Utilizing SBERT For Finding Similar Questions in Community Question Answering

Thi-Thanh Ha ∗†, Van-Nha Nguyen *, Kiem-Hieu Nguyen *, Kim-Anh Nguyen *, Quang-Khoat Than *

* Hanoi University of Science and Technology, Hanoi, Vietnam
† Thai Nguyen University of Information and Communication Technology, Vietnam

htthanh@ictu.edu.vn, nha282@gmail.com, {hieunk,anhnk,khoattq}@soict.hust.edu.vn

Abstract—The BERT model was fine-tuned to give state-of-the-art results in sentence-pair regressions. However, this model requires that both questions are fed into the network, which leads to a massive computational overhead. Instead of computing on n pairs of sentences, SBERT was proposed to learn sentence representation by computing on only one query question. This model was proven to work effectively on semantic textual similarity (STS). In this paper, we explore SBERT model for question retrieval in Community Question Answering. Results show that SBERT decreases slightly in performance compared to BERT4ECOMMERCE. However, This model reduces the effort for finding the most similar question from 795 seconds with BERT to about 0.828 seconds with SBERT, while maintaining the accuracy from BERT.

Index Terms—SBERT, question retrieval, community question answering

I. INTRODUCTION

Question retrieval is an important task in Community question answering system⁠1 with the purpose of answering new questions using previous answers in databases [1], [2]. It means that this is an intermediate task that supports problem of finding answer to a new question. We hoped that the answers of these related questions could be useful for the new query.

The problem of finding similar questions is defined as follows: Given a query question and a set of given questions in archive of questions, the goal is to rank these questions according to their similarity with respect to the query question.

Previous research addressed the lexical gap challenge between two questions. Lexical gap is a linguistic phenomenon where two questions have the same meaning but contain different words and phrases. Figure 1 is a typical pair of similar questions in our Vietnamese dataset.

Soft alignment technique originated from machine translation [3], implicitly disambiguated word meaning using topic models [4] and deep neural networks without depending on feature engineering or external knowledge bases [5], [6] were applied to deal with lexical gap challenge.

Recently, BERT [7] establishes state-of-the-art performance on various sentence classification and sentence-pair regression tasks including question retrieval task [8]. However, This model requires that both questions are fed into the network, which causes a massive computation. Furthermore, previous studies on finding similar questions only focused on improving accuracy performance but not pay attention to search time.

SBERT [9] was proposed and proven to be efficient in terms of search time while maintaining the accuracy from BERT on semantic textual similarity (STS) tasks with labeled sentence pairs. Prior researches were generally carried out on English datasets.

In the scope of this paper, we proposed to apply SBERT for question retrieval in both English and Vietnamese datasets. In recent research in [8] on Vietnamese, to find similar questions of n queries using BERT4ECOMMERCE in a collection of n questions, we need $O(m.n)$ inference computations on pair questions while we only need $O(m)$ sentence embeddings computation using SBERT.

We evaluated thorough different pooling strategies (MEAN, MAX, MLP and CLS) with using different ranking loss functions. We find that, first, using the representation of the [CLS] token of SBERT is better sentence representation than using MEAN, MAX and MLP layer. Second, training SBERT with triplet loss for triplet embedding brings improvements on both datasets.

II. RELATED WORK

Over the recent years, there are numerous methods proposed to deal with community question answering tasks [10], [11] such as answer selection [3], [12]–[15], question retrieval [13], [16]–[19] and achieved state-of-the-art results.

Prior approaches for sentence embedding consist of two directions: (1) unsupervised sentence embedding using unlabeled sentences, and (2) supervised learning with labeled sentences. With unsupervised sentence embedding, the input is usually a single unordered sentence such as recursive auto-encoders [20], denoising auto-encoders [21], the paragraph vector model [22] and ordered sentences utilizing the distributional hypothesis namely skip-thought [23] and FastSent [21]. In order to utilize labeled data for sentence embedding, InferSent model [24] was proposed using the Stanford Natural Language Inference dataset (SNLI) and SBERT [9] also applied a siamese network on NLI to encode sentences.

