Byun at SemEval-2024 Task 6: Text Classification on Hallucinating Text with Simple Data Augmentation

Cheolyeon Byun

byuncheolyeon@akane.waseda.jp

Abstract

This paper aims to classify sentences to see if it is hallucinating, meaning the generative language model has output text that has very little to do with the user's input, or not. This classification task is part of the Semeval 2024's task on Hallucinations and Related Observable Overgeneration Mistakes, AKA SHROOM, which aims to improve awkward-sounding texts generated by AI. This paper will first go over the first attempt at creating predictions, then show the actual scores achieved after submitting the first attempt results to Semeval, then finally go over potential improvements to be made.

1 Introduction

How AI and Large Language Models are able to understand and generate human language is elusive, to say the least. The underlying ingenuity behind the architecture of such models, involves technology and methods that are considered a black box like neural networks, named after the fact that the inner workings are impossible to grasp and properly digest for even experts (Castelvecchi, 2016). But language models are far from perfect, to the point where sometimes, text generated by complex natural language generation models are considered to be hallucinating. The term hallucination here refers to text that has been generated or processed to solve tasks like machine translation or Natural Language Generation, that are easily subject to the issue of being grammatically correct but being untethered from the user's input or the source material (Lee et al., 2019). This paper attempts to classify these hallucinating texts using different models and methods, in order to see which can get the best results in terms of accuracy.

2 Task Description

Semeval's task 6, hereinafter denoted as SHROOM, asks participants to successfully classify hallucination texts from non-hallucinating text, where each data has been annotated by 5 different annotators, where a majority vote is done to categorize each data point. Going over the JSON input data presented in Figure 1, The "hyp" row refers to the text that has been generated/processed by a model, so the output. The model here could refer to something like BERT, and can be seen as the "model" value. "src" refers to the user input or source material the model is working with in order to produce the output, and the "tgt" is the expected result that the model should be aiming for. In Figure 1, the model's "task" is "DM", or Definition Modeling. The model is expected to provide the definition for the word asked on the input. In this case input asks for the meaning of surmounting. The output of the model is "hyp", and the correct answer is "tgt". This output is put on a majority vote by 5 people, and finally the data in Figure 1 was labeled as hallucinating, where the probability of 0.6 because 3 out of 5 people voted in favor of hallucinating.

3 System Overview

3.1 Data Pipeline

The structure of the data pipeline is as follows. Language models such as BERT or RoBERTa were used to pre-process and tokenize text, which were then turned into high dimensional vectors by the word embedding layer that is able to capture rich contexts (Vaswani et al., 2017). Hyperparameters include, binary cross entropy as the loss function as it is a binary classification task, epochs were set as 20. The Adam optimizer was utilized for training the model, and learning rates were set as 0.0005. The Adam optimizer was used because it can lead to better results than stochastic gradient descent depending on the task thanks to its dynamic adjusting of learning rates,(Zhang, 2018) and tinkering around with SGD and Adam personally has led to the conclusion that Adam is slightly better in terms of accuracy.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 270–273
June 20-21, 2024 ©2024 Association for Computational Linguistics

```
[
  {
    "hyp": "A sloping top .",
    "ref": "tgt",
    "src": "The sides of the casket were covered with heavy black broadcloth , with velvet
    caps , presenting a deep contrast to the rich surmountings . What is the meaning of
    surmounting ?",
    "tgt": "A decorative feature that sits on top of something .",
    "model": "ltg/flan-t5-definition-en-base",
    "task": "DM",
    "labels": [
      "Not Hallucination",
      "Hallucination",
      "Not Hallucination",
      "Hallucination",
      "Hallucination"
    ],
    "label": "Hallucination",
    "p(Hallucination)": 0.6
  },
```


Figure 1: SHROOM Data

3.2 Input Data

The features that were used from the input data are "src", "data", and "tgt". The three columns were concatenated into one column, but with a prefix of the column name attached at the front of the text, which resulted in a sentence like the following.

Concatenated Columns

```
"src": "<define> Infradiaphragmatic
 </define> intra- and suprasellar
 craniopharyngioma",
"tgt": "(medicine) Below the diaphragm.",
"hyp": "(anatomy) Relating to the
diaphragm."
```

This simply made working with the text data easier and while working with the validation dataset, no significant difference in accuracy was exhibited in this approach compared to using all three different columns. The idea was to let the attention mechanism of the transformer model of BERT's do most of the heavy lifting of figuring out the the context and relationship between the words, in this case the column names and the texts that follow (Vaswani et al., 2017). The column "model" was not used as it also did not lead to any change in the accuracy of all models and methods whatsoever. The text were split into train and validation datasets. Then the BERT or RoBERTa models were fine tuned so that it is able to achieve better results specifically for the classification of texts. The softmax activation function on the output layer of the neural network was used to get the predicted probability between 0 and 1 (as per Devlin et al., 2019). Besides complex models like BERT, classification methods such as logistic regression, SVC, and Naive Bayes were also used with word vectors created from BERT.

