65 definition questions. NIST drops one in the official evaluation.

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

[5] We use the statistics from British National Corpus (BNC) site to approximate words' IDF, http://www.itri.brighton.ac.uk/~Adam.Kilgarriff/bnc-readme.html.

```
Step 1: Initially set the result $A=\{\}$, and get top $j=1$ element from $CA$ and then add it to $A$, $j=2$.
Step 2: Get the $j^{th}$ element from $CA$, denoted as $CA_j$. Compute cosine similarity between $CA_j$ and each element $i$ of $A$, which is expressed as $s_{ij}$. Then let $s_{ik}=max\{s_{1j}, s_{2j}, ..., s_{ij}\}$, if $s_{ik} < threshold$ (we set it to 0.75), then add $j$ to the set $A$.
Step 3: If length of $A$ exceeds a predefined threshold, exit; otherwise, $j=j+1$, go to Step 2.
```

Figure 2. Algorithm for removing redundancy.

5 Experiment & Evaluation

In order to get comparable evaluation, we apply our approach to TREC 2003 definitional QA task. More details will be shown in the following sections.

5.1 Experiment setup

5.1.1 Dataset

We employ the dataset from the TREC 2003 QA task. It includes the AQUAINT corpus of more than 1 million news articles from the *New York Times* (1998-2000), *Associated Press* (1998-2000), *Xinhua News Agency* (1996-2000) and 50 definitional question/answer pairs. In these 50 definitional questions, 30 are for people (e.g., *Aaron Copland*), 10 are for organizations (e.g., *Friends of the Earth*) and 10 are for other entities (e.g., *Quasars*). We employ *Lemur*[6] to retrieve relevant documents from the AQUAINT corpus. For each query, we return the top 500 documents.

5.1.2 Evaluation metrics

We adopt the evaluation metrics used in the TREC definitional QA task (Voorhees, 2003 and 2004). TREC provides a list of essential and acceptable nuggets for answering each question. We use these nuggets to assess our approach. During this progress, two human assessors examine how many essential and acceptable nuggets are covered in the returned answers. Every question is scored using nugget recall (*NR*) and an approximation to nugget precision (*NP*) based on answer length. The final score for a definition response is computed using F-Measure. In TREC 2003, the β parameter was set to 5 indicating that recall is 5 times as important as precision (Voorhees, 2003).

$$F(\beta=5)=\frac{5^2*NP*NR}{(5^2+1)NP+NR} \qquad (14)$$

in which,

$$NR=\frac{\#\ essential\ nuggets\ returned}{\#\ essential\ answer\ nuggets} \qquad (15)$$

$$NP=\begin{cases}1, & (length\ <\ allowance)\\ 1-\dfrac{(length\ \text{-}\ allowance)}{length}, & \text{(otherwise)}\end{cases} \qquad (16)$$

where *allowance = 100 * (# essential + # acceptable nuggets returned)* and *length = # non-white space characters in strings returned.*

5.1.3 Baseline system

We employ the *TFIDF* heuristics algorithm-based approach as our baseline system, in which the candidate answers and the centroid are treated as a bag of words.

$$weight_i = TF_i * IDF_i = TF_i * \ln\ {}^{N}\!/_{DF_i} \qquad (17)$$

where TF_i gives the occurrences of term *i*. DF_i[7] is the number of documents containing term *i*. *N* gives the total number of documents.

For comparison purpose, the unigram model is adopted and its scoring function is similar with Equation (7). The main difference is that we only concern unigram probability $P(t_i|OC)$ in unigram-based scoring function.

For all systems, we empirically set the threshold of answer length to 12 sentences for people targets (i.e., *Aaron Copland*), and 10 sentences for other targets (i.e., *Quasars*).

5.2 Performance evaluation

As the first evaluation, we assess the performance obtained by our language model method against the baseline system without query expansion (QE). The evaluation results are shown in Table 1.

	Average NR	Average NP	F(5)
Baseline (TFIDF)	0.469	0.221	0.432
Unigram	0.508 (+8.3%)	0.204 (-7.7%)	0.459 (+6.3%)
Bigram	0.554 (+18.1%)	0.234 (+5.9%)	0.505 (+16.9%)
Biterm	0.567 (+20.9%)	0.222 (+0.5%)	0.511 (+18.3%)

Table 1. Comparisons without QE.

