Seeing Double: Detecting Duplicate Questions using Sentence-BERT

Marko Opačić, Javier Salvatierra

University of Zagreb, Faculty of Electrical Engineering and Computing
Unska 3, 10000 Zagreb, Croatia
{marko.opacic,javier.corchado}@fer.hr

Abstract

Identification of duplicate questions is an important task for many Q&A platforms, as it can help enhance the user experience and ensure that valuable insights are not split across multiple locations. In this paper, we investigate the performance of Sentence-BERT models on the Quora Question Pairs dataset. We compare the results with the wining solution from the 2017 Quora Question Pairs competition held on the Kaggle platform. We find that the Sentence-BERT models outperform previous competitive architectures on the dataset.

1. Introduction

There are many online forum platforms where users can ask and answer questions on various topics. Especially as the platform grows, different users tend to ask the same questions. If these duplicate questions are not properly reconciled it can make the search for an answer tedious and longer than necessary, leaading to a poor user experience. Similar problems occur in customer support operations, as a significant portion of customer questions will already have known answers. The scale of large online forums also requires the use of automated solutions to identify these duplicate questions.

The Quora Question Pairs dataset was released in 2017 and contains 404k labeled question pairs. A Kaggle competition was organized soon after the public release of the dataset and most of the top performing models used LSTM architectures, including the winning solution, as they were state-of-the-art at the time. Since then, pretrained Transformer models have achieved new state-of-the-art results on a multitude of tasks across the field of NLP. In this paper, we examine the performance of Sentence-BERT models on the Quora Question Pairs dataset, and compare our results to the winning solution of the 2017 Quora Question Pairs

In Section 2. we provide an overview of related work. We describe the Quora Question Pairs dataset in Section 3.. The models included in the final results are described in Section 4.. The training procedure for the Sentence-BERT models is reported in Section 5.1., with results presented in Section 5.2.. Finally, we summarize our findings and conclusions in Section 6..

2. Related Work

The problem of identifying duplicate questions has similarities to other NLP tasks, such as paraphrase detection and semantic similarity detection. Traditional machine learning approaches use algorithms such as SVM with handpicked features including n-gram overlaps, part-of-speech agreement, verb similarity and others (Dey et al., 2016). More recently, deep learning approaches have proved to be very effective in a variety of NLP tasks. Most approaches for detecting sentence similarities use a Siamese architecture, which involves producing encoded representations for each of the two input sentences, which are subsequently processed for classification, usually through some form of distance metric. Some examples include the Siamese GRU (Homma et al., 2016), Siamese MaLSTM (Imtiaz et al., 2020) and Siamese BiLSTM (Fradelos et al., 2023). Convolutional neural networks (CNNs) are leveraged by several authors, also in a Siamese setting (Bogdanova et al., 2015; Prabowo and Herwanto, 2019).

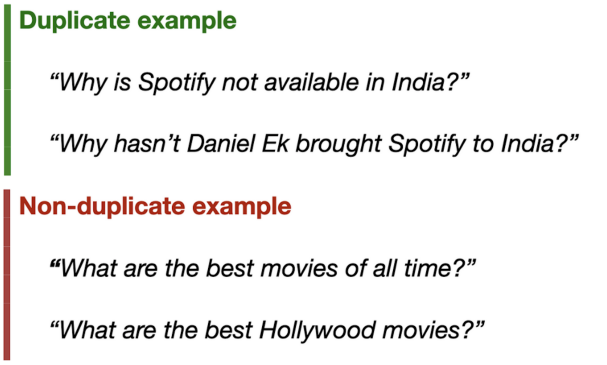

Figure 1: Examples from the Quora Question Pairs dataset.

The Quora Question Pairs dataset by Iyer et al. (2017) has been a valuable resource for research in this area. As part of a Kaggle competition using the dataset, a Siamese LSTM architecture with attention mechanisms was the winning solution (Dadashov et al., 2017). The winning Siamese LSTM architecture is described in Section 4.2..

