Trained Trigger Language Model for Sentence Retrieval in QA: Bridging the Vocabulary Gap

Saeedeh Momtazi*, Dietrich Klakow
Spoken Language Systems
Saarland University, Saarbrücken, Germany
{saeedeh.momtazi,dietrich.klakow}@lsv.uni-saarland.de

ABSTRACT

We propose a novel language model for sentence retrieval in Question Answering (QA) systems called *trained trigger language model*. This model addresses the word mismatch problem in information retrieval. The proposed model captures pairs of trigger and target words while training on a large corpus. The word pairs are extracted based on both unsupervised and supervised approaches while different notions of triggering are used. In addition, we study the impact of corpus size and domain for a supervised model. All notions of the trained trigger model are finally used in a language model-based sentence retrieval framework. Our experiments on TREC QA collection verify that the proposed model significantly improves the sentence retrieval performance compared to the state-of-the-art translation model and class model which address the same problem.

Categories and Subject Descriptors: H.3.3 [Information Storage and Retrieval]:Information Search and Retrieval

General Terms: Theory, Algorithm, Experimentation

Keywords: sentence retrieval, language modeling, triggering, question answering

1. INTRODUCTION

Sentence retrieval plays an important role in QA systems. It aims to find small segments of text that contain an exact answer to the users' questions. The brevity of sentences compared to documents exacerbates the usual term mismatch problem which should taken into consideration while designing a sentence retrieval engine.

The word unigram model is the most common approach used in the majority of information retrieval literature. When estimating word unigrams, the model only considers the exact literal words present in the query. Since no relationship between words is considered by this model, it will fail to retrieve other relevant information. Different techniques such as translation model [1] and class model [7] proposed to overcome the term mismatch problem. In this paper, we introduce a new model called *trained trigger language model* which uses another approach to this aim. In the proposed model, we assume that by training a language model on a large corpus, we can find pairs of trigger and target words; so that in the retrieval step, if the trigger word appears in an input question and the target word appears in a sentence, then we can say there is a relation between the question and the sentence even though they share no or few terms.

The structure of the paper is as follows. Section 2 introduces the proposed trained trigger language model. Different notions of triggering are described in Section 3. In Section 4, we show how the proposed model can be integrated with the exact matching method to improve the system performance. In Section 5, the experimental results are presented. Finally, Section 6 concludes the paper.

2. THE TRIGGER MODEL

In the word unigram model, the probability of generating the query Q given the observation of the sentence model S is estimated by:

$$P(Q|S) = \prod_{i=1}^{M} P(q_i|S) \quad (1)$$

In this model, for each sentence in the search space a separate language model is trained to estimate parameters; and then using the maximum likelihood estimation, $P(q_i|S)$ is calculated:

$$P_{\text{word}}(q_i|S) = \frac{f_S(q_i)}{\sum_w f_S(w)} \quad (2)$$

where $f_S(q_i)$ is the number of times the query term q_i appears in the sentence S and w denotes vocabulary words. As a result, based on the maximum likelihood estimation, $P(Q|S)$ for word unigram is defined as:

$$P_{\text{word}}(Q|S) = \prod_{i=1}^{M} \frac{f_S(q_i)}{\sum_w f_S(w)} \quad (3)$$

In contrast, in the trained trigger language model, a single model is trained on a large corpus first. Then, the trained model is being used for all of the sentences to be retrieved. The trained model is represented by $f(w, w')$, where $f(w, w')$ is the number of times the word w triggers the target w'. There are different notions of triggering that can be used to find pairs of trigger and target words. The notions of triggering used in our experiments are discussed in Section 3. Having a trained model based on trigger and target words, for each word in the query and each word in

*Saeedeh Momtazi's present affiliation is Hasso-Plattner-Institut, Potsdam, Germany

CIKM'11, October 24–28, 2011, Glasgow, Scotland, UK.

