Natural Language Processing

Term Project Report

Semantic Textual Similarity of Quora question pairs

Bhavana Srushti[1] Sai Teja Karlapudi[2]

Abstract

The degree to which two texts have similar meanings is gauged by semantic textual similarity. Based on the words and phrases used in the questions and their context, semantic textual similarity can be used to assess how similar the meanings of two Quora question pairings are. This can be helpful for various natural language processing jobs, such as finding duplicate or similar queries on Quora. The objective of the project is to determine whether or not the question pairings have similar intentions. To measure the semantic relationship between two entities using Pre-trained models like RoBERTa, BERT and Distill BERT. Evaluate the performance of different models using F1 score as the metrics.

1 Introduction

A place to learn and share knowledge about anything is Quora. It serves as a forum for queries and connections with experts who offer insightful observations and thorough responses. People are better able to grasp the world and learn from one another as a result. It's hardly surprising that many questions on Quora are similar in wording given that over 100 million people visit the site each month. Multiple inquiries with the same objective can make readers feel as though they must respond to various variations of the same inquiry, while also making seekers spend more time looking for the best solution to their problem. Canonical questions are highly valued on Quora because they provide active writers and seekers a better experience and more long-term value.

Given how challenging this endeavor is, we believe the Quora dataset presents an intriguing challenge. We selected the pre-trained RoBERTa, BERT, and Distill BERT models as the basis for our analysis. These are widely used models in the field of natural language processing, and they have been shown to have strong performance on various tasks, such as language modelling, text classification, and question answering.

Pre-trained models are models that have been trained on a large corpus of text data, and they can be fine-tuned for specific tasks or datasets. This is useful because it allows us to leverage the knowledge and expertise that has been built into the pre-trained models, and it can save time and resources compared to training a model from scratch.

That is using pre-trained models like RoBERTa, BERT, and Distill BERT, we sought to quantify the semantic relationship between two entities and ascertain whether the question pairs have comparable intentions. We also sought to compare the effectiveness of various models using the F1 score as a metrics.

There are many different methods for measuring semantic textual similarity, including lexical matching, syntactic parsing, and semantic parsing, and semantic analysis. These methods can be applied to a variety of tasks, such as identify duplicate documents, detecting plagiarism, and evaluating the quality of text generation systems.

2 Related Works

Saric et al. (2012) [1] proposed a system for measuring semantic text similarity using multiple content similarity measures, including WordNet path similarity and vector space similarity. They evaluated their system on a dataset of English and Croatian texts, and found that it outperformed several baselines.

Yeh and Agirre (2012) [2] proposed a method for measuring semantic textual similarity based on simple semantic features, such as the overlap of named entities and WordNet synonyms. They evaluated their method on a dataset of English and Spanish texts, and showed that it achieved good performance.

Han et al. (2012) [3] proposed a semantic textual similarity system based on distributional similarity and WordNet path similarity. They evaluated their system on a dataset of English texts, and found that it performed well compared to several baselines.

Liu et al. (2010) [4] proposed a method for measuring semantic textual similarity based on matching n-grams (i.e. sequences of n words) using linear programming. They evaluated their method on a dataset of English and Chinese texts, and found that it achieved good performance in terms of precision, recall, and F-measure.

Wu et al. (2013) [5] proposed a method for measuring semantic textual similarity based on dependency trees, which represent the syntactic relationships between words in a sentence. They evaluated their method on a dataset of English and German texts, and showed that it outperformed several baselines.

3 Experiments

3.1 Data preparation:

We have first split the Quora question pairs dataset into training and testing sets. This is a common practice in machine learning, where the training set is used to train the model, and the testing set is used to evaluate the model's performance. By splitting the dataset into separate sets, we can ensure that the model is not overfitting to the training data, and that it is able to generalize to unseen examples in the testing set.

After splitting the dataset, we removed the columns that are not relevant to our analysis. For example, if we are only interested in the similarity of the questions, we can remove the columns that contain the answers, as they are not necessary for our analysis.

We also removed any missing or incomplete values from the dataset. This is important because missing values can cause errors during the model training and evaluation process, and they can also affect the quality of the results. By removing the missing values, we can ensure that the dataset is clean and complete, and that it is suitable for use in our analysis.

3.2 Model selection:

We selected the pre-trained RoBERTa, BERT, and Distill BERT models as the basis for our analysis, as these are widely used models for natural language processing tasks.

3.3 Word embeddings:

To generate the word embeddings, we passed the questions in the training and testing sets through the RoBERTa, BERT, and Distill BERT models. These models take the words in the questions as input, and they output a vector representation for each word. The vectors capture the meaning of the words in the context of the model's training data, and they can be compared to calculate the similarity between the words.

