# Report

## Assignment 3: Project (graded part)

**Name and fruit 1: Sijie Ju Orange**

## 1. Task and data

The task I did in assignment 3 is SemEval 2020 Task 4 (Wang et. al., 2020). This task is to test whether a model can distinguish statements that make sense from those that do not. This task contains three subtasks: validation, commonsense explanation (multiple choice), and commonsense explanation (generation). I did the subtask A and subtask C.

Subtask A is a classification task. The model needs to judge two natural language statements with similar wordings which one makes sense and which one does not make sense.

Statement 1: *He put a turkey into the fridge. (correct)*
Statement 2*: He put an elephant into the fridge.*

Subtask C is a generation task. The model needs to explain why the given statement doesn't make sense.

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 data of this task was provided by the organizers, which is available on GitHub. For both two subtasks, the dataset contains 10,000 rows of training data, 997 rows of validation data and 1,000 rows of test data. For subtask A, each row contains two sentences and one label that denotes the statement against commonsense (0 for the first sentence and 1 for the second sentence). For subtask C, each row contains a statement against commonsense and three referential reasons to explain it.

## 2. Method

In subtask A, I used BERT as it is an LLM pretrained on a large corpus of English data. It can be finetuned to complete various downstream tasks, including text classification. I worked it out in two ways: finetuning the BERT model with an adapter and finetuning the whole BERT model with no adapter. For adapter finetuning, I used the Adapter trainer provided by Hugging Face. For model finetuning, I wrote the training structure by myself because when I used the Trainer to train the model, the loss of model didn't reduce no matter how I modified the hyperparameters. In subtask C, I used GPT2. As it uses unidirectional Transformer decoder, it works better in text generation task compared to BERT.

For the evaluation of model in subtask A, I calculated the accuracy of its prediction and also used the loss change as reference. In subtask C, since the evaluation of generated text is complicated, I used BLEURT (Sellam, Das & Parikh, 2020) to evaluate the model. It is an evaluation metric for natural language generation which is easily evoked with a Python API. I have also attempted to use the evaluation tool provided by the organizers, which is a scorer based on BLEU, however it couldn't run with my data.

## 3. Results

In subtask A, the model performed best when I finetuned the adapter with learning rate of 4e-4, batch size of 16 and epoch of 10. The test accuracy was 86%.  Out of 1000 pairs of test sentences, the model misinterpreted 140 pairs.

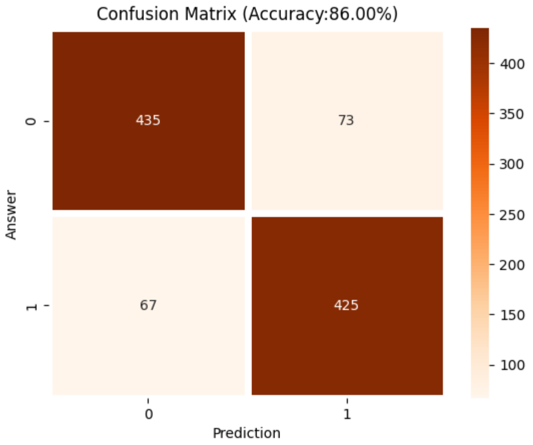


Figure 1 Confusion matrix of the best model with adapter (lr = 4e-4, batch size = 16, epoch =10)

The performance of overall finetuned model without an adapter is a little inferior to the one with an adapter. The best implementation is the model trained with learning rate of 1e-5, batch size of 16 and epoch of 10, which achieved the accuracy of 85.3%.

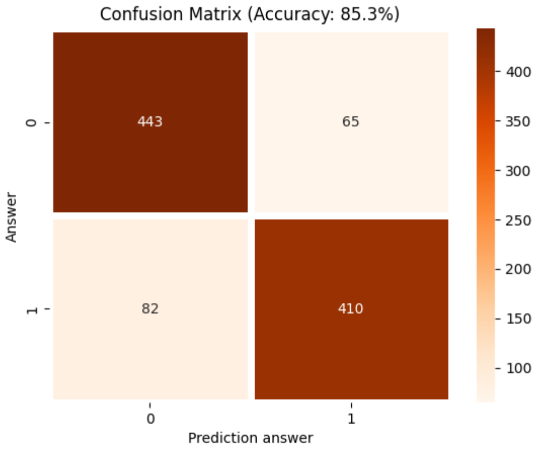
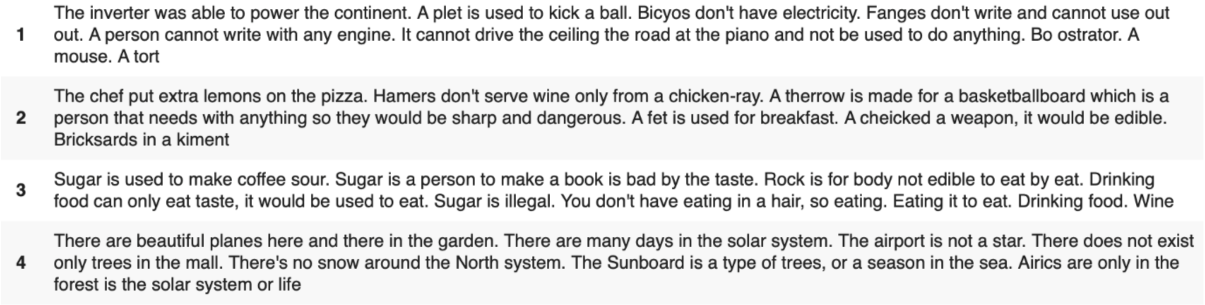


