Comparing Sentence Similarity Methods

Vladimir Saberullin

April 2022

Abstract

This paper presents several approaches used to calculate the semantic similarity of sentences: starting with the calculation of the cosine distance between the average embeddings according to the words in the sentence, and ending with the BERT classifier. Project link: `https://github.com/VldSab/NLP/tree/main/Quora-Question-Pairs`.

1 Introduction

As stated in [Xiaofei Sun and ChunFan., 2022], measuring sentence similarity is a long-standing task in NLP. The task aims at quantitatively measuring the semantic relatedness between two sentences, and has wide applications in text search [Mamdouh Farouk and Bollegala., 2018], natural language understanding [language inference., 2009] and machine translation [Mingming Yang and Zhao., 2019]. The main task of current paper is to choose the effective baseline for semantic similarity task. I tried to apply the existing supervised learning approaches to assess semantic similarity and compare them. I have tested the following methods:

1. Using ready-made w2v embeddings to get sentence embedding as an average of n word vectors:

$$W_i = \frac{\sum_{i=1}^{n} w_i}{n};$$

2. Using two models: Bidirectional LSTM as sentence encoder and FeedForward NN as simple surrogate classifier;

3. Using a classifier based on the BERT.

All methods were tested on the Quora Question Pairs dataset.

1.1 Team

Vladimir Saberullin prepared this document.

2 Related Work

There are many approaches to solving this problem. The main ones are described below.
1. Using WordNet semantic graphs: A Method for Measuring Sentence Similarity and its Application to Conversational Agents [Yuhua Li, 2004].
2. Statistics-based methods for measuring sentence similarity: Bag-of-words (BoW) [Yuhua Li and Crockett, 2006], term frequency inverse document frequency (TF-IDF) [Luhn., 1957], BM25
[Stephen E Robertson, 1995], latent semantic indexing (LSI)
[Scott Deerwester and Harshman., 1990] and latent dirichlet allocation (LDA) [David M Blei and Jordan., 2003].
3. Word Distance Based Methods: Another approach to measure sentence similarity is to calculate the cost of transforming from one sentence to another, and the smaller the cost is, the more similar two sentences are. This idea is implemented by the Word Mover's Distance (WMD)
[Matt Kusner, 2015], which measures the dissimilarity between two documents as the minimum amount of distance that the embedded words of one document need to transform to words of another document. More recently, [Sho Yokoi and Inui., 2020] proposes to disentangle word vectors in WRD has shown significantly performance boosts over vanilla WMD.
4. Supervised learning approaches. Embeddings, classification, attention models, transformers. Sentence embeddings are high-dimensional representations for sentences. They are expected to contain rich sentence semantics so that the similarity between two sentences can be computed by considering their sentence embeddings via certain metrics such as cosine similarity. Works: FastText, Skip-Thought vectors [Ryan Kiros, 2015], Smooth Inverse Frequency (SIF) [Sanjeev Arora and Ma., 2016], Sequential Denoising Autoencoder (SDAEs) [Felix Hill, 2016], InferSent [Alexis Conneau, 2017], Quick-Thought vectors [Logeswaran and Lee., 2018] and Universal Sentence Encoder [Daniel Cer, 2018]. The BERT-based scores [Tianyi Zhang, 2020]; [Thibault Sellam, 2020].
5. Unsupervised learning, contextual approaches. Learning word representations given its contexts with the assumption that the meaning of a word is determined by its context. In Sentence Similarity Based on Contexts [Xiaofei Sun and ChunFan., 2022] is used contextual model, which predicting the probability of appearance sentence in specific context. Based on this work, if we have sentences pairs, we can determine their semantic similarity by compare scores from contextual model using surrogate classifier-model.

3 Model Description

1. First approach: words embedding and cosine similarity.
SpaCy and Gensim pretrained models were used for this approach to get words embeddings. To calculate sentence vector representation was taken the average

of all words embeddings in current sentence. To calculate the semantic, the calculation of the cosine distance between two sentence vectors is used.
2. Second approach: sentence encoder (BiLSTM) + simple classifier (FeedForward NN) 1.

