Computational Linguistics and Chinese Language Processing
Vol. 22, No. 2, December 2017, pp. 17-30

Question Retrieval with Distributed Representations and Participant Reputation in Community Question Answering

Sam Weng[*,+], Chun-Kai Wu[#], Yu-Chun Wang[ǂ] and

Richard Tzong-Han Tsai[+]

Abstract

In recent years, community-based question and answer (CQA) sites have grown rapidly in number and size. These sites represent a valuable source of online knowledge; however, they often suffer from the problem of duplicate questions. The task of question retrieval (QR) aims to find previously answered semantically similar questions in CQA archives. Nevertheless, synony- mous lexical variations pose a big challenge for question retrieval. Some QR approaches address this issue by calculating the probability of correlation between new questions and archived questions. Much recent research has also focused on surface string similarity among questions. In this paper, we propose a method that first builds a continuous bag-of-words (CBoW) model with data from Asus's Republic of Gamers (ROG) forum and then determines the similarity between a given new question and the Q&As in our database. Unlike most other methods, we calculate the similarity between the given question and the archived questions and descriptions separately with two different features. In addition, we factor user reputation into our ranking model. Our experimental results on the ROG forum dataset show that our CBoW model with reputation features outperforms other top methods.

[*] AsusTek Computer Inc., Taiwan

[+] Department of Computer Science and Information Engineering, National Central University, Taiwan
E-mail: thtsai@csie.ncu.edu.tw

[#] Department of Computer Science, National Tsing Hua University, Taiwan

[ǂ] Department of Buddhist Studies, Dharma Drum Institute of Liberal Arts, Taiwan
E-mail: ycwang@dila.edu.tw

The authors for correspondence are Yu-Chun Wang and Richard Tzong-Han Tsai.

Keywords: Question Retrieval, QR, Community-based Question and Answer, CQA

1. Introduction

Over the past decade, there has been a proliferation of online user forums and community question and answer (CQA) sites such as Yahoo! Answers, Quora and Baidu Zhidao. These sites provide a platform for people to discuss questions and solutions to common problems in a wide variety of fields, and they have generated massive amounts of data. Question retrieval (QR) is the task of sorting through this data to find previously answered questions in CQA archives that are similar to a user's current query.

A major challenge for QR is matching the user's query to its lexical variations in the dataset. For example, the system needs to be able to estimate the similarity between synonymous keywords and phrases like "blue screen", "BSOD" and "system crash" that may all refer to the same event. Four main approaches have been proposed to deal with synonyms, such as language model information retrieval (LMIR) (Ponte & Croft, 1998; Song & Croft, 1999; Zhai & Lafferty, 2004), language model with category smoothing (LMC) (Cao, Cong, Cui, Jensen & Zhang, 2009), translation-based language modeling (TBLM) (Xue, Jeon & Croft, 2008), and distributed-representation- based language modeling (DRLM) (Mikolov, Chen, Corrado & Dean, 2013).

Language model information retrieval (LMIR) estimates probabilities of word sequences between query and candidate question. Another approach, language model with category smoothing (LMC), represents each question category as a dimension in a vector space. In both LMIR and LMC, words are represented as indexed in a vocabulary, and similarity of words is ignored. Still another approach is translation-based language modeling (TBLM), which uses QA pairs to learn semantically related words to improve traditional IR models. The basic assumption is that QA pairs are parallel texts and the relationship of words can be established through word-to-word translation probability. In practical use, however, TBLM may take too long to learn a translation table. Finally, distributed-representation-based language modeling (DRLM) uses distributed representations of data to replace the word-to-word translation probability in TBLM with the probability calculated using word2vector. DRLM further combines the similarity of a word vector and a category vector as the final retrieval model.

In this paper, we propose a method that first builds a continuous bag-of-word (CBoW) model with data from the Asus Republic of Gamers (ROG) forum and then determines the similarity between a given new question and the Q&As in our database. Unlike most other studies, we calculate the similarity between the given question and the archived questions and descriptions separately with two different features. In addition, we factor user reputation into our ranking model. Our experimental results on ROG forum dataset show that our CBoW model with reputation features outperforms other top methods.

