# Love It or Hate It?

*Author Sentiment Prediction*

By Ronith Mudhuganti(113244355) & Hong Wei Chen(113361519)

# 1 Project Overview

As news reporting tends to be biased and influenced by an author's personal views, it's important for us to identify the author's sentiment towards the subject being reported on to better evaluate the author's objectivity and credibility. Being able to determine the sentiment of news articles quickly also enables us to gauge the general sentiment towards an entity across multiple media. The overall goal of this project is to find an effective way of classifying author sentiment towards an entity in news articles by using large language models. We will be using the PerSent dataset, taking as inputs news articles and the entity being reported on, and classify the author's sentiment towards that entity as either positive, neutral, or negative.

As stated in the paper, "Author's Sentiment Prediction", this is a difficult task because of the amount of irrelevant information in an article that doesn't contribute to sentiment analysis, and the model has to learn how to aggregate local information in paragraphs to come up with a document sentiment (Bastan et al., 2020). This is due to the nature of news articles, as there will be paragraphs where the entity being analyzed isn't even mentioned, and paragraphs with conflicting sentiments towards an entity. Thus, combining all of this information to get an overall sentiment of a news article becomes a difficult task.

To address this task, we are using three approaches: fine-tuning a small sized pre-trained model on this task, Zero-Shot classification with a pretrained model, and Few-Shot classification with a pretrained model. All of these methods take advantage of the power and robustness of transformer-based large language models that were shown to outperform previous models on a wide array of different tasks.

To evaluate the effectiveness of each of these 3 approaches, we will be measuring their performance metrics, specifically their precision, recall, and F1 scores on the random_test dataset provided in the PerSent dataset. This performance analysis will reveal which of these 3 approaches is best suited for this specific classification task. We will also be identifying and analyzing the types of inputs that our model performs poorly on.

The results show that out of the 3 approaches, fine tuning a small sized pre-trained model showed the best performance on this task. However, the Zero-Shot model showed a significant performance improvement when only considering positive and negative sentiments and omitting neutral sentiment.

# 2 Ideas

## 2.1 Fine Tuning A Small Sized Pre-Trained Model

Our first approach to this task is fine-tuning a small sized pre-trained model using the train, dev, and test datasets. Specifically, we used Hugging Face's "distilbert-base-uncased" model. DistilBERT is a BERT based language model. Its size is 40% less than BERT, it has around 97% of BERT's language understanding capabilities and is 60% faster.

We fine-tune DistilBERT to perform a multiclass sentiment analysis task by loading a pre-trained distilbert-base-uncased model, tokenizing the news article document, and converting the tokens and sentiment labels (negative, positive, neutral) into tensors grouped by their respective input IDs. Then we use the train data and dev data to train and evaluate the model, respectively. We compute the performance metrics such as precision, recall, f1, and loss to determine the best model to use for testing. We then test the best model using the test data to get our final performance metrics. This final analysis tells us how effective fine-tuning a small pre-trained language model, such as DistilBERT, is at predicting author sentiment in news articles.

## 2.2 Zero-Shot Performance On A LLM

Our second approach to this task is doing Zero-Shot classification on the test dataset using a large language model. Specifically, we used Hugging Face's "flan-t5-xxl" model. T5 is an encoder-decoder model originally introduced back in 2019 by Google. Flan-T5, released by Google in 2022, is an improved version of the original T5 model. Although both models contain the same amount of parameters, Flan-T5 has been fine tuned on more than 1,000 tasks, greatly improving its reasoning skills.

Zero-Shot classification is a technique that prompts a model to classify unseen classes without additional training data, unlike traditional models that require training on a large amount of labeled training data before being able to start making correct classifications. This feature is exclusive to large language models with millions of parameters, and in T5's case, more than 11 billion. Given the large parameter count, we expect Flan-T5 to perform moderately well on challenging sentiment classification tasks like this one. For this task, we will be prompting the Flan T5 model with, "classify the author sentiment on [target entity] as positive, neutral, or negative: [document]".

### 2.3 Few-Shot Performance On A LLM

Our last approach to this task is doing Few-shot classification on the test dataset using a large language model. To remain consistent with Zero-shot, Hugging Face's "flan-t5-xxl" model was used for this approach as well. Unlike Zero-Shot, the Few-shot learning technique provides the language model with a small amount of labeled data for it to quickly adapt to an unseen task. For this task, we will be providing the model with 3 labeled examples, one for each of the 3 classes, with the hope of improving its performance since it will be more accustomed to the task compared to Zero-Shot learning.

## 3 Experimental Setup

### 3.1 Models

We have used two transformer-based models to perform a multi-class author sentiment analysis task. The LLM model we have used to perform Zero Shot and Few Shot is the FLAN-T5-XXL model based on the T5 architecture. The smaller model we have used to fine-tune is the DistilBERT-base-uncased model based on the BERT LLM. The Distillbert model is a pre-trained model that can be fine-tuned to perform a specific task while the Flan model requires Few Shot/Zero Shot learning. Distillbert has only 6 layers as opposed to Flan's 24 layers, as it is much smaller than a typical Large Language Model

To train the Distillbert model, we had to pass in some hyperparameters in order to be able to fine-tune it to perform our specific task. Such parameters included batch size (set to 16) and epochs (set to 3). We also used an optimizer which had a learning rate of 0.00003 and epsilon value of 0.00000001. With these hyperparameters set, we utilized Colab's GPU to train and test the data, taking around 10 minutes each to complete, respectively.

