VNU Journal of Science: Comp. Science & Com. Eng., Vol. 35, No. 2 (2019) 47–54

VNU Journal of Science:
Computer Science and Communication Engineering

Journal homepage: http://www.jcsce.vnu.edu.vn/index.php/jcsce

Original Article

A Deep Learning Model of Multiple Knowledge Sources Integration for Community Question Answering

Van-Tu Nguyen[1], Anh-Cuong Le[2,*]

[1]*VNU University of Engineering and Technology, 144 Xuan Thuy, Cau Giay, Hanoi, Vietnam*
[2]*Natural Language Processing and Knowledge Discovery Laboratory, Faculty of Information Technology, Ton Duc Thang University, Ho Chi Minh City, Vietnam*

Received 22 March 2019
Revised 19 September 2019, Accepted 30 September 2019

Abstract: The community Question Answering (cQA) problem requires the task that given a question it aims at selecting the most related question-answer tuples (a question and its answers) from the stored question-answer tuples data set. The core mission of this task is to measure the similarity (or relationship) between an input question and questions from the given question-answer data set. Under our observation, there are either various information sources as well as different measurement models which can provide complementary knowledge for indicating the relationship between questions and question-answer tuples. In this paper we address the problem of modeling and combining multiple knowledge sources for determining and ranking the most related question-answer tuples given an input question for cQA problem. Our proposed model will generate different features based on different representations of the data as well as on different methods and then integrate this information into the BERT model for similarity measurement in cQA problem. We evaluate our proposed model on the SemEval 2016 data set and achieve the state-of-the-art result.

Keywords: Community question answering, Multi knowledge sources, Deep learning, The BERT model.

1. Introduction

As you know there are now many cQA forums that have become very popular and an extremely valuable source of information for users, such as StackOverflow[1] and Quora[2]. In a cQA forum whenever a user submit a new question, the cQA must have a mechanism to return the most related questions in relation to the new question.

For measuring the similarity between a new question and the existed questions the most

* Corresponding author.
E-mail address: leanhcuong@tdtu.edu.vn

https://doi.org/10.25073/2588-1086/vnucsce.230

[1]https://stackoverflow.com/
[2]https://www.quora.com/

important task is how to represent questions. Early studies used simple information such as words and phrases to information requiring further analysis of the language such as syntax and semantics [1–3]. However, although using grammatical and semantic information can be effective, it takes a lot of computation time and requires a extension tools (syntactic parser, lexicons, knowledge bases, etc.) that are not always available, especially for new application domains or for new languages. Actually, adapting knowledge based systems to a new domain requires not only additional efforts to tune feature extraction pipelines but also adding new resources that may not even exist.

Natural language processing (NLP) studies often face the problem of lack of knowledge resources, specially for labeled data. To address this challenge NLP models have recently been improved by deep learning and representation learning models that takes advantage of using unlabeled data. These studies use pre-train data sets of word representation by vectors which are learnt from unlabeled texts by the models such as Word2vec, Glove, Fasttext. Some Question Answering studies have used these word representations in deep learning models such as in [4, 5]. More advanced models of language representation such as BERT [6] have also been used. BERT extends the capabilities of previous methods by creating contextual representations based on previous words and next words to lead a linguistic model with richer semantics.

Although BERT has achieved incredible results in many natural language understanding tasks, its potential has not yet been fully explored. There is very little research to use BERT to improve the performance of cQA systems. In this study of cQA problem, we focus on building a model using BERT as the core component. We will investigate the effectiveness of integrating BERT with other models such as the Convolutional Neural Network (CNN), especially using additional knowledge sources from other approaches. This work is motivated by the sense that deep learning models require a lot of training data while knowledge based approaches use pre-defined features so it avoids sparse data problem. Furthermore, combining data representations according to different methods will complement the mutual knowledge between these approaches.

In the rest of the paper, we will present related studies, present the proposed model and experimental on various configurations (with different methods and data representations) on the SemEval 2016 data set to find the best model.