the sentence, the probability of generating the query term given the sentence term is calculated as follows:

$$P_{\text{trigger}}(q_i|s_j) = \frac{f_C(q_i, s_j)}{\sum_{q_i} f_C(q_i, s_j)} \quad (4)$$

where s_j is the j^{th} term in the sentence and $f_C(q_i, s_j)$ is the number of times the query term q_i triggers the sentence word s_j based on the training corpus C. Using the proposed triggering method for calculating $P(q_i|s_j)$, the likelihood of generating query term q_i given the sentence S is defined as:

$$P_{\text{trigger}}(q_i|S) = \frac{1}{N} \sum_{j=1}^{N} P_{\text{trigger}}(q_i|s_j) \quad (5)$$

where N is the sentence length. As a result, $P(Q|S)$ based on the trained trigger language model is defined as follows:

$$P_{\text{trigger}}(Q|S) = (\frac{1}{N})^M \prod_{i=1}^{M} \sum_{j=1}^{N} \frac{f_C(q_i, s_j)}{\sum_{q_i} f_C(q_i, s_j)} \quad (6)$$

Similar to the basic word unigram model, a smoothing technique is required to deal with zero probabilities of unseen words. Any of the smoothing methods proposed for the word unigram model can also be applied to this model.

Comparing the time complexity of the proposed model with the normal unigram model. For each input query, the word unigram model is running in $O(KM)$ time, where K is the number of sentences in the search space and M is the query length. In the trained trigger language model, we have an additional factor which is the number of words in the sentence to be retrieved. As a result, the running time of the trigger model is $O(KMN)$. Although the time complexity of our model is higher than the word unigram model, it is still identical to the translation model.

3. TYPES OF TRIGGERING

3.1 Inside Sentence Triggering

This model assumes that there is a relation between the words which appear in the same sentences. To use *inside sentence triggering*, the model should be trained on a large unannotated corpus while considering that each word in a sentence triggers all other words in the same sentence.

Using this model, we can retrieve sentences which do not share many terms with the query, but their terms frequently co-occur with the query terms in the same sentences of the training corpus. To be more precise, consider the sample sentences in Figure 1 and the following question and the answer sentence:
Q: *"Who invented the automobile?"*
A: *"Nicolas Joseph Cugnot invented the first self propelled mechanical vehicle."*
Using the question as a query in word unigram, we only have one common term *"invent"* between the query and the sentence which is not enough to rank this correct sentence highly, since there are many sentences in the search space which contain the word *"invent"*, such as:
– *"Edison invented the first commercially practical light."*
– *"Alexander Graham Bell of Scotland is the inventor of the first practical telephone."*
Inside sentence triggering, however, gives a higher score to the correct sentence; because it knows the word *"automobile"* triggers the target word *"vehicle"*, since they frequently co-occur in the same sentences of the training corpus.

S:	*"The word automobile means a vehicle that moves itself."*
S:	*"An automobile includes at least two seats located one behind the other and attachable to a vehicle floor."*

Figure 1: Inside sentence triggering samples

S1:	*"Wembley Stadium was built by Australian company Brookfield Multiplex."*
S2:	*"The stadium was scheduled to open in 2006."*
S1:	*"The structure of Eiffel Tower was built between 1887 and 1889."*
S2:	*"The tower was inaugurated on 31 March 1889, and opened on 6 May."*

Figure 2: Across sentence triggering samples

3.2 Across Sentence Triggering

Across sentence triggering uses a wider context than inside sentence triggering: it considers that each word in a sentence triggers all words in the next sentence of the training corpus. Because two adjacent sentences normally talk about the same topic and it is very likely that different words with the same concept are used in two consecutive sentences.

