Text Mining and Search

Optimizing Stack Overflow Question Retrieval using BERT-based Re-ranking

Kevin Capano

Abstract

This work presents a novel approach for an information retrieval system in the domain of Stack Overflow question-answers.

The proposed system consists of two main parts: a first answer retrieval phase and a re-ranking phase. The base information retrieval system is designed to take a Stack Overflow question as input and return the most relevant answers for that question, using Word2Vec centroids.

To improve the performance of the IR system, a re-ranking phase is performed using a fine-tuned BERT model, trained with negative sampling on a relevant-non relevant binary task. The re-ranking is used to boost the performance of the IR system, which is evaluated using the Recall@5, 10, and 20.

The proposed system achieved promising results, demonstrating the effectiveness of this approach.

1. Introduction

The need for efficient and accurate question answering systems has been growing as the amount of available information on the internet increases. Especially in the field of computer science, web question-answer pairs are often very similar in terms of context and subjects, with only a few parts changing from one question to another on the same subject. In this scenario, capturing the semantics of the sentences inside these pairs is essential for providing accurate and relevant answers to users.

The proposed system addresses this need by using a two-phase approach to retrieve and re-rank the most relevant answers for a given Stack Overflow question.

The first phase of the proposed system is a base information retrieval task that takes a Stack Overflow question as input and returns the most 1000 relevant answers using the centroids of Word2Vec embeddings. While this method provides a fast and simple approach to retrieve answers, it does not capture the semantics of the query or answers due to its naive nature.

To address this limitation, the proposed system includes a second phase that uses a fine-tuned BERT model to re-rank the answers retrieved in the first phase. The BERT model is trained with negative sampling on a relevant-non relevant binary task, which allows it to better capture the semantics of the question-answer pairs. The re-ranking is used to boost the performance of the IR system, which is evaluated using the Recall@5, 10, and 20 metrics.

The detailed methodology and the results of the experiments are discussed in the following sections of this project.

2. Dataset

The dataset used in this project was collected from the Stack Overflow website. A web scraping process was used to collect more than 300k web pages, one for each question. This phase took approximately one month to complete.

After collecting the data, a pre-processing step was performed to select only the questions that had an accepted answer. This was done to ensure that the information retrieval system would not retrieve non-relevant or useless answers given a question. As a result, only one question-answer pair was extracted per question, even though multiple answers were present for each question. This process resulted in 244095 raw question-answer pairs.

It is important to note that, during the scraping and filtering process, not only the question-answer pairs were collected but also the description of the question that the user submitted. This was done to feed more data to the Word2Vec model to boost its robustness in creating the embeddings.

It is important to note that, during the filtering process, non-answered questions were also removed. This was done to ensure that the information retrieval system would only retrieve answers for questions that had been answered by the community.

3. Pre-Processing

Pre-processing is an essential step in any natural language processing task, and this work is no exception. The pre-processing steps were designed to prepare the raw question-answer pairs for the first Word2Vec naïve information retrieval phase and to ensure that the system would be able to better understand and process the data. The following is a detailed description of the pre-processing steps performed in this work.

- **Lowercasing**: All the words in the question-answer pairs were lowercased to ensure that the system would treat "python" and "Python" as the same token.
- **Removal of new line characters**: The new line characters were removed from the data as they were not relevant for the information retrieval system.
- **Expansion of English contractions:** A pre-defined dictionary was used to expand the English contractions present in the data. This step was taken as a preliminary normalization step and was done to account for the non-formal writing style in Stack Overflow.
- **Removal of particular words**: Words with digits, URLs, and extra spaces were removed as they were not relevant for the information retrieval system.
- **Removal of stop words**: Stop words were removed using the spacy library, which is widely used for natural language processing tasks. Stop words are common words that do not carry much meaning and are often removed from the data to improve the performance of the system.
- **Lemmatization**: Lemmatization was used to reduce the words to their base form. This step was taken because it uses morphological rules, believing that it would preserve more of the semantics of the words than other techniques such as stemming.

These pre-processing steps were performed in the exact order described above and were essential in preparing the data for the Word2Vec information retrieval system.

4. Word2Vec model

The first phase of the information retrieval system in this project is based on the Word2Vec model. The goal of this phase is to create two base vectors for each question-answer pair, using a centroid representation using all the word embeddings for that sentence, which will be used to retrieve the most relevant answers for a given question.

