UoR at SemEval-2020 Task 4: Pre-trained Sentence Transformer Models for Commonsense Validation and Explanation

Thanet Markchom, Bhuvana Dhruva, Chandresh Pravin, Huizhi Liang
University of Reading
White Knights, Berkshire, RG6 6AH
United Kingdom
`t.markchom@pgr.reading.ac.uk, bhuvna_99@yahoo.co.uk,`
`c.pravin@pgr.reading.ac.uk, huizhi.liang@reading.ac.uk`

Abstract

SemEval Task 4 Commonsense Validation and Explanation Challenge is to validate whether a system can differentiate natural language statements that make sense from those that do not make sense. Two subtasks, A and B, are focused in this work, i.e., detecting against-common-sense statements and selecting explanations of why they are false from the given options. Intuitively, commonsense validation requires additional knowledge beyond the given statements. Therefore, we propose a system utilising pre-trained sentence transformer models based on BERT, RoBERTa and DistillBERT architectures to embed the statements before classification. According to the results, these embeddings can improve the performance of the typical MLP and LSTM classifiers as downstream models of both subtasks compared to regular tokenised statements. These embedded statements are shown to comprise additional information from external resources which help validate common sense in natural language.

1 Introduction

Against-common-sense statements are statements which are contradictory, either partly or entirely, to the fact or do not make sense in reality. Understanding these statements can be challenging due to the variety and complexity of natural languages, yet it can be highly beneficial for many purposes such as detecting false information or validating statements generated by computer program. To explore this approach, Task4: Commonsense Validation and Explanation Challenge (ComVE) is introduced as a challenge in SemEval-2020. The main goal of ComVE is to develop a system that can differentiate and explain counterfactual natural language statements. This task consists of three subtasks, Subtask A, Subtask B and Subtask C. In Subtask A, two statements are given, one is against common sense while the other is not. The objective of Subtask A is to identify the statement that is contradictory to common sense. For Subtask B and C, the objectives are to explain why a given statement is contradict to the fact by using different methods. In Subtask B, an explanation is selected from the given options, while, an entire explanation has to be generated from scratch in Subtask C. The total detail can be found in (Wang et al., 2019; Wang et al., 2020)

To decide whether a given statement makes sense, additional context may be required. In such case, a sentence embedding model becomes useful for including this additional information by semantically encoding statements before classification. In this work, we study on using pre-trained sentence transformer models (Reimers and Gurevych, 2019) which are state-of-the-art transformer models fine-tuned for sentence embedding to solve the problem in Subtask A and Subtask B. These models were modified based on three widely used transformer models with different configurations. In our approach, these pre-trained models are used to embed the statements so that the embedded statements can be classified more effectively to detect against-common-sense statements in Subtask A and find an explanation for against-common-sense statements in Subtask B.

2 Related Work

Language pre-training methods have been long utilised in NLP tasks and have, indeed, been proven to be effective for developing a successful model (Dai and Le, 2015; Devlin et al., 2018). Peters et al. (2018) proposes the ELMo (Embedding from Language Models) representation, a feature-based approach which makes use of pre-trained representations of the data as additional features in the network and promises improved performance over using a long short-term memory (LSTM) top layer (Young et al., 2018). A variation of the Hybrid Neural Network, as explored by He et al. (2019) is the Knowledge Enhanced Hybrid Neural Network (KEHNN). This model makes use of prior knowledge representations and multiple channels to improve matching of long text data-sets (Wu et al., 2018).

In recent time there have been notable advancements in commonsense validation and reasoning models, many of the new developments stemming from the BERT language representation model proposed by Devlin et al. (2018) such as the Robustly Optimised BERT (RoBERTa) pre-training approach developed by Liu et al. (2019) and DistilBERT, which used knowledge reduction, or distillation, techniques whilst retaining the models language understanding capabilities and decreasing its training and computation timings (Sanh et al., 2019). Another development on the BERT model is the Commonsense Auto-Generated Explanations (CAGE) model proposed by Rajani et al. (2019) which, in their study, shows an improved accuracy when classifying the CQA (Commonsense Question Answer) dataset when compared with other similar networks.