In the past few years, pretrained Transformer models have set a new benchmark on sentence-pair regression tasks (Devlin et al., 2018). Although the original BERT model performs well on such tasks, it has some limitations, such as requiring both sentences as inputs to determine their similarity, which results in a large computational overhead. The Sentence-BERT models significantly reduce the computational load and achieve state-of-the-art results on semantic textual similarity tasks (Reimers and Gurevych, 2019).

3. Data

The Quora Question Pairs (QQP) dataset consists of 404k question pairs, along with labels indicating whether the questions are duplicates or not. A duplicate and a non-

Table 1: Dataset split characteristics, showing the number of examples and the percentage of duplicates in each set.

Set	No. of examples	Duplicate percentage
Train	243k	37.25%
Dev	80k	35.05%
Test	80k	40.38%

duplicate example are shown in Figure 1.

We use the same dataset split as the SBERT paper, which is a 60/20/20 split resulting in 243k training examples, 80k dev examples and 80k test examples (Reimers and Gurevych, 2019). The splits are not stratified, as different percentages of duplicates are present between the sets. The percentage of duplicates per set is presented in Table 1.

4. Models

In this section, we describe the models used in the experiments. Cosine similarity is used as the simplest baseline, while the winning solution from the 2017 Quora Question Pairs competition on Kaggle serves as the baseline we are attempting to outperform using Sentence-BERT.

4.1. Cosine similarity

The input questions are first tokenized using spaCy, after which embeddings are calculated for each token. GloVe embeddings are used for this purpose, specifically the 300-dimensional vectors trained on a corpus 6B tokens from Wikipedia 2014 and Gigaword 5 (Pennington et al., 2014). Aggregate embeddings for each question are derived by summing the embedding vectors for each token in the question element-wise. The cosine similarity is then calculated for the aggregate embeddings to obtain a similarity score. The similarity score is fed into a simple binary classifier which produces the final output label. This classifier is implemented as a feed-forward layer with 2 inputs and a single output.

4.2. Siamese LSTM

The Siamese LSTM (SLSTM) is the winning solution of the 2017 Kaggle competition on the QQP dataset. It employs two identical LSTMs that process two input questions simultaneously and then compares their final vector representations to determine their similarity. The final model uses an ensemble of the Siamese network with an LSTM and a Sequence-to-Sequence LSTM with Attention. This shows improved performance over the single Siamese LSTM model, even though the Seq2Seq LSTM with Attention performs significantly worse on its own (Dadashov et al., 2017). We use the reported results for the ensemble model on the QQP dev set for this model for comparison with our results in Section 5.2..

4.3. Sentence-BERT

Sentence-BERT (SBERT) is a method for creating semantically meaningful sentence embeddings (vector representations) using transformer-based models like BERT. It fine-tunes BERT to generate fixed-size representations of input sentences that capture their semantic meaning. SBERT is trained using a siamese or triplet network architecture, where it learns to map similar sentences closer together in the embedding space.

In our experiments we used the pretrained RoBERTa base model (RoBERTa) described by Liu et al. (2019), as well a distilled version of the same model (DistilRoBERTa). The distilled version uses the same distillation process used for DistilBERT (Sanh et al. (2019).

It is worth noting that with pretrained large language models there is a risk of test set contamination. This means that part of the test set on which the model is evaluated is used during model training. This violates the assumption that the model is evaluated on unseen data, and undermines the integrity of the results. Specifically for the RoBERTa and DistilRoBERTa models used, their training data does not include the QQP dataset, so we have no such concerns.

5. Experiments

In this section we explain the training process for the models used in the experiments. We then provide an overview and a brief discussion of the results.

5.1. Training

Two SBERT models were trained and evaluated, the RoBERTa base model and it's distilled version, the DistilRoBERTa base model. The models were fine-tuned as cross encoders, where both sentences were fed to the model as input and a similarity score between 0 and 1 was produced as output. In this setup, the output of the Transformer layer is fed to a feedworward layer with two output classes which correspond to whether the question pair is a duplicate or not. The training for both models was performed in the same manner, using the same parameters. For the SLSTM model we used the reported results from the competition solution (Dadashov et al., 2017).

