Stanford CS Course + Quarter Classification based on CARTA Reviews

Stanford CS224N Custom Project

Enok Choe
Department of Computer Science
Stanford University
echoe720@stanford.edu

Juben Rana
Department of Computer Science
Stanford University
stilakid@stanford.edu

Abstract

Given a new student review for a possible class, we predict the Stanford class and quarter most associated with that user review. To achieve this, we created our custom dataset through Stanford CARTA platform. We created a dataset that composes of all available reviews of Stanford computer science courses (209 courses) to train our baseline model (basic LSTM), followed by two transformer models (BertModel and DistilBertModel), then a generative model (FLAN-T5). From the initial results we observe that the BERT models and FLAN-T5 performed better than the LSTM baseline model, and the BERT models outperform FLAN-T5 in both course and quarter classification. Interestingly, BERT model significantly outperforms DistilBERT model only in the course classification task.

1 Key Information to include

- Mentor: Tathagat Verma
- Contributions: We both worked on compiling the dataset and experimenting with each of the models together.

2 Introduction

Even though text classification is a well-established task, its application in finding patterns within Stanford courses and quarters is a unique work that has never been done before. Currently, there is an official platform for all students and faculty to view student feedback to courses (both through CARTA and AXESS), but there is little to no outside work that utilizes these data for potential applications of the future.

We embarked in this work with a recognition that the results of this work could have immense positive impact in assisting both the students and faculty for today and the future, being able to more deeply and practically analyze the states of classes and student feedback regarding specific courses, as well as discovering potential patterns throughout the past few years regarding each quarter.

With this context in mind, we aimed to also compare performances of baseline LSTM models, state of the art transformer models such as BERT, and generative models such as FLAN-T5, to see if any of these approaches perform better over the other- as it would not only help with this particular task of course / quarter prediction, but also perhaps would reveal unique attributes regarding the compared models themselves on a more technical level.

3 Related Work

3.1 Discriminative vs Generative Models

Traditionally, we have been using discriminative language models for simple NLP tasks such as text classification. Since such models learn boundaries between different categories within a dataset, their success has been conditioned on the abundance and stability of training data. Generative models on the other hand do not have these restrictions as they learn the distribution of the data in the dataset instead of the boundary. (Yogatama et al., 2017) All things considered, these are two different powerful ways of approaching the same task of text classification.

Discriminative models have lower asymptotic rates of error than generative models with the caveat that they approach these lower rates slower than generative models. Therefore, it is accepted that generative models are better than their discriminative models for smaller datasets. For larger datasets, the results are reversed. (Ng and Jordan, 2001) There are recent empirical data to back these up as well. (Ding and Gimpel, 2019)

3.2 Baseline for Finetuning BERT

BERT is a discriminative model. Hence, as described above, it is susceptible to catastrophic forgetting and further instability resulting from smaller datasets. However, researchers have discovered that these two potential reasons for the observed lack of stability during finetuning does not completely explain this instability. In fact, optimization problems, especially those that eventually lead to vanishing gradients, play a huge role in causing this phenomenon. Based on their analysis, they proposed hyperparameters that would give finetuning BERT-based models more stability. (Mosbach et al., 2021)

As per their result, they proposed the following guidelines:

1. Chose smaller learning rates.
2. Use bias correction to overcome vanishing gradients during the early stages.
3. Increase the number of epochs and train till the training loss is almost zero.

Based on the guidelines, they suggested a simple baseline scheme:

- Use ADAM optimizer with bias correction.
- Choose a learning rate of 2e-5.
- Train for 20 epochs.
- Increase the learning rate linearly for the initial 10% of the steps.
- Decrease the learning rate linearly after the initial 10% of the steps.
- Keep the rest of the hyperparameters unchanged.

It is interesting to note, however, that the number of epochs suggested by this team of researchers differs greatly from the one suggested by the researchers working on the original BERT model, according to whom, the following range of values for the hyperparameters excelled across different tasks: (Devlin et al., 2019)

- Use a batch size of 16 or 32.
- Choose a learning rate of 5e-5, 3e-5, or 2e-5.
- Train for 2-4 epochs.