2. Related work

Evaluating the similarity between questions is an issue that has been widely studied in the problem of community question answering. Previous approaches for this task built on neural network based models using pre-trained word embeddings such as word2vec, Glove or FastText. In [4], the authors propose a model of deep fusion LSTMs (DF-LSTMs) to model the strong interaction of text pairs in a recursive matching way. The DF-LSTMs model consists of two LSTMs that are interdependent to model two chains of mutual influence. Another study presented in [5] proposed an architecture that uses one LSTM network to measure the semantic similarity between a pairs of sentences.

Recently, a great progress in language modeling has been achieved. Represent the two-dimensional encoder from the BERT model [6] using the mask language model. Language modeling is often refined in tasks such as text classification or question answering tasks. In [7], the authors have applied BERT to the Arabic language to handle some NLP tasks such as Sentiment Analysis (SA), Named Entity

Recognition (NER), and Question Answering (QA). Recent studies [8, 9] have shown that fine-tuning pre-trained transformer networks may outperform previous approaches to a variety of natural language processing tasks, including question answering. In [10], the authors investigated the use of the pre-trained BERT language model to solve Question Generation tasks from answers and context. They introduced three neural architectures built on BERT for Question Generation tasks. The first is to use a simple BERT model, which shows the shortcomings of using BERT directly for document generation. They then proposed two other models by restructuring BERT into a sequential way to get information from previously decoded results. These models were evaluated on the recent SQuAD QA dataset. Test results show that their best model significantly improves compared to previous models on the same dataset. In [11], the authors focus on improving the BERT model, reducing the number of model parameters to reduce memory consumption and increase the training speed of BERT. In [12], the authors have proposed three transformer-based question answering systems using the BERT, ALBERT and T5 models. Experimental study on the the Kaggle COVID-19 Open Research Dataset and CORD-19 dataset 2. The BERT-based QA system achieved the highest F1 score (26.32%), while the ALBERT-based QA system achieved the highest Exact Match (13.04%). In [13], the authors develop TransTQA, a community question answering system that provides automatic responses by retrieving appropriate responses based on the same questions answered true in the past. TransTQA is built on top of the ALBERT network, allowing it to respond quickly and accurately. Study [14] improves the performance of question answering systems based on BERT and RoBERTa by pruning the structure of parameters from the transformer model. Specifically, (1) they investigated structured pruning to reduce the number of parameters in each transformer class, (2) applicability to both BERT and RoBERTa-based models, (3) applicability for both SQuAD 2.0 and natural question, and (4) combine structured pruning with distillation. In [15], the authors proposed to improve the performance of Arabic community question answering system. They have integrated different types of similar features, in addition to exploring the effects of using preprocesscing. Furthermore, they developed a new deep neural network integration model that had better performance than before. This synthetic model benefits from semantic and lexical similarities. In addition, synthetic modeling has made use of recent advances in linguistic models using the BERT model.

3. The proposed model

3.1. BERT model

BERT is a transformer-based model that allows the contextual representation of a word based on its relation to its surrounding words [6]. BERT implements the self-attention mechanism in encoding words with the context of words, using multi-layered architecture with modeling the problem for word prediction using bi-dimensional contextual information. Many studies such as in [6, 16] have shown that NLP problems using BERT give much better results than other approaches.

BERT is also used effectively in classification problems. The common model is that it uses BERT for representing input text, and then connecting the encoding vector (obtained by BERT) with some fully connected neural layers for the task of classification. This work is called the BERT tuning task.

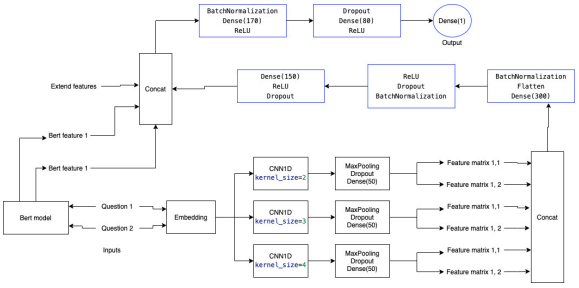

Fig. 1. The proposed model

3.2. BERT integrated with additional features

In this paper we will model the similarity measurement between the input question and an existed question in the databases as a two-class classification problem, determine whether two questions are similar or not, and use the output's probability to measure the similarity degree between the two questions.