To train the Word2Vec model, 85% of the data was randomly selected (233.111 question-context-answer triples) and 5% of the data was

left out for evaluation purposes (10984 question-answer pairs). This was done to ensure that the evaluation of the system would not be biased.

The Word2Vec model was trained using the following methods:

- the additional context of the question was used
- the embedding size used was 500
- sliding window of 5 words was used
- with the skip-gram model.

The model was trained for about 30 minutes, parallelizing the training on an 8-core CPU, ignoring words that do not appear at least 2 times.

After the model was trained, it was used to create embeddings for each answer in the test set. A kdtree was then built for efficient retrieval and all the query questions were used to retrieve the most similar questions from the tree in the evaluation phase.

The evaluation metric used for the system was recall@5, 10, and 20. This means that the number of times the unique relevant question appears in the top 5, 10, or 20 retrieved questions was counted and divided by the total number of queries.

This Word2Vec model, even though it was trained on a large dataset, it is a naive method to retrieve information, because it does not account for the semantics of the words, therefore it does not perform very well in the retrieval of questions and answers. The re-ranking process using BERT, which is described in the next section, was used to account for this limitation and boost the performance of the system.

5. BERT re-ranking

The second phase of the information retrieval system is based on fine-tuning a vanilla-base BERT model. The goal of this phase is to re-rank the answers retrieved in the first phase of the system.

To account for this task, the weights of a small version of BERT (768 dimension for sentence embedding) were fine-tuned by adding some layers on top of it. The weights of BERT were frozen and only the top layers were trained. A dropout layer was added to prevent overfitting, and a dense linear layer was added to map the 768-dimensional embedding of BERT into 1 dimension. The output was then activated with a sigmoid function to restrain the output in the (0, 1) domain.

The dataset was augmented using negative sampling, as it originally contained only one relevant answer for each question. To train a model that could classify an answer as relevant or non-relevant for a given question, non-related answers were randomly sampled from the training data for each question. This random number of negative samples was used to avoid the model always predicting the most present class in the data and boost the generalizing power of the model.

From 233.111 samples, we obtained 767676 question-answer pairs, where each query appears several times ranging from 1 to 10. Only 1 pair had a label of 1 and the others were labelled as 0. This resulted in a proportion of relevant-non-relevant pairs of 27% - 73%, respectively. With these proportions, the model learns to distinguish the actual correct answer from the remaining non-relevant answers.

During the training phase, the fine-tuned BERT model was trained using a specific set of parameters to optimize its performance. The number of training epochs was set to 6 to ensure that the model had enough time to learn the patterns in the data while avoiding overfitting. The Adam optimization algorithm was used as the optimizer, which is a well-known method for training deep learning models, with a learning rate of 2e-5. This learning rate value was chosen after some experimentation, as it provided a good balance

between the speed of convergence and the final performance of the model.

The loss function used for this task was the binary cross-entropy with logit, which is widely used for binary classification problems. The batch size was set to 128 question-answer pairs, which is a common value for deep learning models. This batch size was chosen to make the most efficient use of the available GPU memory and computational resources.

A clipping gradient method was used to avoid the explosion of gradients during training. This method normalizes the gradients and clips them to a maximum value of 1. This method is known to be effective in preventing the gradients from becoming too large, which can cause the model to diverge.

The representation of the question-answer pairs used in this project was the sum of CLS and SEP token embeddings, where CLS represents the beginning of the sentence and SEP represents the end of the sentence. This representation was chosen because it is a standard way of representing input text in BERT models and it has been shown to be effective in many natural language processing tasks.

The training was performed using a NVIDIA GeForce GTX 1070 for about 3 days.

The validation set was extracted from the train set with 67675 pairs, to evaluate the performance of the model and to see how well it generalizes to new data. The results of the classification task are presented in the following table:

Model	Accuracy	Precision	Recall	F1
BERT	87.3	89.7	59.8	71

Table 1. In this table we can see the performance of the BERT model as a classifier.

These results indicate that the fine-tuned BERT model was able to accurately classify most of the question-answer pairs as relevant or non-relevant. The high accuracy and precision scores suggest that the model is good at correctly identifying relevant pairs, while the lower recall score suggests that there may be some relevant pairs that the model is not correctly identifying. The F1 score, which is a balance between precision and recall, is also relatively high, which further supports the overall effectiveness of the model.

The method and the results of this re-ranking process are discussed in the following sections of this project.