2. Related Work

In this section, we offer an overview of existing community-based question retrieval models.

2.1 Language Model for Information Retrieval (LMIR)

In recent years, LMIRs and their extensions have been widely used for question retrieval on community Q&A data (Ponte & Croft, 1998; Song & Croft, 1999; Zhai & Lafferty, 2004). It is a statistical way to estimate the probabilities of word sequences between query and candidate questions. Measurement can be expensive, since sentences can be arbitrarily long and the size of a corpus needs to be very large. In practice, the statistical language models are often approximated by N-gram models. The unigram model makes a strong assumption that every single word occurs independently, and consequently, the probability of a word sequence becomes the product of probabilities of the individual words. The bigram and trigram models take the local context into consideration. As for the bigram, the probability of a new word depends on the probability of the previous word. While for a trigram, the probability of a new word depends on the probabilities of the previous two words. The basic language modeling approach (unigram language model) has performed quite well empirically in several information retrieval tasks (Ponte & Croft, 1998; Song & Croft, 1999; Zhai & Lafferty, 2004) and has also performed quite well in question search (Zhai & Lafferty, 2004). The basic idea is to estimate a language model for each query and then rank candidates by likelihood of the query according to the estimated model. Given a query q and a candidate Q, the ranking function is as follows:

$$P(q|Q) = \sum_{w\in q}(1-\lambda)P_{ml}(w|Q) + \lambda\, P_{ml}(w|C) \qquad (1)$$

Where q is the queried question and w is a word in it. Q is an archived question and C is whole data collection. $P_{ml}(w|Q)$ presents the maximum likelihood estimated of w in Q. $P_{ml}(w|C)$ is a smoothing item which is calculated as the maximum likelihood in a large corpus C. The smoothing item avoids zero probability when the words appear in q but not in Q. λ is a parameter ranging from 0.0 to 1.0.

2.2 Language Model with Category Smoothing (LMC)

On most community question and answer sites, each question belongs to one or several categories by askers' tagging actions. Category information of archived questions is utilized such that category-specific frequent words will play an important role in comparing the relevancy of archived questions across categories to a query (Cao *et al.*, 2009). Instead of finding patterns among individual words, a language model may be designed to discover relationships among word groupings or categories. This idea can be realized as follows: the category language model is first smoothed with the whole question collection, and then the question language model is smoothed with the category model. To utilize category information, LMC expands LMIR with a new smoothing value estimated from questions under the same category. Given a user search question q and a candidate Q, LMC is described as follows:

$$P(q|Q) = \sum_{w\in q}(1-\lambda)P_{ml}(w|Q) + \lambda P_s(w|C) \tag{2}$$

$$P_s(w|Q) = (1-\beta)P_{ml}(w|C) + \beta P_{ml}(w|Cat(Q)) \tag{3}$$

In this, w means a word in the query question q, and λ and β are two different smoothing parameters. $Cat(Q)$ denotes the category of the candidate question Q, which is usually a root or a leaf category in the category hierarchy of a community Q&A site. $P_{ml}(w|Q)$ is the maximum likelihood estimate of w in Q. $P_{ml}(w|Cat(Q))$ means the maximum likelihood estimate for w in $Cat(Q)$. $P_{ml}(w|C)$ represents the maximum likelihood estimate of w in a large corpus C. λ and β both range from 0.0 to 1.0.