The SBERT model, a variation on the BERT model, which makes use of Siamese and triplet network architectures, has been found to be particularly effective with deriving sentence embedding whilst retaining the underlying semantics within the sentences (Reimers and Gurevych, 2019). The resultant embeddings can then be further applied to classification networks, or regression models equally. However, there has been no work where this pre-trained embedding model is applied in a commonsense validation and explanation task. Therefore, in this work, we propose to use sentence transformer models including SBERT and other embedding models based on RoBERTa and DistillBERT developed by Reimers and Gurevych (2019) to embed the statements and classify them in Subtask A and Subtask B.

3 System Overview

The proposed system consists of two parts: (1) The pre-processing stage, in which the original datasets are adjusted to fit our classification model, and NLP techniques, which are applied for cleaning and preparing the data before further analysis. (2) The classifier with pre-trained sentence transformer models. In this stage the pre-processed statements are embedded by using pre-trained sentence transformer models and then passed to a multilayer perceptron (MLP) model and long short-term memory (LSTM) model for classification. These two parts are detailed in the following subsection.

3.1 Pre-processing

Firstly, the input and output data are converted to a compatible format for the proposed system. In Subtask A, the input data consists of two statements and the output data is a set of labels identifying sentences which do and do not follow common-sense language. To simplify the problem, instead of comparing two statements at the same time, each statement is considered individually to decide if it agrees with common sense or not. Specifically, a pair of given statements is decoupled and each of them is assigned with either 0 if the statement is false or 1 if the statement is true. As a result, considering any pair of statements decoupled, the statement with a lower predicted value will be considered as an against-common-sense statement.

In Subtask B, for each false statement, three options (A, B and C) are given as possible reasons why the statement contradicts to common sense. The objective is to identify a correct statement from any incorrect statements, this can be can be interpreted as a mulitclass classification problem. We propose a model in which this problem is formulated as a binary classification task. To do so, considering each sample, a false statement is concatenated with each of the options creating three new concatenated statements. These statements are labeled with 1 if an option concatenated is correct, otherwise it is labeled with 0. For prediction, the option with the highest value is selected as an explanation for each false statement.

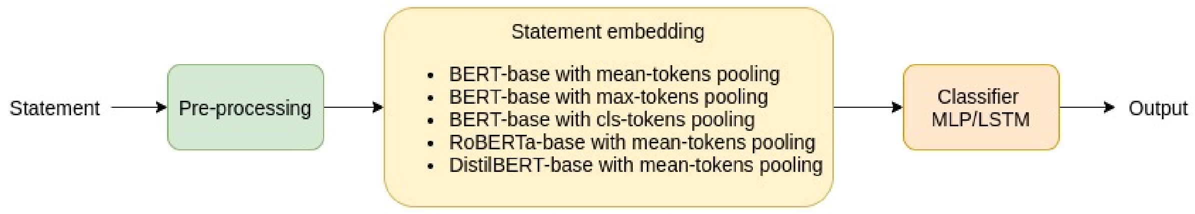
Statement
Pre-processing
Statement embedding
BERT-base with mean-tokens pooling
BERT-base with max-tokens pooling
BERT-base with cls-tokens pooling
RoBERTa-base with mean-tokens pooling
DistilBERT-base with mean-tokens pooling
Classifier MLP/LSTM
Output

Figure 1: The proposed framework

Moreover, there are a number of statements in which names of a person appear. These names are firstly removed and replaced with a word "person" to decrease any ambiguity during training. Additionally, numbers are also replaced by a corresponding word. All punctuation marks are also removed from the statements and all characters are finally converted to lower cases.

3.2 Classifier with Pre-trained Sentence Transformer Embedding Model

Validation of common sense within natural language statements require additional information to provide more context in reality. Thus, for the better commonsense comprehension, we propose to include auxiliary knowledge gained from other natural language resources. This knowledge is from pre-trained sentence transformer models capable of embedding the statements into meaningful vectors. These sentence transformer models were developed on different state-of-the-art transformer models, i.e., BERT, RoBERTa and DistilBERT and were trained on either only **Natural Language Inference (NLI)** dataset (Bowman et al., 2015) or both NLI and **Semantic Textual Similarity (STS)** (Cer et al., 2017) datasets combined. The generated embeddings are accordingly enriched with semantic knowledge from these resources and are suitable for the challenge goals.