6. Method

The proposed information retrieval system consists of two main parts: a first answer retrieval phase and a re-ranking phase. The system takes a Stack Overflow question as input from the user. In the first phase, the base information retrieval system is designed to return the most relevant answers for that question, using Word2Vec centroids. The Word2Vec model is trained on a subset of the data, using the additional context of the question and creating a vector representation for each answer in the test set. A KDTREE is built for efficient retrieval, and all the query questions are used to retrieve the first 1000 most similar questions from the tree.

In the second phase, the BERT model is used to re-rank the top 1000 retrieved answers from the first phase. The BERT model gives a score between 0 and 1 for each of the 1000 query-answer pairs (where the query is always the same). These scores are then used to re-rank the answers, with the highest scoring answers appearing first in the final list of results. The results, which are presented in the next section, suggest that the system better capture the semantics of the question-answer pairs and boost the performance of the IR system.

7. Results

The performance of the IR system was evaluated using the recall@5, 10, and 20

metrics. This metric measures the number of times the unique relevant question appears in the top 5, 10, or 20 retrieved results, divided by the total number of queries. The results from the Word2Vec base IR system and the full system are reported in table below:

Recall top k	Word2Vec	W2V + BERT
Recall@5	18.1	21.8
Recall@10	22.8	29.8
Recall@20	28.7	35.0

Table 2. In this table we can see the improvements in performance of the IR system given by the BERT re-ranker.

These results demonstrate that the base IR system, which uses the centroid of the phrases of a Word2Vec model, it struggles to capture the semantics of the question or answers and performs worse.

The results from this two-phase approach demonstrate a significant improvement in the performance of the IR system. It is worth noting that the recall values obtained from the complete system are a bit low, however it is important to take in consideration that this dataset is very large, containing many very similar questions and answers.

8. Comparison

To compare the performance of the IR system with the available research, this paper [1] was chosen to have an idea of the goodness of the proposed work. Although they use quite the same method, using BM25 has starting point and then fine-tuning a BERT model, they evaluate their method with two different datasets: MS MARCO and TREC-CAR. As regards this work, a custom dataset has been used and for this reason it will not be a direct comparison of performance.

Using the MAP metric as a comparison method, only the second dataset is considered in this evaluation (TREC-CAR). To reproduce the exact same metric the authors used, the Average Precisions for the first 11 ranks was computed for every question-answer pair retrieval, then the simple arithmetic mean was computed between these values.

The performance of their work is reported in the table below, as well as the results obtained in this work:

Model	MAP	Dataset
BM25 + BERT [1]	31.0	TREC-CAR
W2V + BERT [this work]	19.4	StackOverflow QA

Table 3. In this table we can see the comparison in performance of the two proposed IR system, although they have been tested and fine-tuned on different datasets.

However, as suggested in the previous section, the stack overflow corpus have a huge amount of similar and very related question-answer pairs, which talk about the same thing but just a bit shaded. This brings the system to give a very high similar score to a lot of answers and this cause the unique right answer to rank low most of the time.

9. Conclusions

In this work it was proposed a novel approach for an information retrieval system in the domain of Stack Overflow question-answers. Our system consisted of two main parts: a first answer retrieval phase using Word2Vec and a re-ranking phase using a fine-tuned BERT model. The results of our system were evaluated using the Recall@5, 10, and 20 and MAP. The use of BERT allowed for the capture of the semantics of the question-answer pairs and thus, improved the performance of the IR system. Overall, this approach demonstrated promising results and highlighted the effectiveness of the two-phase approach in the information retrieval task.

In comparison to the work done in [1], our approach was applied to a custom dataset, while they used TREC-CAR as their dataset. In their work, Nogueira et al. achieved a MAP of

31.0 using the BERT base model, which is the same model used in this work. Although we do not have a direct comparison of our results to theirs, due to the difference in datasets, our results can be seen as a promising indication of the effectiveness of the proposed approach in improving the performance of information retrieval systems.

10. Future work

In future work, it could be considered to cluster questions or arguments into fewer classes to consider more relevant documents for each cluster. This would allow for a more thorough search for a particular answer on the Stack Overflow website, as it is often the case that a problem can be solved through consulting more than one question-answer pair. Additionally, it could be beneficial to extract more than one answer per question, but careful consideration would need to be given to the popularity of the question to ensure that irrelevant information is not included.

References

[1] Nogueira, Rodrigo, and Kyunghyun Cho. "Passage Re-ranking with BERT." arXiv preprint arXiv:1901.04085 (2019).