From Table 1, it is easy to observe that the unigram, bigram and biterm-based approaches improve the F(5) by 6.3%, 16.9% and 18.3% against the baseline system respectively. At the same time, the bigram and biterm improves the

[6] A free IR tool, http://www.lemurproject.org/

[7] We also use British National Corpus (BNC) to estimate it.

F(5) by 10.0% and 11.3% against the unigram respectively. The unigram slightly outperform the baseline. We also notice that the biterm model improves slightly over the bigram model since it ignores the order of term-occurrence. This observation coincides with the experimental results of Srikanth and Srihari (2002). These results show that the bigram and biterm models outperform the VSM model and the unigram model dramatically. It is a clear indication that the language model which takes into account the term dependence among centroid vector is an effective way to rerank answers.

As mentioned above, QE is involved in our system. In the second evaluation, we assess the performance obtained by the language model method against the baseline system with QE. We list the evaluation results in Table 2.

	Average NR	Average NP	F(5)
Baseline (QE)	0.508	0.207	0.462
Unigram (QE)	0.518 (+2.0%)	0.223 (+7.7%)	0.472 (+2.2%)
Bigram (QE)	0.573 (+12.8%)	0.228 (+10.1%)	0.518 (+12.1%)
Biterm (QE)	0.582 (+14.6%)	0.240 (+15.9%)	0.531 (+14.9%)

Table 2. Comparisons with QE.

From Table 2, we observe that, with QE, the bigram and biterm still outperform the baseline system (VSM) significantly by 12.1% (*p*[8]*=0.03*) and 14.9% (*p=0.004*) in F(5). Furthermore, the bigram and biterm perform significantly better than the unigram by 9.7% (*p=0.07*) and 12.5% (*p=0.02*) in F(5) respectively. This indicates that the term dependence is effective in keeping improving the performance. It is easy to observe that the baseline is close to the unigram model since both two systems are based on the independent assumption. We also notice that the biterm model improves slightly over the bigram model. At the same time, all of the four systems improve the performance against the corresponding system without QE. The main reason is that the qualities of the centroid vector can be enhanced with QE. We are interested in the performance comparison with or without QE for each system. Through comparison it is found that the baseline system relies on QE more heavily than our approach does. With QE, the baseline system improves the performance by 6.9% and the language model approaches improve the performance by 2.8%, 2.6% and 3.9%, respectively.

F(5) performance comparison between the baseline model and the biterm model for each of 50 TREC questions is shown in Figure 3. QE is used in both the baseline system and the biterm system.

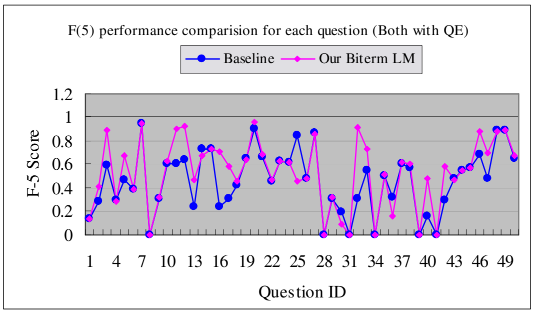

Figure 3. Biterm vs. Baseline.

We are also interested in the comparison with the systems in TREC 2003. The best F(5) score returned by our proposed approach is 0.531, which is close to the top 1 run in TREC 2003 (Voorhees, 2003). The F(5) score of the best system is 0.555, reported by BBN's system (Xu et al., 2003). In BBN's experiments, the centroid vector was learned from the human made external knowledge resources, such as encyclopedia and the web. Table 3 gives the comparison between our biterm model-based system with the BBN's run with different β values.

Run Tag	F(β) Score				
	β=1	β=2	β=3	β=4	β=5
BBN	0.310	0.423	0.493	0.532	0.555
Ours	0.288	0.382	0.470	0.509	0.531

Table 3. Comparison with BBN's run.

5.3 Case study

A positive example returned by our proposed approach is given below. For *Qid: 2304*: "Who is Niels Bohr?", the reference answers are given in Table 4 (only vital nuggets are listed):

vital	Danish
vital	Nuclear physicist
vital	Helped create atom bomb
vital	Nobel Prize winner

Table 4. Reference answers for question "Who is Niels Bohr?".