Each sentence transformer model consists of word embeddings and a pooling model. First, one of the sentence transformer models is chosen as a word embedding model to generate embeddings of all words (tokens) in a statement including a special classification token ('cls'), inserted at the beginning of a statement by the transformer model as an aggregate of the entire statement for classification (Devlin et al., 2018). Then, these embeddings are passed to the pooling model to form a single vector as a statement embedding. There are three pooling methods that can be applied in this part: (1) mean-pooling using the mean of token embeddings, (2) max-pooling using the max-over-time of token embeddings and (3) cls-pooling using the embedding of 'cls' token as a statement representation. After pooling, a 768-dimensional vector can be obtained as an embedding of a statement given to the sentence transformer model.

To build the classifiers, after the statements in both subtasks are pre-processed, they will be embedded with the pre-trained sentence transformer models. All sentence transformer models with the configurations adopted in this work can be seen in Table 1-4 where the first column shows the transformer model used as a base model and the second column shows the dataset used for pre-training. On top of this embedding procedure, an MLP model and LSTM model are used as a classifier to validate common sense according to the embedding of statements. The whole proposed framework is illustrated in Figure 1. Moreover, since multiple classifiers using different sentence transformer model are built, voting-based ensemble model is adopted to make use of all implemented classifiers. It selects the final prediction based on majority voting of all predictions.

4 Experimental Setup

The models were trained with the training and development set provided in both subtasks. However, in Subtask B, after conversion, one original sample can be converted to three new samples, one sample labeled with 1 and two samples labeled with 0, causing an imbalance in the dataset. Therefore, the data from Subtask C was used to compensate the positive samples. In Subtask C, for each false statement, there are three examples that can be used to explain how the given statement is false. Thus, one of these example distinct from the correct option in Subtask B was randomly selected and added to Subtask B dataset.

As for the pre-processing step, we used existing libraries, i.e., spaCy Named Entity Recognition[1] to detect and replace person's names, num2words[2] to convert numbers to words and the punctuation set in Python string library to remove punctuation marks. We selected two architectures, MLP and LSTM models, to build classifiers for both subtasks. The MLP model consists of four hidden layers with 512, 256, 128, 64 number of neurons and a rectified linear unit (ReLU) as an activation function. Dropout layers were included after the input layer with dropout rate 0.1 and between each hidden layer with dropout rate 0.5 to avoid overfitting. For LSTM model, we used an LSTM layer with 64 dimensions of an output. Then, it is followed by a fully-connected layer consists of 64 neurons. The proposed models were trained with Adam optimiser for 20 epochs. A binary cross-entropy and accuracy were used as a loss function and metric respectively. All models were evaluated on both trial and test set by accuracy which will be shown in the next section.

4.1 Baseline models

- **MLP without statement embedding**: To compare the performance of the statement embedding, the same MLP model used in the proposed system are trained on regular tokenised statements without embedding procedure. The number of layers, the size of each layer and all hyper-parameters are set identically as in the proposed approach.

- **LSTM without statement embedding**: As the MLP baseline model, the LSTM model were also trained on regular tokenised statements for comparison. The structure model and parameter settings are the same as used in the proposed framework.

5 Result

The MLP and LSTM classifier results of Subtask A are shown in Table 1 and Table 2 respectively. For the MLP classifier, the RoBERTa-base model with mean-tokens pooling pre-trained on NLI data can achieve the highest accuracy (80.3%) for the trial set, while the BERT-base model with mean-tokens pooling pre-trained on NLI data can achieve the highest accuracy (84.9%) for the test set. The same RoBERTa-base model can also achieve the highest accuracy for both trial (63.2%) and test set (67.3%) with the LSTM classifier. However, comparing these two classifiers, the MLP classifier can perform better than the LSTM classifier in all cases. Lastly, the ensemble model of an MLP approach can compute the voted result with 81.1% accuracy for trial set and 86.8% accuracy for test set in post-evaluation submission which are higher than relying on only one classifier.