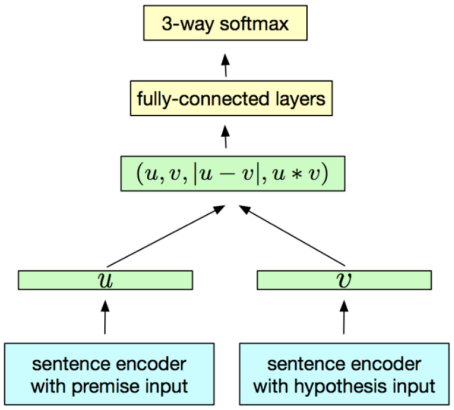

Figure 1: Second approach architecture

First model was written in PyTorch and has following parameters:

1. Nuber of hidden layers - 3;
2. Number of classes - 6;
3. Batch size - 256;
4. Input size - 300;
5. Hidden size - 128;
6. Length of sequence - 25.

All words in dataset have been vectorized with gensim W2V. Then encoder model has been trained. Now, in evaluation mode, we can vectorise sentences with this model, by getting output from N-1 model layer. This model was trained on ODS dataset for sentiment analysis task.

Second model is the simple fully connected neural network for classification.

1. Nuber of hidden layers - 3;
2. Number of classes - 2;
3. Dropout on first layer - 0.1.

We got encoding vectors v1 and v2 for each sentence from first model. Then constructed dataset with features: v1, v2, v1*v2, |v1 - v2|. Then we flattened each row and put to classifier.
Third approach: BERT classifier. A pre-trained BERT architecture model was chosen - "prajjwal1/bert-tiny" with Hugging Face. This is a lightweight version of the BERT model. Two sentences separated by the symbol '.' were submitted to the input of the model. The model should have attributed this line either to class 0 - "lack of semantic similarity", or to class 1 - "semantically close".

4 Dataset

Toxic Comment dataset was chosen for encoder training. This dataset is for sentiment analysis task. It has 6 lable classes and it is available here - Kaggle[1].

	Train	Valid
Train pairs	40,000	5000

Table 1: Toxic Comment Classification

Main dataset for semantic similarity task is Quora Question Pairs[2]. It has 2 classes.

	Train	Valid	Test
Train pairs	384,075	20,215	2,000,000

Table 2: Quora Question Pairs

5 Experiments

5.1 Metrics

F1-score was used as main metric for all experiments:

$$Precision = \frac{TP}{TP + FP};$$

$$Recall = \frac{TP}{TP + FN};$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall}.$$

[1]Toxic Comment Classification dataset - Kaggle Comment dataset
[2]Quora Question Pairs dataset - Kaggle Quora dataset

5.2 Experiment Setup

There were 3 experiments for each approach. Quora Question dataset has 2,000,000 unlabeled test samples, so the validation set was used as indicator. Data split - 95%/5%, with 384,075 train and 20,215 validation samples. Encoder dataset has only 40,000 training samples and 5,000 validation. Models hyper-parameters were described in third section 3

6 Results

First approach: w2v embeddings.
F1-score - 0.65;
This method does not take into account dependencies between words in a sentence. Thus this approach is not robust to negative particles for example. More over we loose a lot of information about order of words in sentence. That is why this approach has such poor accuracy.
Second approach: Bidirectional LSTM as sentence encoder and FeedForward NN as simple surrogate classifier.
Encoder model was trained on 40,000 samples and has accuracy score - 0.48 on validation set.

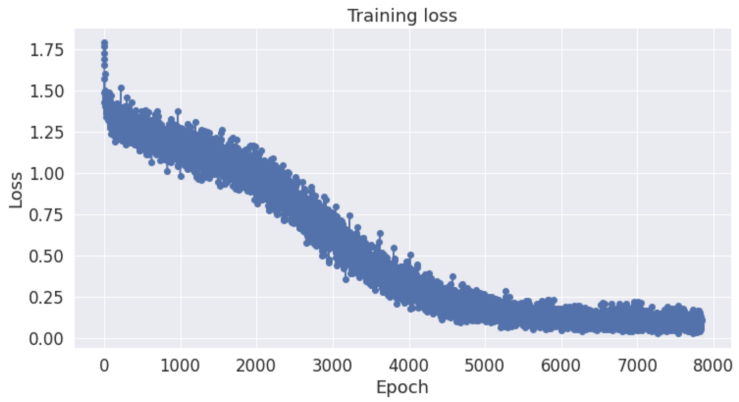

Figure 2: LSTM loss

The classifier model gave an F1-score of 0.38. This result worse than constant algorithm. The main assumption about such results is little data for encoder model and quality of this data: Toxic Comment Classification dataset from ODS has only 40,000 samples with rough language. It is not good idea to use model trained on such data as encoder.
Third approach: BERT classifier.
This approach gave the best F1-score - 0.8583.
There is another metric was used at Quora Question Pairs Kaggle competition. There is log-loss, and competition winner has 0.12 loss. My BERT model gives

Figure 3: BERT F1

only 4.89. I have some assumption about methods I can use to improve model score (including using a larger not tiny model and increasing epochs).