3.3 Data Augmentation

Overall, the pipeline was relatively simple, but one strategy that was employed to achieve better accuracy was to increase the amount of data available. The trial and validation data provided by Semeval was on the smaller side, which contributed to overfitting. The data also had the issue of being somewhat imbalanced with the non-hallucinating data in the validation dataset amounted to 295, whereas the hallucinating data amounting to 206. Data Augmentation was utilized to combat these issues. Data Augmentation is the modification, and augmentation of the input data itself. The text inside the input data is sometimes dropped randomly, replaced by synonyms, or words can be randomly inserted, thus creating more sentences with labels to work with. As dropping or inserting random words seemed detrimental as described by previous studies (see e.g. Wei and Zou, 2019), for this task, the data augmentation was restricted to adding data where words had been replaced by synonyms for a subset consisting of 10% of the total data.

Before Data Augmentation

```
(idiomatic, intransitive) To begin a new
endeavor with vigor.
```

After Data Augmentation

```
idiomatic intransitive to start out a new
endeavor with vigor
```

4 Results

Refer to Table 1 for the results from the initial attempt. The accuracies there are what was achieved after using a simple 80-20 train and test split on the data. Models like BERT and RoBERTa were fine-tuned while regular classification methods such as logistic regression, SVC and Naive Bayes were used with the word embedding vectors retrieved from BERT. After reviewing the results, the predictions made with BERT were submitted to Semeval, as it had the highest accuracy. However the submitted results actually achieved an accuracy of only 60%.

Table 1: Accuracy of Models/Methods

Model	Accuracy (%)
BERT	80
RoBERTa	76
SVC	50
Naive Bayes	48
Logistic Regression	46

4.1 Probability of Hallucination

A Spear-man correlation score of the expected train and test data obtained from the soft-max layer of BERT resulted in a 0.64. But the actual submission spear-man correlation coefficient was a 0.23. Compared to the top ranked team whose numbers were above 0.7, a 0.23 is a bit underwhelming, and has lots of room for improvement.

5 Conclusion and Limitations

Some potential improvements that could be employed are the following.

5.1 Cross Validation

First, cross validation instead of a simple train and test split. Though data augmentation allowed for more data, which in turn made a simple train and test split theoretically suffice, since the more data one has the less likely it is that a train test split, by pure luck, can affect the accuracy significantly, simply trying out cross validation could have led to more insight on the actual accuracy on the validation data (Bates et al., 2022).

5.2 Reduce Over-Fitting

Second, methods of reducing over-fitting. It is highly likely that BERT was being over-fit with the input data, considering how the BERT model using only the validation dataset with a train and test split resulted in a 0.8 accuracy, but a 0.6 with the final test data. To counteract over-fitting, lasso regression could be incorporated to add a penalty term for high variance (Ranstam and Cook, 2018).

5.3 Ensemble Learning

Third, the "task" column was not utilized as it did not impact accuracy in, but perhaps a different approach could have been to separate data based on tasks and then to feed those data to the pipeline. Which means a holistic ensemble learning model, that uses multiple models, whether it be using the same model or different ones, in order to get a more generalized correlation score that leads to less over-fitting can be a great method. This holistic approach can lead to not just better accuracy, but also a better spear-man correlation score. The current data pipeline did not utilize the probability feature of the input dataset, and an ensemble learning pipeline that can utilize the probability feature properly, alongside the stacking of generalization could be a way of achieving better spear-man scores. (Su et al., 2013)

5.4 Better Utilization of Data Augmentation

Fourth, a more thorough utilization of data augmentation. In supervised machine learning, limited data often leads to over-fitting, which is precisely why data augmentation was the key strategy to counter the issue, but a more thorough and systematic approach to utilizing data augmentation seems to be the key. In the final attempt, the amount of data point was a 1000 each for hallucinating text and non-hallucinating text, for a total of 2000. Perhaps starting with 2000, then 10,000, then 20,000, while trying out different strategies of how the data is augmented like random deletions and addition of words, instead of just relying on replacement of words with synonyms, would have definitely benefited this research immensely.(Ying, 2019)

6 Code

https://github.com/esohman/SemEval2024

References

Stephen Bates, Trevor Hastie, and Robert Tibshirani. 2022. Cross-validation: what does it estimate and how well does it do it?

Davide Castelvecchi. 2016. Can we open the black box of ai? *Nature*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Katherine Lee, Orhan Firat, Ashish Agarwal, Clara Fannjiang, and David Sussillo. 2019. Hallucinations in neural machine translation.

J Ranstam and J A Cook. 2018. LASSO regression. *British Journal of Surgery*, 105(10):1348–1348.

Ying Su, Yong Zhang, Donghong Ji, Yibing Wang, and Hongmiao Wu. 2013. Ensemble learning for sentiment classification. In *Chinese Lexical Semantics*, pages 84–93, Berlin, Heidelberg. Springer Berlin Heidelberg.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6383–6389, Hong Kong, China. Association for Computational Linguistics.

Xue Ying. 2019. An overview of overfitting and its solutions. *Journal of Physics: Conference Series*, 1168(2):022022.

Zijun Zhang. 2018. Improved adam optimizer for deep neural networks. In *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*, pages 1–2.