Model	**Pretrained dataset**	**Accuracy (%)**	
		Trial set	**Test set**
MLP	-	50.4	50.8
MLP + BERT-base with mean-tokens pooling	NLI	78.7	**84.9**
MLP + BERT-base with max-tokens pooling	NLI	77.9	80.7
MLP + BERT-base with cls-tokens pooling	NLI	78.2	84.4
MLP + RoBERTa-base with mean-tokens pooling	NLI	**80.3**	84.1
MLP + DistilBERT-base with mean-tokens pooling	NLI	75.4	82.9
MLP + BERT-base with mean-tokens pooling	NLI + STS	78.0	83.9
MLP + RoBERTa-base with mean-tokens pooling	NLI + STS	78.2	81.5
MLP + DistilBERT-base with mean-tokens pooling	NLI + STS	74.9	80.8
Ensemble model	-	**81.1**	**86.8**

Table 1: MLP classifier results of Subtask A.

[1]version 0.5.10 https://pypi.org/project/num2words/

[2]version 2.2.5 https://spacy.io/models/en

Model	Pretrained dataset	Accuracy (%)	
		Trial set	Test set
LSTM	-	50.7	53.4
LSTM + BERT-base with mean-tokens pooling	NLI	61.1	65.5
LSTM + BERT-base with max-tokens pooling	NLI	61.7	64.9
LSTM + BERT-base with cls-tokens pooling	NLI	59.2	63.5
LSTM + RoBERTa-base with mean-tokens pooling	NLI	**63.2**	**67.3**
LSTM + DistilBERT-base with mean-tokens pooling	NLI	57.3	61.4
LSTM + BERT-base with mean-tokens pooling	NLI + STS	59.7	59.9
LSTM + RoBERTa-base with mean-tokens pooling	NLI + STS	57.2	60.6
LSTM + DistilBERT-base with mean-tokens pooling	NLI + STS	57.5	62.7
Ensemble model	-	**64.1**	**70.0**

Table 2: LSTM classifier results of Subtask A.

Model	Pretrained dataset	Accuracy (%)	
		Trial set	Test set
MLP	-	34.0	32.0
MLP + BERT-base with mean-tokens pooling	NLI	**66.8**	71.1
MLP + BERT-base with max-tokens pooling	NLI	62.3	67.7
MLP + BERT-base with cls-tokens pooling	NLI	66.0	69.4
MLP + RoBERTa-base with mean-tokens pooling	NLI	66.5	69.2
MLP + DistilBERT-base with mean-tokens pooling	NLI	63.3	67.5
MLP + BERT-base with mean-tokens pooling	NLI + STS	64.4	**71.2**
MLP + RoBERTa-base with mean-tokens pooling	NLI + STS	66.4	69.7
MLP + DistilBERT-base with mean-tokens pooling	NLI + STS	63.7	68.8
Ensemble model	-	**67.5**	**72.3**

Table 3: MLP classifier results of Subtask B.

For Subtask B, as shown in Table 3, using embedded statements can increase the accuracy of the MLP classifier approximately twice. The BERT-base model with mean-tokens pooling pre-trained on NLI and STS data can perform best on the test data with 71.2% accuracy. Meanwhile, the same BERT-base model but pre-trained on only NLI data can achieve the best accuracy for the trial set with 66.8% accuracy. Considering the LSTM resutls in Table 4, the LSTM model coupled with BERT-base with mean-tokens pooling can achieve the highest accuracy for both trial set (56.8%) and test set (60.6%). However, the results show no significant improvement when the embedded input statements are used unlike Subtask A where the LSTM classifier can perform better with embedded input statements. Finally, combining all predictions from the MLP classifiers, the ensemble model can yield 67.5% accuracy for the trial set and 72.3% accuracy for post-evaluation submission.