Figure 2 Confusion matrix of the best model without adapter (lr = 4e-4, batch size = 16, epoch =10)

Comparing to the results of other participants of this task, this model isn't good. The best achieved the accuracy of 97%. I also found a project (Lee, 2021) submitted on GitHub. The author finetuned the model Roberta-Large with learning rate of 1e-5, batch size of 128 and epoch of 15, and finally achieved the accuracy of 94.3%.

In the subtask C, the result is quite awful. I trained the model with learning rate of 2e-4, epoch of 5 and batch size of 8. The score given by BLEURT is very low. As the generation text would change with new trial, I asked the model to generate the explanation three times. Within three generations, its highest score was -0.286. The generated texts don't make any sense as shown in the table.



| | |
|---|---|
| 1 | The inverter was able to power the continent. A plet is used to kick a ball. Bicyos don't have electricity. Fanges don't write and cannot use out out. A person cannot write with any engine. It cannot drive the ceiling the road at the piano and not be used to do anything. Bo ostrator. A mouse. A tort |
| 2 | The chef put extra lemons on the pizza. Hamers don't serve wine only from a chicken-ray. A therrow is made for a basketballboard which is a person that needs with anything so they would be sharp and dangerous. A fet is used for breakfast. A cheicked a weapon, it would be edible. Bricksards in a kiment |
| 3 | Sugar is used to make coffee sour. Sugar is a person to make a book is bad by the taste. Rock is for body not edible to eat by eat. Drinking food can only eat taste, it would be used to eat. Sugar is illegal. You don't have eating in a hair, so eating. Eating it to eat. Drinking food. Wine |
| 4 | There are beautiful planes here and there in the garden. There are many days in the solar system. The airport is not a star. There does not exist only trees in the mall. There's no snow around the North system. The Sunboard is a type of trees, or a season in the sea. Airics are only in the forest is the solar system or life |

Table 1 Generated text by model

The first sentence is the given statement which is against commonsense. The followings are the texts generated by the model. It is evident that there is no relation between the statement and the explanation, either not to mention the expected causal logic.

The best generated texts out of these 1000 explanations scored 0.185.

**Statement:** *She put the giraffe in the freezer.*
**Explanations:** *A giraffe is too big to big to fit into a pen. A giraffe is not a place to fit on a fridge. Giraffe are much smaller than a closet. Giraffe are located in a ovens and very large. Giraffe. Giraffe. A giraffe for put in ocean in kitchen.*

We can see that although these sentences are literally uninterpretable and have grammatical errors, at least all the generated explanations are centered with the keyword *giraffe*.

## 4. Analysis

During the finetuning process in subtask A, I encountered several problems. The loss wouldn't decrease in some cases.

1. Batch size is too small. Initially I set the batch size as 8. No matter how I adjusted other hypermeters, the model didn't learn anything. However, when I changed the batch size to 16, it started to be trained in some situations.

2. The learning rate isn't fit for the model. The best learning rate for adapter finetuning and model finetuning is also different. The adapter doesn't learn when the learning rate is small. On the contrary, finetuning the whole model requires a small learning rate.

| Models | Learning rate | Loss | Accuracy |
|---|---|---|---|
| Model with adapter | 4e-4 | Reduced | 86% (best) |
| Model with adapter | 1e-4 | Reduced | 80.7% |
| Model with adapter | 1e-5 | Didn't reduce | / |
| Model without adapter | 1e-5 | Reduced | 84.7%(best) |
| Model without adapter | 5e-5 | Reduced | 80.2% |
| Model without adapter | 1e-4 | Didn't reduce | / |

Table 2 Subtask A performance of model with different settings

As the table shows, when finetuning the adapter, lower learning rate led to poor performance. Accuracy reduced to 80.7% when learning rate fell to 1e-4. When the learning rate is 1e-5, the loss of model didn't decrease and remained around 69%.

On the contrary, when finetuning the whole model, lower learning rate led to better performance. The model performed best when the learning rate equals 1e-5, with the accuracy of 84.8%. When the learning rate is adjusted to 1e-4 or more, the loss didn't decrease.