5.1.1. Loss function

Binary cross entropy loss is used as the loss function:

$$L(\mathbf{p}, \mathbf{y}) = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log\left(\hat{p_i}\right) + \left(1 - y_i\right)\log\left(1 - \hat{p_i}\right)\right]$$

where $p_i \in [0, 1]$ is the predicted output of the model, representing a similarity score between the two input questions. Also:

$$\hat{p_i} = \sigma(p_i)$$

$$\sigma(p_i) = \frac{1}{1 + e^{-p_i}}$$

5.1.2. Parameters and training procedure

Following advice from Gkouti et al. (2024), we focused only on tuning the learning rate. For selecting the learning rate we tested values of 4e-5, 3e-5 and 2e-5, similar to the original BERT paper (Devlin et al. (2018). Higher learning rate values were avoided to circumvent the problem of catastrophic forgetting of the pretraining data (Sun et al., 2019). The learning rate for the model with the highest F1 score on the dev set was selected, and the results are included in Section 5.2.

Table 2: Hyperparameters used for training SBERT models

Parameter name	Value
Batch size	16
Epochs	4
Learning rate	2e-5

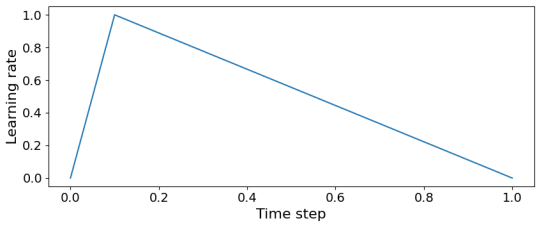

Figure 2: Linear warmup learning rate schedule

The batch size used was 16 and the models were trained for 4 epochs, with performance evaluation on the dev set every 5000 steps. Model checkpoints were also saved every 5000 steps. For the test set results the model checkpoint with the best performance on the dev set was selected. The parameters mentioned are listed in Table 2.

We used linear warmup and linear learning rate decay as described in Devlin et al. (2018), with 10% of the total training steps across all epochs used for warmup. The resulting learning rate schedule is shown in Figure 2.

Both models were trained on a system with a single Nvidia T4 GPU.

5.2. Results

The results are presented in Table 3. All other models significantly outperform the simple cosine similarity baseline using GloVe embeddings. Expectedly, the SBERT models outperform the Siamese LSTM on accuracy and F1 scores

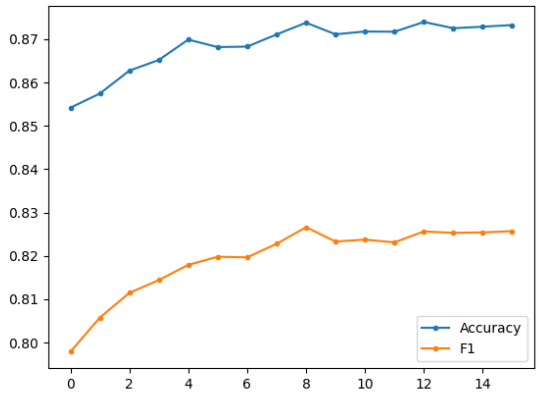

Figure 3: Accuracy and F1 score during training - DistilRoBERTa

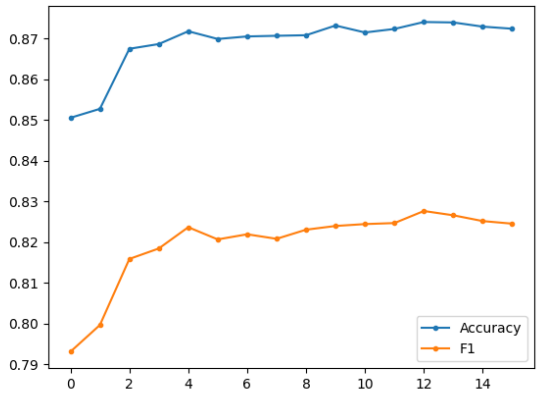

Figure 4: Accuracy and F1 score during training - RoBERTa

on both the dev and the test set.