Nevertheless, due to both time and resource constraints, we will be sticking to a lower number of epochs for our experimental analysis. As for the learning rate, an aggressive value of about 4e-4 is already enough to trigger catastrophic forgetting, (Sun et al., 2020) which explains why both the papers referenced above suggest learning rates less than or equal to 5e-5. Within this range of values, we have conducted further analysis using different learning rates to identify the most optimal learning rate for our task.

4 Approach

First, we created and evaluated a custom baseline LSTM model for this task of text classification. We constructed the LSTM model to have a sequential layer, embedding layer, two LSTM layers, and a softmax layer, attempting different variations or combinations of these layers.

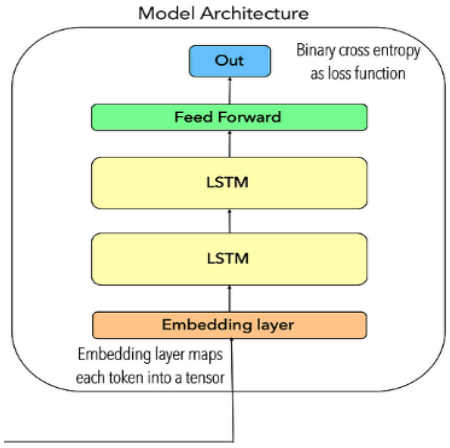

Figure 1: LSTM Model Architecture

Then, we attempted to fine-tune BERT / DistilBERT models derived from pretrained models in huggingface (distilbert-base-cased and bert-base-cased) and fine-tuned it using our dataset.

We made sure to shuffle the data that we feed into our models, which is especially important for these discriminative models to avoid over-training. As discussed in the literature review, they are incredibly bad at learning all data from one class before moving onto the next.

We also ran an analysis of the optimal learning-rate to use from the range recommended for the BERT model before running it on our CS dataset.

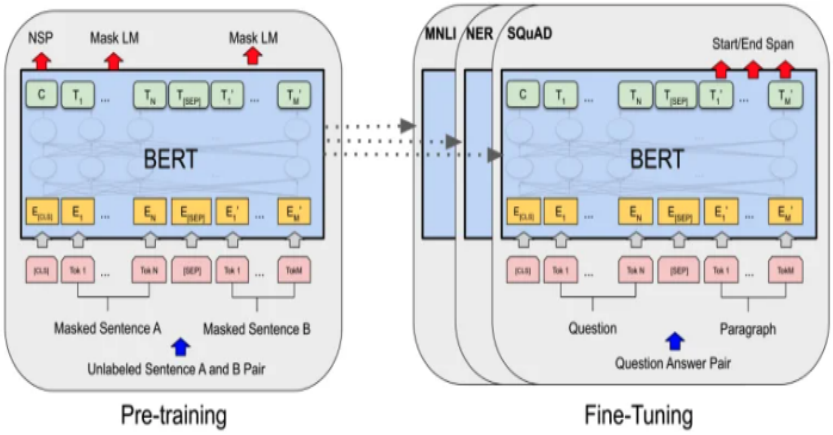

Figure 2: BERT Model Architecture

We did likewise with the FLAN model in terms of deriving an existing pre-trained model and fine-tuning (google/flan-t5-base). However, given that the FLAN model is a generative model, we added a prefix prompt as a part of data processing, adding the prefix "Please output a number from 0 to (NUM LABELS - 1) based on the following course review: " to every review. Then, we took the generated output, which would be a single integer between 0 to NUM LABELS - 1 in a stringified format, turned it into an integer, and compared with the integer labels which represent the course / quarter depending on what we are predicting to evaluate.