Although the model with a finetuned adapter works better than that with all layers finetuned, the difference is rather unobvious.

Among all the errors, some mistakes perhaps make sense although most of them are ridiculous. Some statements are usually true, but there exist cases that they would be against the reality. Some statements are both true in some cases. Here shows two examples.

(1) a. Orange juice is made of oranges.  b. Orange juice is made of apples.
(2) a. Pens are for writing  b. Pens are for painting

In the first pair of sentences, the model labels the first sentence as nonsense. The oranges juice wouldn't be made of apples, but sometimes orange juice could also not be made of oranges, but of artificial flavors and colors. In the second pair of sentences, from the human perspective, two sentences can both be correct. So it would be difficult for the model to tell which one is against commonsense.

For the model in the subtask C, the training of text generation requires longer time and RAM resource. At first, I used the same epoch and batch size as in the subtask A. However, the code stopped running as colab ran out of RAM. Later, I reduced the epoch and batch size. The model started to be trained but it cost 4 hours to train the model. Thus, I used the accelerator provided by Hugging face to optimize the training process. The time required for training reduced to 20 minutes.

The quality of training data is quite important for model to learn how to generate natural language. Initially I didn't do any modification to the original data. The performance of the model is even worse. Many of the generated sentences are uncompleted. Some were just phrases, and some were not divided by any delimiter. Thus, I checked and modified the data. I added the period at the end for those sentences without it and also capitalized the beginning letters of sentences that wasn't capitalized before. Since then, the model performed better and makes less incomplete sentences.

Although subtask C is essentially a text generation task, it's high demanding for the models to generate appropriate answers for the nonsenses. The model needs not only to parse the given statement, but also to explain the problems. The causal logic behind the explanation and statements requires more training to learn. Here shows an example and its expected answer.

**Statement**: He loves to stroll at the park with his bed.
**Generated text**: The gym is too heavy to go in the zoo. Shoes are so too difficult to be carried in the kitchen. You cannot transport a garage on a supermarket.
**Answer**: A bed is too heavy to carry with when strolling at a park. The park does not have beds. A bed would be really heavy and awkward to carry through a park.

We can see that the format of the generated sentences looks like explanations, but the content are just combinations of unrelated words. They go against the semantic rules. What's more, the generated texts don't have any relation with the statement.

I think to train a model for subtask C, more rounds of training are necessary. The model I trained only learnt how to generate text of similar style with the reference. Perhaps the model firstly needs to learn whether the sentence is against commonsense, then starts to learn how to generate the explanations. The dataset should also be adjusted and improved with the training.

## 6. Conclusion

In this project, I trained BERT and GPT2 to complete the subtask A and C of SemEval 2020 Task 4. The best implementation in subtask A achieved the accuracy of 86% in distinguishing the sentences against commonsense from sentences that make sentence. The model trained for subtask C managed to generate complete sentences in most cases, but there exist various grammatical and semantic problems. It failed to give reasonable explanation for statements are against commonsense.

## 6. References

Sellam, T., Das, D., & Parikh, A. P. (2020). BLEURT: Learning robust metrics for text generation. *arXiv preprint arXiv:2004.04696.*

Lee,. L. T. (2021). MSc Project: Commonsense Validation and Explanation in Natural Language Processing. https://github.com/LetianLee/MSc-Project-SemEval2020-Task4

Wang, C. X., Liang, S.L., Jin, Y. L., Wang, Y. L., Zhu, X. D., Zhang.Y., (2020). SemEval-2020 Task 4: Commonsense Validation and Explanation. *Proceedings of The 14th International Workshop on Semantic Evaluation.* Association for Computational Linguistics. https://github.com/wangcunxiang/SemEval2020-Task4-Commonsense-Validation-and-Explanation

## 7. Resources used

SemEval 2020 Task 4 Dataset https://github.com/wangcunxiang/SemEval2020-Task4-Commonsense-Validation-and-Explanation
BLEURT https://github.com/google-research/bleurt/tree/master
BERT finetuning tutorial
https://colab.research.google.com/drive/1pTuQhug6Dhl9XalKB0zUGf4FIdYFlpcX
https://colab.research.google.com/drive/1-Va0EW0NfRNvrRGL1e5HzkCAasCUOjkL?usp=drive_link#scrollTo=ZtGHC_ZRzckM
GPT finetuning tutorial
https://colab.research.google.com/github/philschmid/fine-tune-GPT-2/blob/master/Fine_tune_a_non_English_GPT_2_Model_with_Huggingface.ipynb
https://www.youtube.com/watch?v=98jROp7Ij9A

In this project, Chat GPT was used to explain the tutorial codes, but it wasn't used to generate codes or the content of report.