BERT (Bidirectional Encoder Representations from Transformers) was proposed in [7] as a kind of language model, which was performed to various NLP tasks with state-of-the-art performance, including sentence classification, question answering, and sentence pair regression. The limitation of BERT network structure is that independent sentence embeddings cannot be computed directly, which is difficult to derive

⁠¹https://stackoverflow.com/, https://www.qatarliving.com/
sentence embeddings from BERT. To pass this drawback, a common solution is that single sentences were pushed BERT and then a fixed-sized vector was derived by averaging the outputs (similar with average word embeddings) or by using the output of the first token (the CLS (classification) token) [25]. However, it produces quite bad sentence embeddings.

Sentence-BERT (SBERT) [9] used a Siamese network structure for semantic textual similarity (STS) problem to derived semantically meaningful sentence embeddings. Then, using a similarity measure like cosine-similarity or Manhattan/Euclidean distance, semantic textual similarity between two sentence embeddings is calculated. The CF-BERT [26] was proposed to push sentences with different lengths to generate fixed-length representations. This model focused on the important components of a sentence with their syntactic relations. These researches were generally conducted on Semantic Textual Similarity (STS) and Natural Language Inference (NLI) benchmarks.

Moreover, for finding similar questions task, the studies aimed to improve the accuracy performance but not attend to improve the time of searching similar questions. In this paper, we explore SBERT for question retrieval task in community question answering.

### III. SBERT for Question Retrieval in Vietnamese E-commerce Dataset

![Figure 1. An examples of similar questions](image)

![Figure 2. SBERT with cross entropy loss function](image)

![Figure 3. SBERT with contrastive loss function](image)

![Figure 4. SBERT with triple loss function](image)

![Figure 5. SBERT with multiple-negatives loss function](image)

**Question 1**: Tôi muốn hỏi là khi làm hồ sơ trả góp online khi tôi đến cửa hàng thì làm những gì
(I want to ask, what should I when I make an online installment document when I go to the store)

**Question 2**: Cho em hỏi nếu mua góp điện thoại Realme không trả trước cần hồ sơ những gì a
( Please tell me what documents do I need if I buy a Realme phone without prepayment?)
SBERT was proposed by Reimers [9]. It is BERT model added a pooling layer to the output to get a fixed-sized sentence embedding. We implemented with four pooling strategies: Using the output of the CLS-token, computing the mean of all output vectors (MEAN-strategy), computing a max of the output vectors (MAX-strategy) and MLP layer.

We fine-tuned SBERT to update the weights and produce semantically meaningful sentence embeddings of given questions in archive of questions. When a new question is posted on CQA system, this question is passed in to SBERT to compute sentence representation. Cosine-similarity is chosen to find a similar question with a query question.

SBERT based on siameses neural networks [27] proved very effective for feature extraction, metric learning. SBERT includes two or three backbones of BERT which share weights. Different loss functions are proposed for training SBERT. Four commonly used functions are cross-entropy loss, triplet loss, contrastive loss and multiple-negatives ranking loss.

**SBERT with cross entropy loss function**: Softmax layer is added in this model and cross entropy loss is optimized to learn weights. This structure is displayed in figure 2 [9]. This function is followed:

$$L = -\frac{1}{N} \sum_{i=1}^{N} (y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i)), \quad (1)$$

where $N$ is number of pair questions, $y_i$ and $\hat{y}_i$ are true and predicted value.

**SBERT with contrastive loss function**: Unlike Cross-Entropy Loss, whose objective is to learn to predict directly a label, the objective of ranking loss is to predict relative distances between questions. Commonly used one is contrastive loss which is shown in Figure 3.