Additionally, we can see that DistilRoBERTa has similar performance to RoBERTa, even with a lower number of training parameters and with less training time used. An interesting result to note is that DistilRoBERTa performed better on the dev set while RoBERTa performed better on the test set. This may be interpreted as DistilRoBERTa converging earlier with the RoBERTa model having a higher peak performance, but it is not a conclusive or statistically proven result. The above shows that, at least in our case, the DistilRoBERTa model comes with important advantages with regards to resource utilization and computing time without much sacrifice, if any, in performance.

The accuracy and F1 scores during training are shown for DistilRoBERTa and RoBERTa models in Figure 3 and Figure 4, respectively.

6. Conclusion

The task of identifying duplicate questions is of particular interest to many Q&A platforms, as it can help enhance the user experience. The Quora Question Pairs dataset is an important resource in this area of research. Previous best-performing architectures on this task used Siamese LSTM based models. An example of such an architecture is the Siamese LSTM model, which won the 2017 Quora Question Pairs Kaggle competition. We investigated the performance of Sentence-BERT models on this task, and compared them to the winning solution. We find that Sentence-BERT models outperform the Siamese LSTM architecture on the Quora Question Pairs dataset. This is in line with expectations since newer pretrained Transformer architectures have dominated the field as of late.

References

Dasha Bogdanova, Cicero dos Santos, Luciano Barbosa, and Bianca Zadrozny. 2015. Detecting semantically equivalent questions in online user forums. In *Proceedings of the nineteenth conference on computational natural language learning*, pages 123–131.

Table 3: Accuracy and F1 scores on the dev and test sets for the evaluated models.

	Dev set		Test set	
Model	F1 score	Accuracy	F1 score	Accuracy
Cosine similarity	36.4	37.1	35.8	36.2
SLSTM	79.5	83.9	79.5	83.8
RoBERTa	82.5	87.2	83.9	86.6
DistilRoBERTa	82.7	87.4	82.2	86.1

Elkhan Dadashov, Sukolsak Sakshuwong, and Katherin Yu. 2017. Quora question duplication.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Kuntal Dey, Ritvik Shrivastava, and Saroj Kaushik. 2016. A paraphrase and semantic similarity detection system for user generated short-text content on microblogs. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2880–2890.

Georgios Fradelos, Isidoros Perikos, and Ioannis Hatzilygeroudis. 2023. Using siamese bilstm models for identifying text semantic similarity. In *IFIP International Conference on Artificial Intelligence Applications and Innovations*, pages 381–392. Springer.

Nefeli Gkouti, Prodromos Malakasiotis, Stavros Toumpis, and Ion Androutsopoulos. 2024. Should i try multiple optimizers when fine-tuning pre-trained transformers for nlp tasks? should i tune their hyperparameters? *arXiv preprint arXiv:2402.06948*.

Yushi Homma, Stuart Sy, and Christopher Yeh. 2016. Detecting duplicate questions with deep learning. In *Proceedings of the International Conference on Neural Information Processing Systems (NIPS*, pages 25964–25975.

Zainab Imtiaz, Muhammad Umer, Muhammad Ahmad, Saleem Ullah, Gyu Sang Choi, and Arif Mehmood. 2020. Duplicate questions pair detection using siamese malstm. *IEEE Access*, 8:21932–21942.

Shankar Iyer, Nikhil Dandekar, and Kornél Csernai. 2017. First quora dataset release: Question pairs. Accessed: 2024-06-06.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Damar Adi Prabowo and Guntur Budi Herwanto. 2019. Duplicate question detection in question answer website using convolutional neural network. In *2019 5th International conference on science and technology (ICST)*, volume 1, pages 1–6. IEEE.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In *Chinese computational linguistics: 18th China national conference, CCL 2019, Kunming, China, October 18–20, 2019, proceedings 18*, pages 194–206. Springer.