The proposed model is shown in Figure 1. The goal of this model is to integrate features from other approaches into the BERT-based classification model in order to enrich the BERT based representation of the input, helping to overcome the problem of sparse data of all deep learning models.

We divide the additional features into two kinds, the first one is word embeddings from one other word2vec model (not BERT), and the second one is from other methods such as using question classification, question categories, and word phrases (n-grams). Figure 1 shows two additional feature kinds which are integrated with the vector encoded from BERT through the module 'Concat'. Moreover for embeddings feature, we have used CNN layers (Convolutional Neural Networks) to generate synthesized features before concatenation. Note that Figure 1 presents a full model, but in the experiments we perform many different models, with or without using the additional features, with or without using the CNN layers.

The next section will describe in detail how these features are generated.

4. Feature Determination

We inherit the task of feature determination from our previous study [17]. We divided extracted features into the two kinds: the first one belongs to the Word2Vec form (or word embeddings) using a word representation learning method, the second one contains features used in conventional machine learning approaches.

4.1. Word Embedding

A word embedding is essentially a vector of real numbers representing a word. Word embeddings are learned from unlabeled data (a set of natural language sentences) and express the semantic relationship between words.

For this kind of additional features, we use the continuous Skip-gram method to generate vector representations of the words in question. This representation of words is different from word representation of the BERT in the aspect that BERT encodes each word by different vector depending on its different context, while the Skip-gram method generates the same vector for one word without consider the difference of its contexts. Word embeddings are created using a different method so that will add more information to our BERT based model as show in Fig. 1 .

4.2. Extend Features

The second type of features we will use in our model extracted from conventional methods and called by the general name 'Extend Features'. They include n-grams, question types, and question categories which are determined as follows.

Conventional Features

We use some common features extracted from the surface forms of sentences, calculated based on the overlap between the input question and the related questions and answers. We use the ratio of the number of words and sentences between them. Beside that we also use features that are bag of overlap words, overlap nouns and overlap name entities. These features form a vector of features that we call F_1.

Question Type

From our observations, each question in the database of any cQA system usually uses question words to determine to determine different asking type. It is easy to understand that question type provides useful information to identify similarity between questions.

To identify question type features for each question, we will the set of question words which includes *"who", "when", "how", "why", "which", "where"*, and *"what"*. We then represent each question type of a question as an one-hot vector of that question words. For example, the question with question word "who" will be represented as the one-hot vector: [0, 1, 0, 0, 0, 0, 0]. Note that the vocabulary for question types is V = {*"what", "who", "when", "why", "where", "which", "how"*}. This form a feature vector that we call F_2

Question Category

We use the term 'question category' to represent the set of questions which belong to the same category which represents for question topic. Note that we are given the dataset Q includes question - answer pairs extracted from cQA sites, in which each question is assigned to a question category label. In order to use this information to measure the similarity between the input question and a related question in the database we implement the following steps:

Step 1: we will build a question classifier for determining the question category/label for each input question. This classifier is built using SVM classification. The training data for this classifier get from the Q dataset, each question in Q has been assigned a category label (question category).

Step 2: for each input question we firstly obtain its question category (we call this the category A) from the first step. We then get the representation of the set A by averaging all vectors (using the word embeddings module) of all words in all questions labeled A in the Q dataset. In the same way we also compute the representation of the set B of the related question.

Step 3: finally, we calculate the similarity between the input question and the related question by the Eq. 1. This forms the feature F_3

$$cosin\_sim(u, v) = \frac{\sum_{i=1}^{m} u_i * v_i}{\sqrt{\sum_{i=1}^{m} (u_i)^2} * \sqrt{\sum_{i=1}^{m} (v_i)^2}} \quad (1)$$

where u and v are two m-dimensional vectors represent the related question category and the input question category, respectively. u_i is the i^{th} element of u vector.