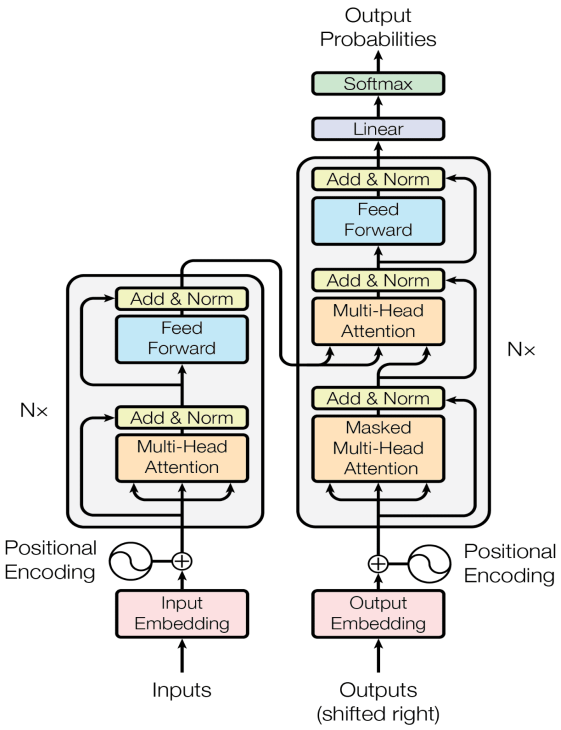

Figure 3: FLAN-T5 Model Architecture

5 Experiments

5.1 Data

We created our own datasets by developing scripts in JavaScript that scrapes relevant resources from Stanford CARTA. CARTA contains reviews for more than 15,000 classes, and they are segregated by year and quarter. We scrapped the links to all the class report web pages, using which we automatically loaded each of those web pages and scanned through the html of the reviews section to extract all the reviews and quarters associated with that class. The output files were in json format, which we used to create a singular CSV dataset.

We combined the data and made two different datasets:

1. A dataset that contains reviews and quarters of when those reviews were written as input / features, and the course name as the label
2. A dataset that contains reviews and courses associated with the reviews as input / features, and the quarter of when those reviews were written, as the label.

   The dataset contains reviews of all classes at Stanford, though we ended up using a subset of this dataset for this particular project (reasoning elaborated in experimental details), evaluating the results for all computer science classes (209 classes in total).

The dataset contained over 294,801 reviews, out of which 8,649 reviews had more than 128 words. Due to this, we decided to use use a max token length of 128 when we preprocessed the dataset for finetuning our models.

5.2 Evaluation method

For our evaluation method, we used classification accuracy, f1 score, precision, and recall metrics. Since different classes can have varying number of student reviews due to factors such as the size

of the class and the demand for it in general, we cannot rely completely on accuracy and precision, which is why we used f1 scores as well.

5.3 Experimental details

First, for BERT models, we began our experiment with the default model configurations that came with the BERT models from Huggingface as a starting point.

Then, we tested out several different learning rates that fell in the range recommended by researchers who had worked on the original BERT model. For comparison, we fixed the batch size to 16 and the number of epochs to 3.

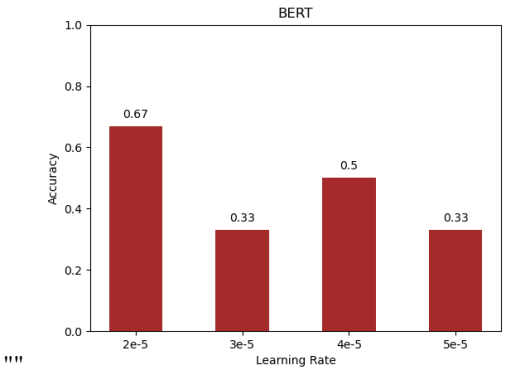

Figure 4: BERT: Learning Rate to Accuracy

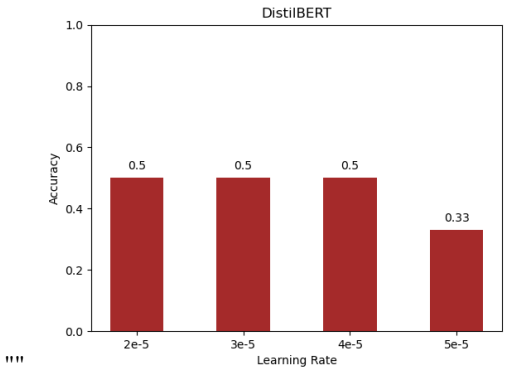

Figure 5: DistilBert: Learning Rate to Accuracy

Additionally, we tested out two different batch sizes of 8 and 16 while keeping other model parameters constant and also the number of epochs that was ideal for fine-tuning the model.