2.3 Translation Model (TM)

The idea of statistical machine translation was first introduced by Warren Weaver in 1949. The basic idea is based on a string-to-string noisy channel model. The channel converts a sequence of words from one language (such as Spanish) into another (such as Chinese) according to the probability distribution. The channel operations are movements, duplications and translations, applied to each word independently. The movement is conditioned only on word classes and positions in the string, and the duplication and translation are conditioned only on the word identity. Statistical translation models were initially word-based (Models 1-5 from IBM hid- den Markov model from Stephan Voge and Model 6 from Franz-Joseph Och), but significant advances were made with the introduction of phrase-based models. In word-based translation, the fundamental unit of translation is a word in natural language. Previous work (Berger, Caruana, Cohn, Freitag & Mittal, 2000; Xue *et al.*, 2008) consistently reported that word-based translation models yielded better performance than traditional methods (such as language model) for question retrieval. These models exploited a modeling framework. The ranking function can be written as follows:

$$P(q|Q) = \prod_{w\in Q}(1-\lambda)P_{tr}(w|Q) + \lambda P_s(w|Q) \tag{4}$$

$$P_{tr}(w|Q) = \sum_{t\in Q} P(w|t)P_{ml}(t|Q) \tag{5}$$

Where $P_{ml}(w|C)$ and $P_{ml}(t|Q)$ can be estimated similarly as in the language model above, $P(w|t)$ denotes the translation probability that w is a translation of t, and it is assumed that the probability of self-translation is 1, meaning that $P(w|t) = 1$.

2.4 Translation-Based Language Model (TBLM)

A recent approach to question retrieval is the translation-based language model (TBLM) (Xue *et al*., 2008), which combines LMIR and TM. It has been shown that this model achieves better performance than both LMIR and TM. TBLM uses word-to-word translation probabilities estimated from questions to find semantically similar questions. The TBLM ranking score is computed as follows:

$$P(q|Q) = \prod_{w \in q}(1-\lambda)P_{mx}(w|Q) + \lambda P_{ml}(w|C) \tag{6}$$

$$P_{mx}(w|Q) = (1-\beta)P_{ml}(w|Q) + \beta P_{tr}(w|Q) \tag{7}$$

$$P_{tr}(w|Q) = \sum_{v \in Q} P_{tp}(w|v)P_{ml}(v|Q) \tag{8}$$

Where λ and β are two different smoothing parameters controlling the translation component's impact, and $P_{tp}(w|v)$ is the translation probability from word w in query question to word v in historical candidate question Q. The difference between TM and TBLM is that TBLM calculates with one extra element $(1-\beta)P_{ml}(w|Q)$.

2.5 Word Embedding Learning

A distributed representation (word embedding) (Mikolov *et al*., 2013) stores the same contextual information in a low-dimensional vector. Every word is now represented by a *D* dimensional vector, where *D* is a relatively small number (usually between 50 and 1000). Each dimension of the embedding corresponds to a semantic or grammatical attribute of the words. The hope is that similar words get to closer to each other in that space. In place of counting word co-occurrences, the vectors can be learned.

The basic algorithm starts from a random vector for each word in the vocabulary. It then crosses a large corpus, and at each step, observes a target word and its context. The vector of the target word and the context word will be updated to bring them close together in the vector space, thus increasing the similarity between them. Other vectors will be updated to become more distant from the target word. After the processing, the vectors become meaningful, representing similar words with similar vectors. The advantage of word embedding is that it allows the model to generalize sequences that do not appear in the set of training data but are similar in terms of their features.

3. Approach

In this section, we will describe the proposed approach consisting of three parts: (1) word embedding learning: given a forum data collection, questions are treated as basic units. Each word in a question is transformed into a word vector. (2) score generation: once the word vectors are learned, question retrieval can be performed by calculating the similarity between a query question and a candidate question. (3) utilizing reputation information: we enhance the ranking function by introducing reputation points of each archived question's participants.

3.1 Word2vec

Word2vec is an open-source software program that was created by a team of researchers led by Tomas Mikolov at Google[1]. It is a group of related models that are used to produce word embeddings. This tool provides an efficient implementation of the continuous bag-of-words (CBoW) and skip-gram architectures for computing vector representation of words. Using the CBoW architecture, the model predicts the current word by using the context words. The order of context words does not influence prediction. The input could be w_{i-2}, w_{i-2}, w_{i+1}, w_{i+2}, the previous words and the following words of the current word w_i. With the skip-gram architecture, the model uses the current word to predict the context words. The input is w_i and the output would be w_{i-2}, w_{i-2}, w_{i+1}, w_{i+2}. Furthermore, the context words are not limited to the immediate context. Training instances could be created by skipping a constant number of words in w_i's context–for instance, w_{i-4}, w_{i-3}, w_{i+3}, w_{i+4}.