Finally, concatenate the three feature vectors extracted as above, we obtained the new feature vector $r = \{F_1, F_2, F_3\}$ which is denoted by 'Extend Feature' in the Fig. 1 which is added to our BERT based model.

5. Experiments

5.1. Dataset and Evaluation Metrics

In this section we build experiments on datasets. The first dataset is SemEval-2016 task 3, subtask B[3]. This is a dataset of questions and answers extracted from cQA Qatar Living (http://www.qatarliving.com/forum). The dataset includes 337 input questions and 3369 related questions. The dataset is pre-split into 267 input

[3]http://alt.qcri.org/semeval2016/task3/index.php?id=data-and-tools

Table 1. The statistics of SemEval 2016 dataset

Data set	Train data	Test data	Total
Input questions	267	70	337
Related question-answer pairs	2669 - 26690	700 - 7000	3369 - 33690

Table 2. The statistics of Quora dataset

Data set	Question pairs	Average of words	Average of characters
Train data	363665	11.17	60.11
Test data	40417	11.03	60.05

questions and 2669 related questions for training, as well as 70 input questions and 700 related questions for the test. Each data point is a pair of questions (input question and related question) and a similarity label, which is either 'Relevant' or 'Irrelevant'. We need to predict a binary label were 1 covers 'Relevant', and 0 covers 'Irrelevant', then ranking the related questions according to their similarity with input question. Some statistics of this dataset are shown in Table 1.

The second dataset we use to evaluate the proposed approach is the Quora[4] dataset. This dataset includes questions, answers extracted from cQA https://www.quora.com/. The dataset includes 404082 question pairs (input questions and related questions). The dataset is pre-split into 363665 question pairs for training, as well as 40417 question pairs for the test. Each data point is a pair of questions (new question and related question) and a similarity label, which is either 1 or 0. Table 2 provides statistics for Quora dataset.

5.2. Experiments and Results

We design various experimental models including BERT model without using additional attributes, BERT model using extra attributes (not including word2vec) which we call 'feature', and model BERT using add word2vec (combine the CNN tier) along with the 'feature'. We also compared with related models in our previous study [17] without using BERT.

Table 3. Our experiment results using SemEval 2016 dataset

Models	Acc	P	R	F1
word2vec + CNN [17]	73.71	53.65	62.19	57.60
word2vec + CNN + feature [17]	82.57	71.24	75.11	73.13
BERT	82.14	79.35	62.66	70.02
BERT + feature	81.14	67.35	84.12	74.81
BERT + word2vec+CNN	65.71	46.53	20.17	28.14
BERT + word2vec+CNN + feature	79.57	66.54	77.68	71.68

Table 4. Our experiment results using Quora dataset

Models	Acc	P	R	F1
word2vec+CNN [17]	77.85	60.54	71.54	65.58
word2vec+CNN + feature [17]	86.81	70.89	89.05	78.91
BERT	89.06	84.83	83.55	84.19
BERT + feature	89.00	81.75	88.11	84.81
BERT + word2vec+CNN	82.98	73.32	80.41	76.70
BERT + word2vec+CNN + feature	88.80	87.27	79.46	83.18

The results from Table 3 and Table 4 show that using more features and word2vec have increased the F1-scores and the accuracy. In which the best model is to use BERT with the 'feature'. When comparing with the previous study [17] that did not use BERT, the performance increased, especially on the Quorra dataset.

Note that we test on two datasets, Table 3 shows the results of the small dataset and Table 4 shows the results of the large dataset. The experimental results show that using extended features has significantly increased the models' accuracy on the small dataset while only slightly improving the accuracy on the large dataset. This confirms the claim that when the training data is insufficient then the additional knowledge is really important for the improvement of the model.

It is also interesting that adding CNN module to these models not only increases but also

[4]https://data.quora.com/First-Quora-Dataset-Release-Question-Pairs

reduces the accuracy of these models. In particular, adding CNN module gives very low results (F1 is 28.14%) in the results in Table 3. It is because the training data in Table 3 is so small that the more complex the model, the more it will cause overfitting. In Table 4 we can see the CNN module doesn't reduce much the quality of the model as in Table 3 , this is because the data in Table 4 has grown.