The contrastive loss uses pairs of questions which can be an original question (anchor) and positive questions or original question and negative question. This function means that if the questions are original question and positive question, they are pulled together so that they are close in vector space; otherwise, their distance is increased. Loss function [28] is:

$$L = \frac{1}{N} \sum_{i=1}^{N} \left[ (1 - y_i) \cdot \frac{1}{2} (D_W)^2 + y_i \cdot \frac{1}{2} \cdot \text{max}(0, m - D_W)^2 \right], \quad (2)$$

where $y$ is of zero and one when the pair questions is original-positive question and original-negative question, respectively. $D_W$ is the euclidean distance between two questions, $m$ is a margin.

**SBERT with triplet loss function**: The triplet loss uses original question (anchor), positive question , and negative question. The triplet loss tries to reduce the distance of original question and positive embeddings and desires to increase the distance of anchor question and negative embeddings. As long as the distances of original - negative question pairs get longer than the distances of original-positive question pairs by a margin $m$, the desired embedding is obtained. While the contrastive loss performs one by one, triplet loss operates simultaneously (shown in Figure 4). We optimize the following triplet loss function [28]:

$$L = \frac{1}{N} \sum_{i=1}^{N} \text{max}(\|s_a - s_p\| - \|s_a - s_n\| + m, 0), \quad (3)$$

$\|\cdot\|$ is a distance metric. $s_a, s_p, s_n$ are sentence embedding of original, positive and negative question.

**SBERT with multiple-negatives loss function**: A triplet loss aims to shorten the embedded distance between the query and positive questions while enlarging the distance between the query and negative questions. In Vietnamese E-commerce dataset, the number of positive questions are smaller than the number of negative questions. There are two ways to handle these negative questions: one is to treat them independently (using triplet loss) and the other is to jointly handle (using multiple-negative loss). Multiple-negatives focus to balance the distance of positive questions over multiple-negative questions. This loss function over multiple-negatives [29] is given by:

$$L = -\frac{1}{N} \frac{1}{K} \sum_{i=1}^{K} [S(x_i, y_i) - \log \sum_{j=1}^{K} e^{S(x_i, y_j)}], \quad (4)$$

where $(y_i, x_i)$ are query question and positive question. $(y_j, x_j), j \neq i$, are query and negative question. $S(x_i, y_i)$ is a distance of two questions. In this formular (4), $K$-1 negative questions are pushed simultaneously (see Figure 5).

**IV. Dataset**

For the experiments, we used Vietnamese E-commerce dataset and SemEval 2017 task 3 [10] to evaluated our method. Vietnamese dataset [8] was collected questions from users in QA section of The gioi Di dong - an e-commerce website on mobiles, laptops and other electronic devices. The data is labeled with the relevance of each related question with respect to the original question. SemEval dataset based on data from Qatar Living forum [10]. They consist of three separated sets: training, development, and test (see table I). In this dataset, each original has 10 related questions. The table I lists the number pairs of questions in which 30% of pairs of questions were annotated as relevant to the original question. We also exploited the unlabeled corpus for pre-trained embeddings (see table II).
Table II
STATISTICS OF UNLABELED CORPUS CRAWLED FROM THE GIOI DI DONG.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Corpus size</td>
<td>1.1M</td>
</tr>
<tr>
<td>Vocabulary size (syllable)</td>
<td>151,735</td>
</tr>
<tr>
<td>Average length (syllable)</td>
<td>31</td>
</tr>
</tbody>
</table>

Table III
MAP SCORE OF SBERT WITH DIFFERENT LOSS FUNCTION

<table>
<thead>
<tr>
<th>Loss function</th>
<th>SemEval 2017</th>
<th>Vietnamese dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cross-entropy</td>
<td>42.81</td>
<td>45.00</td>
</tr>
<tr>
<td>Contrastive</td>
<td>44.07</td>
<td>55.01</td>
</tr>
<tr>
<td>Triplet</td>
<td>46.50</td>
<td>64.70</td>
</tr>
<tr>
<td>Multiple-negative</td>
<td>43.54</td>
<td>47.15</td>
</tr>
</tbody>
</table>