As an example, consider a corpus which contains the adjacent sentences presented in Figure 2. Using such a training corpus, we can find that the word *"build"* triggers the word *"open"* which is in the next sentence. As a result, this trained trigger model can retrieve the correct answer for the following question:
Q: *"Where was the first McDonald's built?"*
A: *"Two brothers from Manchester, Dick and Mac McDonald, open the first McDonald's in California."*
It is clear that word unigram is not able to rank the correct sentence highly; because there are many sentences in the search space talking about *"first McDonald's"*, but non-relevant to the question, such as:
– *"The site of the first McDonald's to be franchised by Ray Kroc is now a museum in Des Plaines, Illinois."*
– *"The first McDonald's TV commercial was a pretty low-budget affair."*

3.3 Question and Answer Pair Triggering

Another notion of triggering derives from using more supervision when training the model. This notion of triggering requires a set of question and answer pairs as a training corpus. In *question and answer pair triggering*, each word in the question triggers all words which appear in the answer sentence. Again this model can retrieve sentences which share no or few words with the question but their terms frequently co-occur with question terms in question and answer pairs that are used to train the model.

As an example, consider the following question and its correct answer sentence.
Q: *"How high is Everest?"*
A: *"Everest is 29,029 feet."*
Similar to the previous examples, the above question and answer share a very limited number of terms, in this case only the term *"Everest"*. As a result, it is very unlikely that the word unigram model ranks the correct answer highly. Because there are many non-relevant sentences in the search space which contain the word *"Everest"*, such as:
– *"Everest is located in Nepal."*

Q:	*"How high is Mount Hood?"*
A:	*"Mount Hood is in the Cascade Mountain range and is 11,245 feet."*
Q:	*"How high is Pikes peak?"*
A:	*"Pikes peak, Colorado At 14,110 feet, altitude sickness is a consideration when driving up this mountain."*

Figure 3: QA pair triggering samples

– *"Everest has two main climbing routes."*
However, the question and answer pair triggering model gives a higher score to the correct sentence because the model knows that in a large portion of questions that contain the word *"high"*, the term *"feet"* appears in the answer, as shown in Figure 3. As a result, in the trained model, the word *"high"* triggers the target word *"feet"*.

4. THE INTERPOLATION MODEL

As mentioned, the proposed trained trigger language model aims to capture relationships between words and use these relations in the retrieval step. As a result, the model is able to find more relevant sentences and increase system recall reasonably. This method may decrease the system precision, however, by retrieving more non-relevant sentences. To avoid this problem, it is essential to use word unigram in combination with the other notions of triggering. To this end, we use the linear interpolation [3] of our proposed triggering model and the baseline exact matching model, thereby benefiting from the advantages of both models and avoiding their disadvantages.

To use the interpolation of word unigram and triggering, the probability of generating q_i is computed from Equation 2 and 5 and interpolated by weighting parameter λ which is tuned using the held-out data:

$$P(Q|S) = \prod_{i=1}^{M} [\lambda \; P_{\text{trigger}}(q_i|S) + (1-\lambda) \; P_{\text{word}}(q_i|S)] \quad (7)$$

5. EXPERIMENTAL RESULTS

5.1 The Dataset

To evaluate our methods, we used the set of factoid questions from TREC[1] 2005 and 2006 QA track. The set of questions from 2006 was used as test data, while the 2005 question set was used as held-out data to study the smoothing parameters and interpolation weights. The TREC QA 2006 contains 75 question-series that each of them focuses on a target. The series contained a total of 403 factoid questions. For sentence-level relevance judgments, we used the *Question Answer Sentence Pair (QASP)* corpus [4].

For each of the proposed notions of triggering, we need to estimate the corresponding parameters during a training phase using a training corpus. For a part of experiments which applies inside or across sentence triggering, a large unannotated corpus is required to train the models in an unsupervised fashion. For this purpose, we used *AQUAINT1* corpus[2]. For the other part of experiments, which uses question and answer pair triggering, we need to train our models in a supervised fashion in which a corpus of question and answer sentence pairs is required. As for the supervised model, we used the QASP corpus[3] and the *Yahoo! Answers Comprehensive Questions and Answers* corpus[4]. Since the questions of TREC QA 2005 and 2006 of the QASP corpus are used as held-out and test data respectively, we only used the questions of 2002-2004 of the corpus for training, so that there is no overlap between the train and evaluation sets.