By generating the word embeddings for the questions in the training and testing sets, we are able to represent the meaning of the words in a numerical format, which can be used to calculate the semantic similarity between the questions. It also enables us to evaluate the performance of the models on the task of semantic textual similarity.

We have encountered some errors while attempting to fine-tune the pre-trained language models. We have faced this issues while tokenizing the question pairs. Therefore we are not able to work on the model training part of the project. We have tried resolving the issue, but not able to fix the issue. We have commented out the part of the code that is being errored.

3.4 Threshold setting:

We set the threshold for the similarity between two questions as 0.6, based on the distribution of the similarity scores in the dataset. This threshold is used to determine whether two questions are considered similar or not, based on their similarity score.

To set the threshold, we first calculated the similarity scores between all pairs of questions in the dataset. This gave us a distribution of the similarity scores, which we used to determine the appropriate threshold. In this case, we chose a threshold of 0.6, which is a reasonable value based on the distribution of the similarity scores in the dataset.

The choice of threshold is an important consideration, as it determines the sensitivity of the model to the similarity between the questions. If the threshold is set too low, the model may be too strict, and it may not consider many pairs of questions as similar. On the other hand, if the threshold is set too high, the model may be too lenient, and it may consider many pairs of questions as similar. By setting the threshold based on the distribution of the similarity scores in the dataset, we can ensure that it is reasonable and appropriate for our analysis.

3.5 Prediction:

We used the word embeddings and the similarity threshold to generate predictions for the question pairs in the testing set. For each pair of questions, we calculated the similarity score using the word embeddings and the similarity threshold, and we assigned a value of 0 or 1 to the pair, depending on the similarity score.

A value of 0 indicates that the questions are not similar, and a value of 1 indicates that they are similar. This allows us to make predictions about the similarity of the question pairs in the testing set, based on the word embeddings and the similarity threshold.

To generate the predictions, we used the word embeddings generated by the RoBERTa, BERT, and Distill BERT models, and we applied the similarity threshold to each pair of questions. We then compared the predictions to the ground truth labels in the testing set, and we used this comparison to evaluate the performance of the models on the task of semantic textual similarity.

3.6 Evaluation:

We compared the predictions for the question pairs in the testing set to the ground truth labels in the dataset. The ground truth labels are the correct answers for the question pairs, and they indicate whether the questions are similar or not.

We used the predictions and the ground truth labels to calculate the F1 score for each model. The F1 score is a measure of a model's performance in a classification task, and it is calculated as the harmonic mean of the precision and recall of the model. Precision is the number of true positive predictions divided by the total number of positive predictions, and recall is the number of true positive predictions divided by the total number of actual positive examples.

By calculating the F1 score for each model, we were able to evaluate their performance on the task of semantic textual similarity. The higher the F1 score, the better the model is able to make accurate predictions and capture the similarity between the question pairs in the testing set. This allows us to compare the performance of the different models and determine which model is the most effective at this task.

4 Results

Based on the given F1 scores, it appears that the RoBERTa model has the highest performance among the three models. The F1 score is a measure of a model's performance in a classification task, and the higher the score, the better the model is able to make accurate predictions.

In this case, the RoBERTa model has an F1 score of 0.6728, which is higher than the scores of the BERT (0.6182) and Distill BERT (0.6638) models. This suggests that the RoBERTa model is able to make more accurate predictions and capture a higher proportion of the actual positive examples than the other two models.

5 References

[1] Saric, F., Glavas, G., karan, ., Snajder, J., and Basic ,B.D. “Takelab: Systems for measuring semantic text similarity”. In proceedings of the First Joint Conference on Lexical and computational semantics, SemEval-2012.

[2] Yeh,E. and Agiree,E. “Srubic: Simple semantic features for semantic textual similarity”. In Proceedings of the First joint conference on Lexical and Computational Semantics, SemEval2012.

[3] Han, L., Kashyap, A., Finin, T., May_eld, J., and Weese, J. “Umbc ebiquity-core: Semantic textual similarity systems. In Second Joint Conference on Lexical and Computational Semantics” (*SEM).

[4] Liu,C., Dahlmeier, D., and Ng,H.T. “Tesla: translation evaluation of sentences with linear-programming-based analysis”. In Proceedings of the joint fifth workshop on Statistical Machine Translation and MetricsMATR.

[5] Wu,X., Yu, H., and Liu, Q. DCU participation in WMT2013 metrics task. In Proceedings of the Eighth Workshop on Statistical machine Translation, pages 435-439, Sofia, Bulgaria.

GitHub Link: https://github.com/saitejakarlapudi/NLP-Term-Project