Answers returned by the baseline system and our proposed system are presented in Table 5.

System	Returned answers (Partly)
Baseline system	1. ..., Niels Bohr, the *great Danish scientist* 2. ...the German physicist Werner Heisenberg and the *Danish physicist*

[8] T-Test has been performed.

	Niels Bohr 3. ...took place between the *Danish physicist* Niels Bohr and his onetime protege, the German scientist ... 4. ... two *great physicists*, the Dane Niels Bohr and Werner Heisenberg ... 5. ...
Proposed system	1. ...physicist Werner Heisenberg travel to ... his colleague and old mentor, Niels Bohr, *the great Danish scientist* 2. ... two *great physicists*, the Dane Niels Bohr and Werner Heisen-berg ... 3. Today's Birthdays: ... *Danish nuclear physicist and Nobel Prize winner Niels Bohr (1885-1962)* 4. the *Danish atomic physicist*, and his German pupil, Werner Heisenberg, the author of the uncertainty principle 5. ...

Table 5. Baseline vs. our system for question "Who is Niels Bohr?".

From Table 5, it can be seen that the baseline system returned only one vital nugget: *Danish* (here we don't think that *physicist* is equal to *nuclear physicist* semantically). Our proposed system returned three vital nuggets: *Danish*, *Nuclear physicist*, and *Nobel Prize winner*. The answer sentence "Today's Birthdays: ... Danish nuclear physicist and Nobel Prize winner Niels Bohr (1885-1962)" contains more descriptive information for the question target "Niels Bohr" and is ranked 3rd in the top 12 answers in our proposed system.

5.4 Error analysis

Although we have shown that the language model-based approach significantly improves the system performance, there is still plenty of room for improvement.

1) Sparseness of search results derogated the learning of the ordered centroid: E.g.: *Qid 2348*: "What is the medical condition shingles?", in which we treat the words "medical condition shingles" as the question target. We found that few sentences contain the target "medical condition shingles". We found utilizing multiple search engines, such as *MSN*[9], *AltaVista*[10] might alleviate this problem. Besides, more effective smoothing techniques could be promising.
2) Term ambiguity: for some queries, the irrelated documents are returned. E.g., for *Qid 2267*: "Who is Alexander Pope?", all documents returned from the IR tool *Lemur* for this question are about "Pope John Paul II", not "Alexander Pope". This may be caused by the ambiguity of the word "Pope". In this case, term disambiguation or adding some constraint terms which are learned from the web to the query to the AQUAINT corpus might be helpful.

6 Conclusions and Future Work

In this paper, we presented a novel answer reranking method for definitional question. We use bigram and biterm language models to capture the term dependence. Our contribution can be summarized as follows:

1) Word dependence is explored from ordered centroid learned from snippets of a search engine;
2) Bigram and biterm models are presented to capture the term dependence and rerank candidate answers for definitional QA;
3) Evaluation results show that both bigram and biterm models outperform the VSM and unigram model significantly on TREC 2003 test set.

In our experiments, centroid words were learned from the returned snippets of a web search engine. In the future, we are interested in enhancing the centroid learning using human knowledge sources such as encyclopedia. In addition, we will explore new smoothing techniques to enhance the interpolation method in our current approach.

7 Acknowledgements

The authors are grateful to Dr. Cheng Niu, Yunbo Cao for their valuable suggestions on the draft of this paper. We are indebted to Shiqi Zhao, Shenghua Bao, Wei Yuan for their valuable discussions about this paper. We also thank Dwight for his assistance to polish the English. Thanks also go to anonymous reviewers whose comments have helped improve the final version of this paper.

References

E. Brill, J. Lin, M. Banko, S. Dumais and A. Ng. 2001. Data-Intensive Question Answering. In *Proceedings of the Tenth Text Retrieval Conference (TREC 2001)*, Gaithersburg, MD, pp. 183-189.

S. Blair-Goldensohn, K.R. McKeown and A. Hazen Schlaikjer. 2003. A Hybrid Approach for QA Track Definitional Questions. In *Proceedings of the Tenth Text Retrieval Conference (TREC 2003)*, pp. 336-343.