From these experiments we can draw the conclusion that the additional knowledge from other approaches will contribute to the increase of information for deep learning models, especially in the case of insufficient data. Another result is that too complex models will cause overfitting, especially when the training data is too small.

6. Conclusion

In this paper we have presented a new proposed model for the CQA problem. Our model is based on the BERT model and integrates a variety of features from other approaches. In our experiment, we performed many different configurations to compare and the obtained results showed that the integrated BERT model with a diverse set of attributes gave the best results. This result also demonstrates that the deep learning model will be improved (solving the sparse data problem) if more information-rich attributes from other approaches are integrated.

Acknowledgement

This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.05-2020.26

References

[1] Alberto, B.C., Bonadiman, D., Martino, G.D.S, Answer and Question Selection for Question Answering on Arabic and English Fora, in Proceedings of SemEval-2016 (2016) 896–903.

[2] Filice, S., Croce, D., Moschitti, A., Basili, R, Learning Semantic Relations between Questions and Answers, in Proceedings of SemEval-2016 (2016) 1116–1123.

[3] Wang, K., Ming, Z., Chua, T.S, A syntactic tree matching approach to finding similar questions in community-based qa services, in SIGIR, (2009) 187–194.

[4] Pengfei, L., Xipeng, Q., Jifan, C., Xuanjing, H, Deep fusion lstms for text semantic matching, in Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) DOI: 10.18653/v1/P16-1098 (2016) 1034–1043.

[5] Jonas, M., Aditya, T, Siamese recurrent architectures for learning sentence similarity, in Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16) (2016) 2786–2792.

[6] Jacob, D., Ming-Wei, C., Kenton, L., Kristina, T, Bert: Pre-training of deep bidirectional transformers for language understanding, in Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (2019) 4171–4186

[7] Wissam, A., Fady, B., Hazem, H, Arabert: Transformer-based model for arabic language understanding, in Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection (2020) 9–15.

[8] Lukovnikov, D., Fischer, A., Lehmann, J, Pretrained Transformers for Simple Question Answering over Knowledge Graphs, ArXiv, abs/2001.11985 (2019).

[9] Aken, B.V., Winter, B., Löser, A., Gers, F, How Does BERT Answer Questions?: A Layer-Wise Analysis of Transformer Representations, in Proceedings of the 28th ACM International Conference on Information and Knowledge Management (2019).

[10] Chan, Y., Fan, Y, A Recurrent BERT-based Model for Question Generation, in Proceedings of the Second Workshop on Machine Reading for Question Answering (2019) 154–162.

[11] Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R, ALBERT: A Lite BERT for Self-supervised Learning of Language Representations, ArXiv, abs/1909.11942 (2020).

[12] Ngai, H., Park, Y., Chen, J., Parsapoor, M, Transformer-Based Models for Question Answering on COVID19, ArXiv, abs/2101.11432 (2021).

[13] Yu, W., Wu, L., Deng, Y., Mahindru, R., Zeng, Q., Guven, S., Jiang, M, A Technical Question Answering System with Transfer Learning, in Proceedings of the 2020 EMNLP (Systems Demonstrations) (2020) 92–99.

[14] McCarley, J.S., Chakravarti, R., Sil, A, Structured Pruning of a BERT-based Question Answering Model, arXiv: Computation and Language (2019).

[15] Almiman, A., Osman, N., Torki, M, Deep neural network approach for arabic community question answering, Alexandria Engineering Journal 59 (2020) 4427–4434.

[16] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Polosukhin, I, Attention is all you need, in Advances in Neural Information Processing Systems (2017) 5998–6008.

[17] Nguyen, V.T., Le, A.C, Nguyen, H.N, A Model of Convolutional Neural Network Combined with External Knowledge to Measure the Question Similarity for Community Question Answering Systems, International Journal of Machine Learning and Computing 11 (3) (2021) 194–201 DOI: 10.18178/ijmlc.2021.11.3.1035.