V. EXPERIMENTS AND DISCUSSIONS

Mean Average Precision (MAP) were used as evaluation metrics with evaluation scripts provided by SemEval organizers. MAP examines the ranks of all the correct answers and it is calculated as follows:

\[
MAP = \frac{1}{|N|} \sum_{j=1}^{N} \frac{1}{m_j} \sum_{k=1}^{m_j} \text{Precision}(R_{jk}),
\]

where \(R_{jk}\) is the set of ranked retrieval results from the top result until you get to the question \(q_k\), \(m_j\) is the number of correct similarity questions for a query question \(q_j\) in \(N\).

Our models were implemented using Tensorflow and all experiments were conducted on GPU Nvidia Tesla p100 16Gb. Hyper-parameters were tuned on the development set.

For SemEval 2017: we fine-tuning BERT and SBERT on finding similar questions with maximum length of 128, batch size of 16, learning rate of \(2 \times 10^{-5}\) with 20000 steps and margin \(m = 0.5\).

For Vietnamese dataset: we pre-trained BERT on unlabeled Vietnamese with maximum length of 200, batch size of 32, and learning rate of \(2 \times 10^{-5}\) with 20000 steps. BERT experiments are performed using Multilingual BERT-BASE model\(^3\). We trained SBERT with 10 epochs, batch size of 64, Adam optimizer with learning rate \(2 \times 10^{-5}\) and we set margin \(m = 1\) on labeled Vietnamese dataset.

Firstly, we fine-tuned SBERT for question retrieval on both labeled datasets with different loss functions detailed in section III (see table III). Secondly, SBERT was compared with other models with different pooling strategies (see table IV on SemEval 2017 dataset, table V on Vietnamese dataset). Lastly, computational efficiency of SBERT is compared to the result of BERT4ECOMMERCE [8] (list in table V).

SBERT with triplet architecture is compared to other models. In order to train SBERT with triplet network, we created 72,711 pairs (including original, positive and negative question) on training set. Each original question has 10 related questions, we added 30 negative questions in other original questions.

Table III shows that MAP score of SBERT with triplet loss function is the highest on both datasets. So we utilized this architecture to compare with other models.

In table V, we compared SBERT with other baseline models: SVM, LSTM, CNN, ABCNN, LSTM attention. The results of these model were reported in [8]. Previous studies attempted to derive sentence embedding with many strategies. In this paper, we also evaluated different pooling strategies as MEAN, MAX, CLS-token, MLP. The results illustrate that the pooling strategy has a large impact. CLS-token strategy performs significantly better than other strategies. This is consistent with SemEval 2017 dataset in table IV. MLP layer is added to reduce dimension from 768 to 256 dimensions and MAP score is the lowest at 52%. It is evident that CLS token can represent usefully the meaning of the entire sentence [7]. Moreover, the performance of SBERT rises significantly by 7.9% comparing to BERT with CLS-token in pre-trained BERT on Vietnamese E-commerce dataset. Reimers [9] and Yan [30] also shown that the approach using CLS token of pre-trained BERT yield unsatisfactory sentence embeddings. However, MAP score of SBERT decrease by 5.8% comparing with BERT4ECOMMERCE in [8]. It is easy to understand because SBERT is less interaction of question pairs [9], [26]. Although of decreasing of prediction performance, the time of finding similar questions is faster many times compare to BERT4ECOMMERCE. The time searching per each query is 0.153 seconds on GPU with GTX 1070 Ti 8GB and 0.828 seconds on CPU with Xeon(R) CPU X5647 @ 2.93GHz 16, faster than BERT4ECOMMERCE by 301 times on GPU and 960 times on CPU. The reason for computation efficiency of SBERT model is as follows: Assume the number of operations of the BERT model is denoted by \(a\); \(n\) is the number of given questions in archive. When a new query question is posted, there are \(n.a\) operations to derive representation pairs of questions in inference process of BERT4ECOMMERCE. For SBERT, because triplet architecture is shared weight, for each query question, we need only \(a\) operations to generate sentence embedding. It means that when there are \(m\) queries, \(m.n.a\), \(m.a\) are the number of operations on BERT4ECOMMERCE and SBERT respectively.