[9] http://www.msn.com
[10] http://www.altavista.com

S. F. Chen and J. T. Goodman. 1996. An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th Annual Meeting of the ACL*, pp. 310-318.

Hang Cui, Min-Yen Kan and Tat-Seng Chua. 2004. Unsupervised Learning of Soft Patterns for Definitional Question Answering. In *Proceedings of the Thirteenth World Wide Web conference (WWW 2004)*, New York, pp. 90-99.

Guihong Cao, Jian-Yun Nie, and Jing Bai. 2005. Integrating Word Relationships into Language Models. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development of Information Retrieval (SIGIR 2005)*, Salvador, Brazil.

Jianfeng Gao, Jian-Yun Nie, Guangyuan Wu and Guihong Cao. 2004. Dependence language model for information retrieval. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development of Information Retrieval (SIGIR 2004)*, Sheffield, UK.

Chin-Yew Lin. 2002. The Effectiveness of Dictionary and Web-Based Answer Reranking. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002)*, Taipei, Taiwan.

Lafferty, J. and Zhai, C. 2001. Document language models, query models, and risk minimization for information retrieval. In W.B. Croft, D.J. Harper, D.H. Kraft, & J. Zobel (Eds.), In *Proceedings of the 24th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, New Orleans, Louisiana, New York, pp.111-119.

Magnini, B., Negri, M., Prevete, R., and Tanev, H. 2002. Is It the Right Answer? Exploiting Web Redundancy for Answer Validation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL-2002)*, Philadelphia, PA.

Miller, D., Leek, T., and Schwartz, R. 1999. A hidden Markov model information retrieval system. In *Proceedings of the 22nd Annual International ACM SIGIR Conference*, pp. 214-221.

K. Papineni, S. Roukos, T. Ward, and W.J. Zhu. 2001. Bleu: a Method for Automatic Evaluation of Machine Translation. *IBM Research Report rc22176 (w0109022)*, Thomas J. Watson Research Center.

Ponte, J., and Croft, W.B. 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*, New York, pp.275-281.

J. Prager, D. Radev, and K. Czuba. 2001. Answering what-is questions by virtual annotation. In *Proceedings of the Human Language Technology Conference* (HLT 2001), San Diego, CA.

Deepak Ravichandran and Eduard Hovy. 2002. Learning Surface Text Patterns for a Question Answering System. In *Proceedings of the 40th Annual Meeting of the ACL*, pp. 41-47.

Song, F., and Croft, W.B. 1999. A general language model for information retrieval. In *Proceedings of the 22nd Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, New York, pp.279-280.

Srikanth, M. and Srihari, R. 2002. Biterm language models for document retrieval. In *Proceedings of the 2002 ACM SIGIR Conference on Research and Development in Information Retrieval*, Tampere, Finland.

Ellen M. Voorhees. 2002. Overview of the TREC 2002 question answering track. In *Proceedings of the Eleventh Text REtrieval Conference (TREC 2002).*

Ellen M. Voorhees. 2003. Overview of the TREC 2003 question answering track. In *Proceedings of the Twelfth Text REtrieval Conference (TREC 2003).*

Ellen M. Voorhees. 2004. Overview of the TREC 2004 question answering track. In *Proceedings of the Twelfth Text REtrieval Conference (TREC 2004).*

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

Jinxi Xu, Ana Licuanan, and Ralph Weischedel. 2003. TREC2003 QA at BBN: Answering definitional questions. In *Proceedings of the Twelfth Text REtrieval Conference (TREC 2003).*

Jun Xu, Yunbo Cao, Hang Li and Min Zhao. 2005. Ranking Definitions with Supervised Learning Methods. In *Proceedings of 14th International World Wide Web Conference (WWW 2005), Industrial and Practical Experience Track*, Chiba, Japan, pp.811-819.

Zhang D. and Lee WS. 2003. A Language Modeling Approach to Passage Question Answering. In *Proceedings of The 12th Text Retrieval Conference (TREC2003)*, NIST, Gaithersburg.

Zhai, C, and Lafferty, J. 2001. A Study of Smoothing Methods for Language Models Applied to Information Retrieval. In *Proceedings of the 2001 ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 334-342.