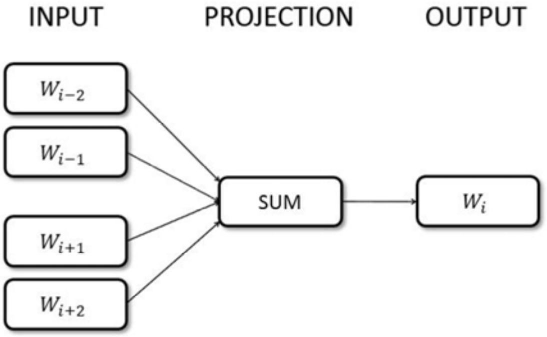

Figure 1. CBoW model & skip-gram model.

According to Mikolov *et al*. (Mikolov *et al*., 2013) and some previous studies (Bansal, Gimpel & Livescu, 2014), the CBoW model performs better in text classification, especially suitable for documents containing very few infrequent words. In addition, training the CBoW model is much faster than the skip-gram model. Based on our preliminary analysis, our ROG forum dataset is composed of 421 thousand posts, most of the posts contain very few infrequent words. We decide to adopt the CBoW model.

3.2 Ranking Function for Question Title and Description

Once the word embedding is learned, questions can be represented by word vectors. Semantic similarities between query questions and archived questions represented by CBoW are

[1] https://code.google.com/archive/p/word2vec/

believed to be more accurate. We calculate *q*'s vector as follows:

$$V_{sen}(q) = \frac{1}{L_{qv}} \sum_{w \in q} v(w) \tag{9}$$

$$v(w) = \begin{cases} V_{cbow}(w), & w \in |V_{cbow}| \\ \text{NULL}, & \text{otherwise} \end{cases} \tag{10}$$

$$L_{qv} = \sqrt{Len_v} \tag{11}$$

$$Len_v = \sum_{e \in V_{sen}(q)} e^2 \tag{12}$$

Where *w* is each word in question *q*, we retrieve the vector of *w* in the training vocabulary. *e* is the value of each dimension in the vector. After getting the V_{sen}, we can calculate the similarity score by this method:

$$S(q,Q) = V_{sen}(q) \cdot V_{sen}(Q) = \sum_{i=1}^{D} e(q_i) \cdot e(Q_i) \tag{13}$$

Here *D* is the dimension size of the sentence vector. $e(q_i)$ and $e(Q_i)$ are the values of each dimension in the query question *q* and archived question *Q*. In our study, we treat the title and description fields of a forum question as two different parts. Several previous approaches such as (Zhang, Wu, Wang, Zhou & Li, 2016; Zhou, He, Zhao & Hu, 2015) combine title and description into one. Ideally, users should describe their main question in the title field and write a more detailed situation in the description field. Often, people write something with no clear connection to their problem in the title field, such as "Need Help!", "Not Happy" or "Error Code". From these kinds of titles, it is hard to interpret what the user's true question is. Even if the problem is clearly depicted in the description yet the title is unclear, combining a meaningless title with a particular description might actually lower the ranking score. Based on the facts mentioned above, we propose a prototype ranking function to measure title and description scores separately as follows:

$$R(q,Q) = \alpha \times S_{title}(q,Q) + \beta \times S_{desc}(q,Q) \tag{14}$$

Where $S_{title}(q,Q)$ is the score of the title and $S_{desc}(q,Q)$ is the score of the description between input question *q* and archived question *Q*. α and β are both free tuning parameters for finding the balance between title and description. Here we let $\alpha + \beta = 1$.