Table IV demonstrates that MAP scores in SemEval set are lower than that scores in Vietnamese dataset for several reasons. Firstly, the size of SemEval is less than that of Vietnamese. Secondly, BERT which using in Semeval was not pre-trained on dataset with same domain with SemEval dataset. Research in [8] proved that when a BERT model pre-trained and fine-tuned on same texts yields significant improvement on question retrieval over BERT trained on general-domain texts.

VI. CONCLUSION

We carried out a range of experiments with SBERT for question retrieval in CQA. In particular, SBERT model could be useful for semantic similarity comparison, clustering, and information retrieval via semantic search by deriving semantically meaningful sentence embeddings.

\(^3\)https://github.com/google-research/bert
Table IV
MAP score of SBERT with triplet architecture on SemEval 2017

<table>
<thead>
<tr>
<th>Models</th>
<th>MAP</th>
<th>Time (seconds per each query)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT</td>
<td>56.03</td>
<td>–</td>
</tr>
<tr>
<td>SBERT (CLS)</td>
<td>46.50</td>
<td>–</td>
</tr>
<tr>
<td>SBERT(MEAN pooling)</td>
<td>44.50</td>
<td>–</td>
</tr>
<tr>
<td>SBERT(MAX pooling)</td>
<td>45.00</td>
<td>–</td>
</tr>
<tr>
<td>SBERT(MLP 256 dimensions)</td>
<td>33.00</td>
<td>–</td>
</tr>
<tr>
<td>KELP [10]</td>
<td>49.00</td>
<td>–</td>
</tr>
<tr>
<td>Simbow [10]</td>
<td>47.22</td>
<td>–</td>
</tr>
</tbody>
</table>

Table V
MAP score of SBERT with triplet architecture on Thegiodiodong dataset.

<table>
<thead>
<tr>
<th>Models</th>
<th>MAP</th>
<th>Time (seconds per each query)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT(ecommerce)</td>
<td>70.50</td>
<td>–</td>
</tr>
<tr>
<td>BERT (CLS)</td>
<td>56.60</td>
<td>–</td>
</tr>
<tr>
<td>SBERT (CLS)</td>
<td>64.70</td>
<td>–</td>
</tr>
<tr>
<td>SBERT(MEAN pooling)</td>
<td>60.83</td>
<td>–</td>
</tr>
<tr>
<td>SBERT(MAX pooling)</td>
<td>60.16</td>
<td>–</td>
</tr>
<tr>
<td>SBERT(MLP 256 dimensions)</td>
<td>52.00</td>
<td>–</td>
</tr>
<tr>
<td>LSTM</td>
<td>52.60</td>
<td>–</td>
</tr>
<tr>
<td>CNN</td>
<td>53.10</td>
<td>–</td>
</tr>
<tr>
<td>ABCNN</td>
<td>51.52</td>
<td>–</td>
</tr>
<tr>
<td>LSTM attention</td>
<td>55.50</td>
<td>–</td>
</tr>
<tr>
<td>ElasticSearch</td>
<td>52.00</td>
<td>–</td>
</tr>
<tr>
<td>SVM</td>
<td>49.75</td>
<td>–</td>
</tr>
</tbody>
</table>

We hope our work can give a boost to applications related to CQA on Vietnamese E-commerce data. In the future, we are going to pre-train BERT on unlabeled SemEval dataset. Moreover, we are going to design a Vietnamese CQA on a specific e-commerce domain and apply experiments in this system.

ACKNOWLEDGEMENT

This work was funded by Gia Lam Urban Development and Investment Company Limited, VinGroup and supported by VinGroup Innovation Foundation (VINIF) under project code VINIF.2019.DA18

REFERENCES