7 Conclusion

The BERT model gives a baseline that is hard to beat. Also there is a lot of methods and opportunities to improve each of the models in this paper, but main task is done: we can definitely say that of these three approaches, the best is to use pretrained BERT model.

References

[Alexis Conneau, 2017] Alexis Conneau, Douwe Kiela, H. S. L. B. A. B. (2017). Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint*, page 1705.02364.

[Daniel Cer, 2018] Daniel Cer, Yinfei Yang, S. y. K. N. H. N. L. R. S. N. C. M. G.-C. S. Y. C. T. Y.-H. S. B. S. R. K. (2018). Universal sentence encoder.

[David M Blei and Jordan., 2003] David M Blei, A. Y. N. and Jordan., M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

[Felix Hill, 2016] Felix Hill, Kyunghyun Cho, A. K. (2016). Learning distributed representations of sentences from unlabelled data. *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, page 1367–1377.

[language inference., 2009] language inference., N. (2009). Graph matching based semantic search engine.

[Logeswaran and Lee., 2018] Logeswaran, L. and Lee., H. (2018). An efficient framework for learning sentence representations. *arXiv preprint*, page 1803.02893.

[Luhn., 1957] Luhn., H. P. (1957). A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of research and development*, 1(4):309–317.

[Mamdouh Farouk and Bollegala., 2018] Mamdouh Farouk, M. I. and Bollegala., D. (2018). Graph matching based semantic search engine. *Research Conference on Metadata and Semantics Research*, page 89–100.

[Matt Kusner, 2015] Matt Kusner, Yu Sun, N. K. K. W. (2015). From word embeddings to document distances. *International conference on machine learning*, page 957–966.

[Mingming Yang and Zhao., 2019] Mingming Yang, Rui Wang, K. C. M. U. E. S. M. Z. and Zhao., T. (2019). Sentence-level agreement for neural machine translation. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, page 3076–3082.

[Ryan Kiros, 2015] Ryan Kiros, Yukun Zhu, R. R. S. R. Z. R. U. A. T. S. F. (2015). Skip-thought vectors. *Advances in neural information processing systems*, 28.

[Sanjeev Arora and Ma., 2016] Sanjeev Arora, Y. L. and Ma., T. (2016). A simple but tough-to-beat baseline for sentence embeddings.

[Scott Deerwester and Harshman., 1990] Scott Deerwester, Susan T Dumais, G. W. F. T. K. L. and Harshman., R. (1990). Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407.

[Sho Yokoi and Inui., 2020] Sho Yokoi, Ryo Takahashi, R. A. J. S. and Inui., K. (2020). Word rotator's distance. *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, page 2944–2960.

[Stephen E Robertson, 1995] Stephen E Robertson, Steve Walker, S. J. M. M. H.-B. M. G. e. a. (1995). Okapi at trec-3. *Nist Special Publication Sp*, page 109:109.

[Thibault Sellam, 2020] Thibault Sellam, Dipanjan Das, A. P. (2020). Bleurt: Learning robust metrics for text generation. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, page 7881–7892.

[Tianyi Zhang, 2020] Tianyi Zhang, Varsha Kishore, F. W. K. Q. Y. A. (2020). Evaluating text generation with bert. *International Conference on Learning Representations*.

[Xiaofei Sun and ChunFan., 2022] Xiaofei Sun, Yuxian Meng, X. A. N. F. W. T. Z. J. L. and ChunFan. (2022). Sentence similarity based on context. *https://arxiv.org*.

[Yuhua Li, 2004] Yuhua Li, Zuhair Bandar, D. M. J. O. (2004). A method for measuring sentence similarity and its application to conversational agents. *Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference, Miami Beach, Florida, USA*.

[Yuhua Li and Crockett, 2006] Yuhua Li, David McLean, Z. A. B. J. D. O. and Crockett, K. (2006). Sentence similarity based on semantic nets and corpus statistics. *IEEE transactions on knowledge and data engineering*, 18(8):1138–1150.