3.3 Utilizing User Reputation in The Forum

User reputation in its simplest form is a ranking of how the community scores a user's contributions to the forum. A user's reputation is given by other forum participants who read the user's posts. Positive reputation should be given to people whose posts are meaningful, helpful and thoughtful. Negative reputation should be given to users posting something that detracts from the conversation. So we improve the ranking function with an extra element:

reputation of participants. The new measurement is described as follows:

$$R(q,Q) = \alpha \times S_{title}(q,Q) + \beta \times S_{desc}(q,Q) + \gamma \times RPU(Q) \tag{15}$$

$$RPU(Q) = \frac{1}{\#u} \sum_{u \in Q} RP(u) \tag{16}$$

In this formula, we extend the function from above and add the reputation point. γ is a tuning parameter for reputation. $RPU(Q)$ is the summation of the reputation points of the users participating in the discussion of Q. Any one of the participants may post several answers in the same thread. To avoid too many reputation points from the same forum user, we only add each participant's reputation point once. To ensure fairness for newer post, we average the reputation point by the number of participants. Here we let $\alpha + \beta + \gamma = 1$. The reputation system of the ASUS ROG forum offers all participants a fairer and equal platform. Each registered user can anonymously offer points to anyone who posts an appropriate and useful answer under a discussion thread. There is only one way to gain points, when someone approves of the post. This is much more objective so we think it is a suitable factor to evaluate candidate questions.

4. Experiment and Evaluation

In this chapter, we present experiments to evaluate the performance of the proposed approach for question retrieval.

4.1 Data Sets

We collect data sets from the official ASUS Republic of Gamers discussion forum[2]. Unlike the general questions on other community sites, people discuss PC-related technical topics on the ROG forum such as overclocking, tweaking and cooling. For our experimental dataset, we extracted 42,899 threads and 420,983 posts archived in the ROG forum. Each thread consists of a title, a description and the discussion of the participants. For question retrieval, we look at not only titles and descriptions fields but also the reputation of participants.

4.2 Validation Set and Test Set

We assume that the title and description of threads already provide enough information for users to understand. We created a test set from the ROG forum by using the Lucene search engine with the default and BM25 similarity scoring functions to index all data from the ROG forum. All questions are stemmed and lowercased. Stopwords, HTML and forum tags are also removed. We randomly choose questions from the database with title length greater than 25 characters so that the title would be more likely to be meaningful. Also, the same criterion is

[2] http://rog.asus.com/forum

applied to the query questions. Then we retrieved 10 candidate questions from the corresponding indexed data using default and BM25 similarity ranking algorithms in Lucene. After retrieval, we labeled the relevance for the candidate questions regarding to the input queries. If a candidate question is considered semantically similar to the query, it will be labeled as relevant; otherwise it will be labeled as irrelevant. We use the labeled dataset of default similarity as the validation set and the dataset of BM25 as the test set. The validation set is used for tuning parameters of different models, whereas the test set is used for evaluating how well the models rank relevant candidates and irrelevant candidates.

4.3 Word2vec Training

In our experiments, we trained word embedding with a whole discussion dataset from the ROG forum site. Before training word embedding, some pre-processing was executed. Each character was converted to lowercase. Forum tag language, redundant spaces and duplicate symbols were removed. Finally, every word was stemmed and stopwords were removed. Here, we trained the word embedding by using the CBoW method. The parameters we set for training are as follows: 200 dimensions for the size of word vectors, and a max skip length between words of 8.

4.4 Baselines

In this paper, we implement several methods to be the baseline for comparison.

4.4.1 Language Model for Information Retrieval (LMIR)

$$P(q|Q) = \prod_{w \in q} (1-\lambda) P_{ml}(w|Q) + \lambda P_{ml}(w|C) \tag{17}$$

LMIR (Ponte & Croft, 1998; Zhai & Lafferty, 2004) is based on the probability of each word in query question q that appears in candidate question Q and the large collection C.