Table 1: Performance of the trained trigger language model with different notions of triggering interpolated with word unigram. (TTLM stands for Trained Trigger Language Model)

Model	MAP	MRR
Word Unigram	0.3701	0.5047
Unigram + Inside Sentence TTLM	0.4351	0.5572
Unigram + Across Sentence TTLM	0.4381	0.5631
Unigram + QA Pair TTLM (QASP)	0.4208	0.5492
Unigram + QA Pair TTLM (Yahoo)	0.4371	0.5654

5.2 Results and Discussion

To evaluate different notions of triggering we interpolated them with the word unigram model which consider the exact matching between query terms and sentence terms. As we mentioned in Section 4, the proposed triggering types only consider contextual information and they do not give a priority to the sentences that contain the query words. As a result, each of the triggering types are interpolated with word unigram as baseline. Table 1 presents the results of our experiments, interpolating different notions of triggering with word unigram. For all experiments, we used Bayesian smoothing with Dirichlet prior. The results show that using any of the contextual information in addition to the exact matching model significantly improves sentence retrieval performance compared to the word unigram model in which all the improvements are highly statistically significant ($p < 0.01$) according to the 2-tailed paired t-test.

The results verify that both exact matching and triggering models play important roles in retrieving answer sentences. When the exact matching model avoids returning non-relevant sentences, it fails to find most of the relevant sentences which share few words with the question. The triggering model, on the other hand, is able to capture more relevant sentences and enhance retrieval performance when it is integrated with the exact matching model. In addition to the above observation, comparing different notions of triggering, the following points should be highlighted:

- Comparing the results of the unsupervised notions triggering, we can see that inside sentence triggering and across sentence triggering perform close to each other. It might be due to the similar word relationship that they capture.
- Comparing the results of question and answer triggering based on different corpora, we can see that the model trained on the Yahoo! Answers corpus outperforms the model trained on the QASP corpus. This observation indicates that even though the Yahoo! Answers corpus is from a different and much broader domain than the QASP corpus and it includes more noisier data than QASP, it significantly improves the system performance due to sheer size. In addition, this result shows that although automatic QA systems are

[1] `http://trec.nist.gov`
[2] See catalog number LDC2002T31 at `http://www.ldc.upenn.edu/Catalog`
[3] `http://homepages.inf.ed.ac.uk/s0570760/data`
[4] `http://webscope.sandbox.yahoo.com`

Table 2: Comparing the results of trained trigger model with the translation model and class model. Significant differences with the translation model and class model are marked by t and c respectively.

Model	Estimation	MAP	MRR
Unigram	-	0.3701	0.5047
Translation	Mutual Information	0.3927	0.5348
Class-based	Word Bigram	0.4174	0.5527
Trigger	Inside Sentence	0.4351^{**t}_{**c}	0.5572
Trigger	Across Sentence	0.4381^{**t}_{**c}	0.5631^{*t}

considered a different topic from community QA forums, the later ones can be a great help to improve the performance of the former ones.

- Overall, the question and answer pair triggering based on the Yahoo! Answers corpus, which achieved the best performance in supervised training, performs similar to the unsupervised notions of triggering. This observation shows that even without using an annotated corpus we can achieve significant improvements that are comparable with the supervised methods.

As we saw, using the trained trigger language model significantly improves sentence retrieval performance compared to the standard word unigram model which does not consider such contextual information. In another step of our experiments, we compared our proposed trained trigger model with the state-of-the-art language modeling techniques which address the same problem of information retrieval. To this aim, we implemented the translation model [1] in which mutual information is used for estimating the translation probability [5]. As shown by Karimzadehgan and Zhai [5], normalized mutual information between word pairs is the best estimation of the statistical translation model and outperforms the original translation model [1] which is estimated based on synthetic queries. Following Karimzadehgan and Zhai [5], we estimated translation models using normalized mutual information between word pairs and regularized the models by self-translation probabilities. To have a reasonable comparison between the translation model and our trained trigger model, we used the AQUAINT1 corpus for calculating mutual information.