For both subtasks, the results show that using the pre-trained sentence transformer models with the MLP classifier can largely increase the performance of commonsense validation and explanation.On the other hand, the LSTM classifier can perform better with the embedded statements only in Subtask A whereas there is no outstanding improvement in performance when it was applied in Subtask B. Considering the sentence transformer models, the BERT-base and RoBERTa-base models show to perform better than the DistilBERT-base models, which consist of fewer parameters, in most cases. For different pooling methods on the same BERT-base model, the results obtained from the mean-tokens and cls-tokens methods with the MLP classifier are not significantly different. Meanwhile, using max-tokens pooling causes a slight accuracy decline compared to the others which agrees with the result in the original paper (Reimers and Gurevych, 2019). Contrary, with the LSTM classifier, the cls-tokens pooling performs worst in both subtasks while mean-tokens and max-tokens pooling methods can perform almost equally better.

Model	**Pretrained dataset**	**Accuracy (%)**	
		Trial set	**Test set**
LSTM	-	46.0	50.0
LSTM + BERT-base with mean-tokens pooling	NLI	**56.8**	**60.6**
LSTM + BERT-base with max-tokens pooling	NLI	55.3	59.1
LSTM + BERT-base with cls-tokens pooling	NLI	52.4	56.7
LSTM + RoBERTa-base with mean-tokens pooling	NLI	55.9	55.8
LSTM + DistilBERT-base with mean-tokens pooling	NLI	36.3	40.0
LSTM + BERT-base with mean-tokens pooling	NLI + STS	35.1	37.6
LSTM + RoBERTa-base with mean-tokens pooling	NLI + STS	49.1	49.1
LSTM + DistilBERT-base with mean-tokens pooling	NLI + STS	41.3	45.0
Ensemble model	-	**55.9**	**58.8**

Table 4: LSTM classifier results of Subtask B.

According to our experiment applying mean-tokens pooling, all token embeddings in a statement are summarised together as well as using the 'cls' token embedding which is treated as a whole statement representation. On the other hand, selecting the maximum value and neglecting the other values may lead to essential information loss causing the lower meaningful embedding performance.

6 Conclusion

Commonsense validation and explanation in natural language statements requires additional knowledge which is, occasionally, inadequate when only a single statement is considered independently. To satisfy this condition, this work proposes a system using the pre-trained sentence transformer models to embed the statements before classification. With semantic knowledge these model gained from both NLI and STS datasets, the statements can be embedded into vectors comprise the meaning and additional commonsense comprehension based on these natural language resources. In our experiment, these embedding models were proved to be useful for detecting and explaining counterfactual statements. The results show classification models with embedding models have better accuracy than those without embedding models. This suggests that pre-trained sentence transformer models can provide additional prior knowledge for commonsense validation and explanation.

References

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada, August. Association for Computational Linguistics.

Andrew M Dai and Quoc V Le. 2015. Semi-supervised sequence learning. In *Advances in neural information processing systems*, pages 3079–3087.

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

Pengcheng He, Xiaodong Liu, Weizhu Chen, and Jianfeng Gao. 2019. A hybrid neural network model for commonsense reasoning. *Proceedings of the First Workshop on Commonsense Inference in Natural Language Processing*.

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.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1*.

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain yourself! leveraging language models for commonsense reasoning. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

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.

Cunxiang Wang, Shuailong Liang, Yue Zhang, Xiaonan Li, and Tian Gao. 2019. Does it make sense? and why? a pilot study for sense making and explanation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4020–4026, Florence, Italy, July. Association for Computational Linguistics.

Cunxiang Wang, Shuailong Liang, Yili Jin, Yilong Wang, Xiaodan Zhu, and Yue Zhang. 2020. SemEval-2020 task 4: Commonsense validation and explanation. In *Proceedings of The 14th International Workshop on Semantic Evaluation*. Association for Computational Linguistics.

Yu Wu, Wei Wu, Can Xu, and Zhoujun Li. 2018. Knowledge enhanced hybrid neural network for text matching. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

T. Young, D. Hazarika, S. Poria, and E. Cambria. 2018. Recent trends in deep learning based natural language processing [review article]. *IEEE Computational Intelligence Magazine*, 13(3):55–75.