4.4.2 Language Model with Category Smoothing (LMC)

$$P(q|Q) = \prod_{w \in q} (1-\lambda) P_{ml}(w|Q) + \lambda P_s(w|Q) \tag{18}$$

$$P_s(w|Q) = (1-\beta) P_{ml}(w|C) + \beta P_{ml}(w|Cat(q)) \tag{19}$$

LMC (Cao *et al.*, 2009) extends LMIR by introducing the probability of each word in q that appears in the category of candidate Q.

4.4.3 Distributed Representation Based Language Model (DRLM)

The last compared configuration employs DRLM (Zhang *et al.*, 2016), which considers creating a retrieval model with learned representations of words. It borrows the idea from

TBLMs and incorporates word-to-word similarity calculated with the learned vectors into LMIR. The model finds the top N similar words for each word with Cosine similarity and defines a word-to-word similarity function as:

$$P_{sim}(w_i|w_j) = \begin{cases} \frac{e^{v(w_i)\cdot v(w_j)}}{\sum_{w' \in Sim(w_j)} e^{v(w')\cdot v(w_j)}}, & \text{if } w_i \in Sim(w_j) \\ 0, & \text{otherwise} \end{cases} \tag{20}$$

Here, $Sim(w_j)$ represents the top N similar words of w_j, and $P_{sim}(w_i|w_j)$ is the translation probability from w_i to w_j. The idea is to replace the translation probability $P_{tp}(w|v)$ in TBLM with $P_{sim}(w_i|w_j)$. DRLM is also combined with a word-category similarity function, which is defined as:

$$Scat(w|c) = \frac{e^{v(w)\cdot v(c)}}{\sum_{w' \in V} e^{v(w')\cdot v(c)}} \tag{21}$$

Therefore, given a query question q and a candidate question Q, the DRLM retrieval model can be represented as follows:

$$P(q|Q) = \prod_{w \in q}(1-\lambda)P_{mx}(w|Q) + \lambda P_s(w|Q) \tag{22}$$

$$P_{mx}(w|Q) = (1-\alpha)P_{ml}(w|Q) + \alpha P_{sim}(w|Q) \tag{23}$$

$$P_s(w|Q) = (1-\beta)P_{ml}(w|C) + \beta Scat(w|Cat(q)) \tag{24}$$

$$P_{sim}(w|Q) = \sum_{v \in Q} P_{sim}(w|v)P_{ml}(v|Q) \tag{25}$$

4.5 Evaluation Metrics

In order to evaluate the performance of different models, we used mean average precision (MAP), and precision at K (P@3, P@1) as evaluation measures. These measures are widely used in the literature for question retrieval in community-based Q&A.

4.6 Main Results

In this section, we present the experimental results on our test sets of the ROG forum data. We compare Lucene, LMIR, LMC and DRLM_nocat against our approach. The number of dimensions of word2vec training is set to 200. We have implemented LMIR, LMC and DRLM models based on the original papers and set all the tuning parameters on our dataset. Table 2 shows the best tuning parameters of title, description and reputation for each approach.

Table 1 shows question retrieval performance in terms of different evaluation metrics. DRLM_nocat is better than Lucene except on P@1. By using only title similarity (Forum- T) or content similarity (Forum-C), our system obtains a comparative score to those of other state-of-the-art methods. After using both title and content scores (Forum-TC), our system performs better than DRLM_nocat. This indicates that considering titles and descriptions separately improves accuracy of similarity scores between questions. We also test our methods with Wiki trained data. In Wiki-T, we use only title score as in Forum-T. In Wiki-TC, we use both scores of title and description as in Forum-TC. Table 3 shows that Wiki performs the worst, indicating that in-domain training data is more effective than out-of-domain training data for word2vec training. Finally, we can see that Forum-TCR outperforms all other methods. It takes advantage of Forum-TC and participants' reputation.

4.7 Positive and Error Cases

One of the positive cases is the input query is "Fan Xpert 2 issues: access violation at address 0040b590...". After our ranking function, the 10th question: "Fan Xpert II Problem" is raised to be the first one. Because both their descriptions describe they can't start the application normally. The error case is like that the input is "Maximus V Extreme fans running after shutdown.", and the ranking dropped the first result: "PC doesn't power off when shutdown" to 9th. We found both questions said the fans are still spinning after PC shutting down. But the archived question does not mention this in its title. So our system gives high description score but low title score for the correct archived question.