In addition to the translation model, we also implemented the class-based model using cluster of words [7]. The class-based model is also aims to relax the exact matching assumption made by the word unigram model. As shown by Momtazi et al. [6] using bigram statistics of terms in a corpus is the best estimation for word clustering. Following Momtazi and Klakow [7], we used Brown word clustering algorithm [2] while using word bigram statistics to cluster lexical items. To estimate word bigrams, we again used the AQUAINT1 corpus. Therefore, we have an identical corpus for all three models. Table 2 presents the results of the translation and the class-based models. To have the models being comparable, we repeated parts of our results from Table 1. We only represented the triggering models which use the same corpus (AQUAINT1) for training. Results are marked as significant* ($p < 0.05$), or highly significant** ($p < 0.01$), or neither according to 2-tailed paired t-test.

As shown by the tabulated results, the proposed trained trigger language model significantly outperforms the translation and the class models. Although these two models also address the word mismatch problem and outperform the standard word unigram model, both of them underperform our trained trigger language model. These results verify the superiority of the trained trigger language model in capturing word relationships compared to the translation and the class models.

6. CONCLUDING REMARKS

In this paper, we proposed a new language model for sentence-level retrieval which is able to capture term relationships by finding pairs of trigger and target words based on their co-occurrence. AQUAINT1 as a large corpus of raw text is used to train our trigger language model in an unsupervised fashion while two different types of triggering have been considered: inside sentence triggering which uses the information of co-occurring words in the same sentences, and across sentence triggering which crosses sentence boundaries and finds relation between terms that appear in two consecutive sentences. The QASP corpus from TREC QA track and the Yahoo! Answers corpus from Yahoo QA forum are used to train our model in a supervised fashion called question and answer pair triggering.

The results indicated that our trained trigger language model outperforms the state-of-the-art translation model and class model. The proposed model achieved 2.63% *absolute* improvement in MAP compared to the class model based on Brown word clustering algorithm [7]. The model also achieved 5.10% *absolute* improvement compared to the translation model based on mutual information [5].

7. ACKNOWLEDGMENTS

The authors would like to thank Yahoo! Labs Webscope for their Yahoo! Answers Comprehensive Questions and Answers corpus. Saeedeh Momtazi is funded by the German research foundation DFG through the International Research Training Group (IRTG 715).

8. REFERENCES

[1] A. Berger and J. Lafferty. Information Retrieval as Statistical Translation. In *ACM SIGIR*, pages 222–229. ACM, 1999.

[2] P. Brown, V. Pietra, P. Souza, J. Lai, and R. Mercer. Class-based N-gram Models of Natural Language. *Computational Linguistics*, 18(4):467–479, 1992.

[3] F. Jelinek and R. Mercer. Interpolated Estimation of Markov Source Parameters from Sparse Data. In *Proceedings of an International Workshop on Pattern Recognition in Practice*, 1989.

[4] M. Kaisser and J. Lowe. Creating a Research Collection of Question Answer Sentence Pairs with Amazon's Mechanical Turk. In *LREC*, 2008.

[5] M. Karimzadehgan and C. Zhai. Estimation of Statistical Translation Models based on Mutual Information for Ad-hoc Information Retrieval. In *ACM SIGIR*, pages 323–330. ACM, 2010.

[6] S. Momtazi, S. Khudanpur, and D. Klakow. A Comparative Study of Word Co-occurrence for Term Clustering in Language Model-based Sentence Retrieval. In *NAACL-HLT*. Association for Computational Linguistics, 2010.

[7] S. Momtazi and D. Klakow. A Word Clustering Approach for Language Model-based Sentence Retrieval in Question Answering Systems. In *ACM CIKM*, pages 1911–1914. ACM, 2009.