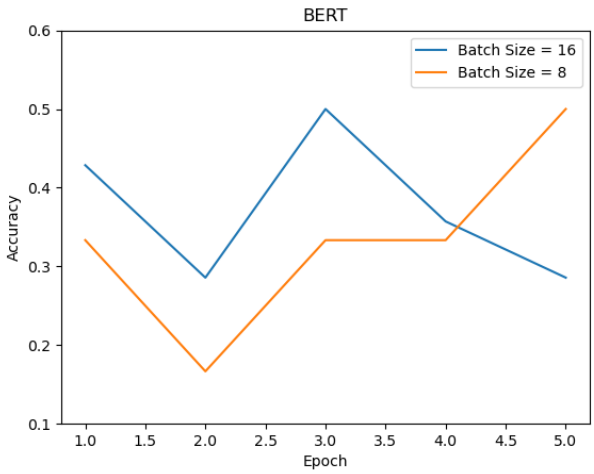

Figure 6: BERT: Epoch to Accuracy Relation

The parameters we arrived to as the best performing for BERT and DistilBERT:

- Learning rate: $2e^{-5}$
- Number of epochs: 3
- Test size: 0.3
- Batch size: 8

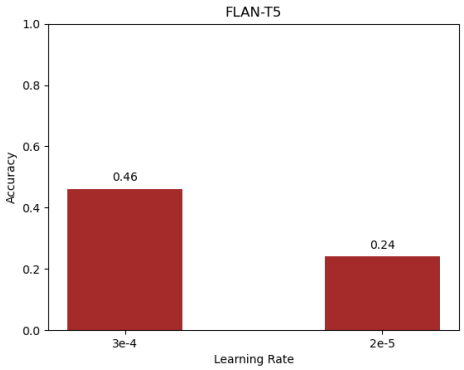

Figure 7: FLAN-T5: Learning Rate to Accuracy

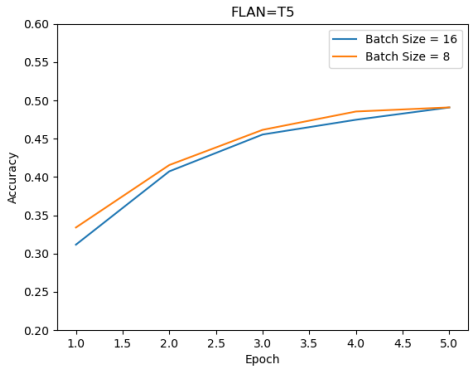

Figure 8: FLAN-T5: Epoch to Accuracy

The parameters we arrived to as the best performing for FLAN:

- Learning rate: $3e^{-4}$
- Number of epochs: 5
- Test size: 0.3
- Batch size: 8

5.4 Results

Best results for Course Classification

Model	Accuracy	Precision	Recall	F1 Score
LSTM	0.24	0.28	0.23	0.26
BERT	0.67	0.67	0.67	0.67
DistilBERT	0.50	0.50	0.50	0.50
FLAN-T5	0.51	0.51	0.51	0.51

Best results for Quarter Classification

Model	Accuracy	Precision	Recall	F1 Score
LSTM	0.11	0.12	0.12	0.15
BERT	0.25	0.25	0.25	0.25
DistilBERT	0.29	0.29	0.29	0.29
FLAN-T5	0.18	0.18	0.18	0.18

There are elements of the results are both expected and different from what we expected, but overall, the results were better than what we expected.

First, it is certainly better in that there are 209 total labels for the course classification- which means that in random prediction, the accuracy of prediction would be $\tilde{0}.004$, but the all the metrics were over 0.23 for all models. Likewise, for the quarter classification, there are 23 total labels which means that the accuracy of prediction would be $\tilde{0}.043$, but the predictions were over 0.15 for all models.