Table 1. Performance of the state-of-the-art methods and our proposed methods.

	MAP	P@3	P@1
Lucene	0.423	0.272	0.408
LMIR	0.446	0.333	0.408
LMC	0.446	0.333	0.408
DRLM_nocat	0.468	0.340	0.398
LMIR TC	0.449	0.344	0.439
LMC TC	0.447	0.337	0.439
DRLM_nocat TC	0.456	0.34	0.459
Forum-T	0.441	0.323	0.418
Forum-C	0.473	0.354	0.408
Forum-TC	0.487	0.354	0.439
Forum-TCR	0.507	0.367	0.510

Table 2. Best parameters.

	Title	Description	Reputation
LMIR TC, LMC TC, DRLM_nocat TC	0.3	0.7	N/A
Forum-TC	0.2	0.8	N/A
Forum-TCR	0.4	0.5	0.1

Table 3. Comparison of using the in-domain word2vec and the out-domain word2vec.

	MAP	P@3	P@1
Lucene	0.423	0.272	0.408
Wiki-T	0.363	0.262	0.286
Wiki-TC	0.369	0.276	0.265

5. Conclusion

This paper proposes to learn vector representation for question retrieval in community forums and to exploit participant reputation to improve retrieval. We believe that title and description fields should be analyzed separately. Unlike baseline methods, our approach calculates the similarity between the query question and each archived question’s title as well as the similarity between the query question and each archived question’s description. As mentioned above, we create a retrieval model which combines user reputation and the learned word embedding representation of title and description. Evaluation results on our ROG forum dataset indicate that our proposed approach can dramatically enhance cQA question retrieval.

Acknowledgement

This research was supported in part by the Ministry of Science and Technology of Taiwan (MOST 106-2633-E-002-001), National Taiwan University (NTU-106R104045), Intel Corporation, and Delta Electronics, and Advantech.

Reference

Bansal, M., Gimpel, K. & Livescu, K. (2014). Tailoring Continuous Word Representations for Dependency Parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics(ACL)*, *2*, 809-815. doi: 10.3115/v1/P14-2131

Berger, A., Caruana, R., Cohn, D., Freitag, D. & Mittal, V. (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(SIGIR '00)*, 192-199. doi: 10.1145/345508.345576

Cao, X., Cong, G., Cui, B., Jensen, C. S. & Zhang, C. (2009). The use of categorization information in language models for question retrieval. In *Proceedings of the 18th ACM conference on Information and knowledge management(CIKM '09)*, 265-274. doi: 10.1145/1645953.1645989

Mikolov, T., Chen, K., Corrado, G. & Dean, J. (2013). Efficient estimation of word representations in vector space. Retrived from arXiv preprint arXiv:1301.3781

Ponte, J. M. & 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(SIGIR '98),* 275-281. doi: 10.1145/290941.291008

Song, F. & Croft, W. B. (1999). A general language model for information retrieval. In *Proceedings of the eighth international conference on Information and knowledge management(CIKM '99)*, 316-32. doi: 10.1145/319950.320022

Xue, X., Jeon, J. & Croft, W. B. (2008). Retrieval models for question and answer archives. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval(SIGIR '08)*, 475-482. doi: 10.1145/1390334.1390416

Zhai, C. & Lafferty, J. (2004). A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems (TOIS)*, *22*(2), 179-214. doi: 10.1145/984321.984322

Zhang, K., Wu, W., Wang, F., Zhou, M. & Li, Z. (2016). Learning Distributed Representations of Data in Community Question Answering for Question Retrieval. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining(WSDM '16)*, 533-542. doi: 10.1145/2835776.2835786

Zhou, G., He, T., Zhao, J. & Hu, P. (2015). Learning continuous word embedding with metadata for question retrieval in community question answering. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, 250-259.