SSN-NLP at SemEval-2020 Task 4: Text Classification and Generation on Common Sense Context using Neural Networks

Rishi Vardhan K, Kayalvizhi S, Thenmozhi D, Raghav R and Kshitij Sharma

Department of Computer Science and Engineering, SSN College of Engineering, Chennai.

rishivardhan18126@cse.ssn.edu.in {kayalvizhis, theni_d}@ssn.edu.in {raghav18119, kshitij18080}@cse.ssn.edu.in

Abstract

Common sense validation deals with testing whether a system can differentiate natural language statements that make sense from those that do not make sense. This paper describes the SSN-NLP's approach to solve this challenge. For common sense validation with multi choice, we propose a stacking based approach to classify sentences that are more favourable in terms of common sense to the particular statement. We have used majority voting classifier methodology amongst three models such as Bidirectional Encoder Representations from Transformers (BERT), Micro Text Classification (Micro TC) and XLNet. For sentence generation, we used Neural Machine Translation (NMT) model to generate explanatory sentences.

1 Introduction

In an increasingly digital and mobile world, it would be difficult to build adaptable and unsupervised NLP systems without common sense. The power of common sense systems is their sense of adaptability to varied topics. This is because common sense systems provide their response based on contextual based understanding of the problem. Its therefore essential to bridge the gap between human language processing and machine natural language processing by improving common sense validation and explanation in the latter. By the ComVE challenge, the task is to directly test whether a system can differentiate natural language statements that make sense from those that do not make sense.

We primarily had focused on sub task B and sub task C of the challenge where Task B is to find the key reason from three options why a given statement does not make sense and Task C asks machine to generate the reasons and we use BLEU metric proposed by Papineni et al. (2002) to evaluate them. For Task B we had used a mix of NLP models among BERT by Devlin et al. (2018), Micro TC by Tellez et al. (2018) and XLNet from Yang et al. (2019) to run a multi classification problem and finally post process the results based on majority voting approach. For Task C we had used NMT (Luong et al., 2017) model to generate sentences that provide explanations as to why the particular sentence is against common sense.

2 Dataset

Each sub task is associated with a separate data set for its processing. In sub task B the input data is separated as a particular statement with three other sentence options amongst which one of them is the most favourable reason to why the statement is not common sense valid. In sub task C the input data is separated as a particular statement with three supporting statements to why the particular statement is against common sense. The train, trial, dev and test set instances for both tasks are of sizes 10000, 2021, 998 and 1000 respectively. The data collection is described by (Wang et al., 2020). The training and testing data under trial phase were collectively used to train the model.

3 Methodology

3.1 Sub Task B: Explanation (Multi-Choice)

The aim of sub task B is to predict which sentence from the given set is the most applicable reason to why a particular statement is against common sense. An example from the data set for task B is as follows

Statement S: He put an elephant into the fridge
A: An elephant is much bigger than a fridge.
B: Elephants are usually white while fridges are usually white.
C: An elephant cannot eat a fridge.

The expected value of the above example is option A. Thereby the task is essentially a multi-class classification task where the option sentences A, B and C are the target variables. The models were trained to predict which sentence out of the three is the most favourable in terms of common sense to the particular statement S. The training input data was processed to split each sentence along with its three options into three records of data such that each option sentences were concatenated along with their respective statements with **¡EOL¿** tag between and a indication feature as class 1 or class 0 to denote which sentence amongst the three is the expected answer. Thereby the data was processed to perform a binary text classification task with class 1 indicating that the record is in fact the reason to why the sentence is against common sense and class 0 indicating that the record doesn't hold the applicable reason to why the sentence is against common sense. After processing of input data for task B, three different architecture models were trained explicitly for the same task.

3.1.1 Model details

The models implemented were BERT, Micro TC and XLNet. The final outputs of the three models were composed into a singular result by majority voting methodology.

Micro TC

Micro TC is a minimalist and multi-propose text-classifier able to tackle tasks independently of domain and language. For any given text classification task, micro TC will try to find a suitable text model from a set of possible models defined in the configuration space, provided a corpus of defined text data. The model was initially fed with train data (corpus) to find the suitable hparams or suitable text model. Along with the derived hparams and the initial train set, the model was trained with use of Linear SVC.

BERT

BERT makes use of Transformers, an attention mechanism that learns contextual relations between words in a text. In its general form, Transformer includes two separate mechanisms — an encoder that reads the text input and a decoder that produces a prediction for the task. Since BERT's goal is to generate a language model, only the encoder mechanism is necessary.

As opposed to directional models, which read the text input sequentially (left-to-right or right-to-left), the Transformer encoder reads the entire sequence of words at once. This characteristic allows the model to learn the context of a word based on all of its surroundings (left and right of the word). BERT's prediction goal is mainly achieved by two training strategies as described below.

- **Masked LM (MLM)**: Before feeding word sequences into BERT, 15% of the words in each sequence are replaced with a [MASK] token. The model then attempts to predict the original value of the masked words, based on the context provided by the other, non-masked, words in the sequence.
- **Next Sentence Prediction (NSP)**: The model receives pairs of sentences as input and learns to predict if the second sentence in the pair is the subsequent sentence in the original document. During training, 50% of the inputs are a pair in which the second sentence is the subsequent sentence in the original document, while in the other 50% a random sentence from the corpus is chosen as the second sentence.

When training the BERT model, Masked LM and Next Sentence Prediction are trained together, with the goal of minimizing the combined loss function of the two strategies. For the task of classification using

BERT, the Next Sentence Prediction (NSP) strategy training of BERT is optimised to add a classification layer on top.