Second, as expected, the BERT and the FLAN-T5 models outperformed the baseline LSTM model. It made sense given that the latter two types of models are pretrained on much larger datasets (e.g. BERT was pretrained on a large corpus of English data, and FLAN was pretrained on large corpus of various languages as well as on more than 1000 additional tasks) while the LSTM model was trained from scratch with our custom dataset. We had this assumption also given that BERT and FLAN models are both transformer models that have more complex architecture and larger parameter spaces than an LSTM model.

However, what particularly surprised us was that while we initially expected the FLAN-T5 model to perform better than BERT for both classification tasks but were proven wrong. Our reasoning was

that BERT has an encoder-only architecture while FLAN-T5 has an encoder + decoder architecture on top of it having been pre-trained on larger datasets. Nevertheless, we can see that FLAN-T5 actually performed worse than BERT in both course and quarter classification tasks.

Finally, the most unexpected result of all was that BERT outperformed DistilBert by a significant margin (0.67 vs 0.50 accuracy) in the course classification task despite the fact that DistilBert is designed to be more efficient than BERT while maintaining similar performance.

6 Analysis

First, given that BERT performed better on both course and quarter classification compared to FLAN-T5, we can speculate why this may be the case. Given that both BERT and FLAN-T5 use bidirectional context modeling (taking preceding and succeeding words into account), it appears unlikely that it is a result of the two models differently processing the semantics of the reviews. However, there is a significant difference between the two in that while the BERT model directly outputs a prediction based on the tokenized review as input, the FLAN-T5 model does have to go through an additional preprocessing step of attaching a prefix to each review, instructing the model to output a certain type of output (e.g. please 'output a number between 0 and 208', for 209 label course classification), which means that each review that is being tokenized for the FLAN-T5 input actually becomes longer in length with the prefix, and may create more room for confusion for the model in isolating and evaluating the reviews themselves.

Moreover, the most interesting result out of all is that BERT somehow performs significantly better than DistilBert, specifically in the course classification task. We could speculate that course classification specifically has specific review features attached to it, requiring a deeper understanding of language or dealing with complex linguistic phenomena for the model- provided that BERT's larger capacity and ability to capture more nuanced features compared to DistilBERT may allow it to capture more semantic patterns that DistilBERT misses.

An important disclaimer to note here is that this evaluation outcome (of BERT significantly outperforming DistilBERT) is not consistently reached upon each attempt of evaluation with the same parameters, so it may not be a result that would be yet appropriate to be considered for a general conclusions regarding distinctions between BERT and DistilBert in this course classification task. However, it is worth a further investigation with other classification tasks as well as more quantity and quality of datasets.

Finally, provided the observation that smaller learning rates could have a large positive impact in the outcome of prediction for the models, we could deduce that there are certain specific minute but complex combination of features and patterns in the review apart from simple features that significantly influence the chances of predicting a course or quarter correctly.

7 Conclusion

In conclusion, for the course and quarter classification task given CARTA student reviews for Computer Science classes at Stanford, we observed that using various models yield varying results. After training our baseline model (basic LSTM), followed by fine-tuning two transformer models (BertModel and DistilBertModel), then fine-tuning a generative model (FLAN-T5), we observe that the BERT and FLAN-T5 models performed significantly better than the LSTM baseline model, with BERT performing better for both course and quarter classification tasks. Moreover, BERT model significantly outperformed DistilBERT model only in the course classification task.

As future work, we would like to complete the course and classification with the entirety of Stanford courses, which we were not able to complete due to computing resources and technical difficulties. We would like to observe whether this changes the performance outcome for either course or quarter classification between the models. In particular, we would like to observe if the FLAN-T5 model can catch up entirety to the BERT model result in the two classification tasks, as well as whether the discrepency of results between DistilBert and Bert disappear or not, with more diversity and quantity of reviews.

References

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

Xiaoan Ding and Kevin Gimpel. 2019. Latent-variable generative models for data-efficient text classification.

Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. 2021. On the stability of fine-tuning bert: Misconceptions, explanations, and strong baselines.

Andrew Ng and Michael Jordan. 2001. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Advances in neural information processing systems*, 14.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2020. How to fine-tune bert for text classification?

Dani Yogatama, Chris Dyer, Wang Ling, and Phil Blunsom. 2017. Generative and discriminative text classification with recurrent neural networks.