During training the available pre-trained model BERT-Base Un-Cased was used. The BERT-Base Uncased comprises of 12 Transformer blocks, 12 self-attention heads and 768 hidden dimension with a total parameters of 110M. The BERT-Large model was not used although it fared better than BERT-Base due to its computational intensive need for training the model.

XL-Net

XLNet is a generalized AR pretraining method that uses a permutation language modeling objective to combine the advantages of AR (autoregression) and AE (autoencoding) methods. XLNet uses a subset of the bidirectional context each time it predicts a word. The neural architecture of XLNet is developed to work seamlessly with the AR objective, including integrating Transformer-XL and the careful design of the two-stream attention mechanism. XLNet manages to overcome the deficiencies of BERT whilst requiring more compute power and memory (GPU/TPU memory) in comparison to BERT.

For XLNet, the pre-trained model XLNet Base Cased model was used. The XLNet Base model comprises of 12 Transformer blocks, 12 self-attention heads and 768 hidden dimensions.

3.1.2 Implementation

The Neural Network models, BERT and NMT were implemented using Hugging Phase Library of Transformers (Wolf et al., 2019). The models were trained with a batch size of 32 for 2 epochs. The drop out probability was set to 1 for all layers and learning rate of 2e-5 was set. Since Micro TC is based on a general machine learning model, it was implemented using sckit (Pedregosa et al., 2011) and microtc packages. The training was performed on a Nvidia Tesla v100 smx2 GPU.

3.1.3 Post-Processing

The individual outputs obtained from BERT, Micro TC and XLNet were subjected to post-processing where the output probability scores of three sentences (records) for a particular statement are processed into a single record by means of high probability scores for class 1. Further the processed results from the three stacked models were composed into final form of output by performing majority voting approach. The result thereby obtained after two stages of post-processing are concluded to be the final output values.

3.2 Task C: Explanation (Generation)

The main aim of Task C is generate the reason why a particular statement is against common sense. An example from the data set for task C is as follows.

Statement: He put an elephant into the fridge.
Referential Reasons:
1. An elephant is much bigger than a fridge.
2. A fridge is much smaller than an elephant.
3. Most of the fridges aren’t large enough to contain an elephant.

The input data is processed into train, development, vocabulary and test files where the train file is a composition of test data and train data from trial stage and the vocabulary file is based on the input data fed.

3.2.1 Model details

For task C, a sequence2sequence or seq2seq architecture model is employed, i.e **Neural Machine Translation (NMT)** . The architecture of NMT is based on a single neural network comprised of two RNNs.

- **Encoder RNN**: It extracts all of the pertinent information from the source sentence to produce an encoding or a *thought vector*
- **Decoder RNN**: It generates the target sentence conditioned with the encoding created by the encoder

The decoder is trained with a method called "teacher forcing". The target sequence is the input sequence offset by one. Thereby the decoder is effectively trying to generate contextual meaning based text.

3.2.2 Implementation

The Neural Network model was implemented using Pytorch. The model was trained as a 2 layer LSTM seq2seq model with 128-dim hidden units and embeddings for 150 epochs. The train batch size was set to 128 with gradient norm set to maximum of 5. We used a dropout value of 0.2. The training was performed on a Nvidia 1070 GPU.

4 Results

4.1 Practice Phase

Task B

In the practice phase the individual models achieved accuracy as listed in Table 1.

Table 1: Practice Results

Models	**Accuracy**
XLNet	83.83
BERT	82.63
MicroTC	84.06
Combined Accuracy	80.10

Task C

During practice phase, the NMT model resulted in a BLEU score of 39.2.

4.2 Evaluation Phase

Task B

The Evaluation phase results of Task B had an accuracy of **68.3** which ranked our team at 21st position out of 27 participating teams. In the Post-Evaluation phase the individual models performance are as observed in Table 2.

Task C

The evaluation results of Task C reported a BLEU score of 2.2 and a Human Evaluation Score of 0.59 ranking us at 16th position.

4.3 Post Evaluation Phase

The results obtained from post evaluation phase for the individual models of Task B are as listed below.

Table 2: Post Evaluation Results

Models	**Accuracy**
XLNet	83.93
BERT	79.50
MicroTC	50.50
Combined Accuracy	68.3

From table 2 we observe that Micro TC performed significantly poor on the test set and therefore contributed to the low combined accuracy for task B.

5 Conclusion

We have implemented both traditional machine learning and deep learning approach for the task of classifying and generating sentences based on the context of common sense. For the task of classifying sentences or sub task B, BERT, XLNet and Micro TC classifiers are implemented in binary classification structure and majority voting approach is implemented further to obtain final results. The metric used to judge performance is accuracy. For the task of generating explanatory sentences or sub task C, NMT model is implemented. The NMT model is measured for performance using BLEU score. The results from sub task B for individual models as observed from Table 2 signify that deep learning models BERT and XLNet performed better than Micro TC. In sub task C, NMT performed poorly on the test set due to the less amount of time or iterations it was trained. The performance can be improved further by incorporating external data-sets and increasing the number of training steps while training the model.

References

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*.

Minh-Thang Luong, Eugene Brevdo, and Rui Zhao. 2017. Neural machine translation (seq2seq) tutorial. *https://github.com/tensorflow/nmt*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Eric S. Tellez, Daniela Moctezuma, Sabino Miranda-Jiménez, and Mario Graff. 2018. An automated text categorization framework based on hyperparameter optimization. *Knowledge-Based Systems*, 149:110–123.

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.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. cite arxiv:1906.08237Comment: Pretrained models and code are available at https://github.com/zihangdai/xlnet.