### 3.2 Dataset

For this task, we are using the PerSent corpus. The data contains around 5k documents and 38K paragraphs annotated on the author's sentiment towards the main entity in the news article. Each data file contains a document index, article title, target entity of the article, document of article, masked target entity document, true overall article sentiment, and sentiment of each paragraph up till 15 paragraphs. There are 3355 training instances and 578 validation instances, all of which were used in 210 training batches and 37 validation batches, respectively.

We have been given four files - train_data.csv, val_data.csv, random_test.csv, and fixed_test.csv. The training data will be used to fine-tune the language model, the val data will be used to evaluate the training, and finally the test data will test on randomly organized test instances.

### 3.3 Evaluation Metrics

To evaluate the training model, we used automatic validation using the dev data set. This evaluate method compares the performance metrics (precision, recall, f1, loss) of the training data to its own to determine the true analytics. The performance metrics of the evaluate method are the true measurements of the model. This method also helps in determining the best model of each epoch, which the model we will end up testing on. Finally, we compare the 3 different models by their performance on the random_test dataset.

## 4 Results

| Method | Precision | Recall | F1 |
|---|---|---|---|
| DistillBERT | **0.5895** | **0.5895** | **0.5895** |
| T5-Flan (Zero Shot) | 0.47 | 0.47 | 0.47 |
| T5-Flan (Few Shot) | 0.46 | 0.46 | 0.46 |

Table 1: Comparison of the three methods on the Random test set of the PerSent dataset: DistillBERT in the fine-tuned model achieves the best performance across all measures. T5-Flan (ZS) is the zero shot use of T5-Flan and T5-Flan (FS) is the few shot use of T5-Flan with K=3 examples.

# 5 Analysis and Discussion

Here we will be doing further analysis on the results of the Zero-Shot model to find out what types of inputs the model succeeds and fails on.

## 5.1 Long Inputs

One hypothesis on the kind of inputs the Zero-Shot model fails on is long inputs that exceed the token limit of the Flan-T5 model. When running the model on the test dataset, we limited the article length to be no more than 3500 letters, meaning longer articles are truncated. Important sentiment information towards the end of a long article may be lost which causes the model to make wrong classifications. For example, in document 3976 in the "random_test" dataset, the true sentiment of the article is neutral, but the model identified the article to be negative due to some sentences at the beginning of the article.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| Positive | 0.65 | 0.69 | 0.67 | 254 |
| Neutral | 0.46 | 0.06 | 0.11 | 185 |
| Negative | 0.24 | 0.82 | 0.38 | 62 |
| accuracy | | | 0.47 | 501 |
| macro avg | 0.45 | 0.52 | 0.38 | 501 |
| weighted avg | 0.53 | 0.47 | 0.42 | 501 |

As we can see from the results, however, the length of the articles doesn't seem to be an important factor, as the F1 scores are almost identical when we only consider articles with length less than 3500 characters. This shows that there are probably other reasons that the model made the wrong predictions.

## 5.2 Neutral Sentiment

Another hypothesis on the kind of inputs the Zero-Shot model fails on is neutral sentiment articles. Given the length of articles, it's difficult for the model to come to a neutral conclusion when some sentences/paragraphs in the article are bound to be either positive or negative. The model tends to lean towards classifying an article as either positive or negative instead of neutral. For example, in document 3951 in the "random_test" dataset, the true sentiment of the article is neutral, but the model classified this article as negative, possibly due to the critical second paragraph, "Critics call him a political prisoner a powerful oligarch arrested for daring to become involved in politics when then-President Vladimir Putin was in power".

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| Positive | 0.95 | 0.69 | 0.80 | 293 |
| Neutral | 0.00 | 0.00 | 0.00 | 0 |
| Negative | 0.44 | 0.84 | 0.58 | 73 |
| micro avg | 0.72 | 0.72 | 0.72 | 366 |
| macro avg | 0.46 | 0.51 | 0.46 | 366 |
| weighted avg | 0.85 | 0.72 | 0.75 | 366 |

As we can see from the micro average and weighted average results, when we disregard articles with a neutral sentiment and only consider articles with positive or negative sentiments, we can see significant improvement in the F1 score.

### 5.3 Mixed Paragraph Sentiments

Another hypothesis on the kinds of inputs the Zero-Shot model fails on is articles containing paragraphs that are mixed in sentiment, especially ones with both positive and negative sentiments. When the sentiment of the individual paragraphs don't agree with each other, it may be hard for the model to decide which of them represents the true sentiment of the entire article. For example, in document 4065, the overall sentiment of the article is positive towards Manuel Zelaya but the model classified the article as being negative. This might be caused by the fact that the sentiment of the very first paragraph is negative, even though the rest of the paragraphs are dominated by positive or neutral sentiments.

However, after modifying the article to delete the negative paragraph, the model still predicted a negative sentiment even though there are no negative paragraphs anywhere. This shows that there are probably other reasons that the model made the wrong prediction.

# 6 Code

https://drive.google.com/drive/folders/1qs-G5c9G1XvhZq0YxaiI3aQL0MsremQH?usp=sharing

# 7 Learning Outcomes

Through exploring the 3 ideas, we were able to gain experience working with large and small transformer-based language models in 3 different ways: fine-tuning, zero-shot, and few-shot. We were able to gain insights on the task of author sentiment analysis, specifically with the PerSent dataset, and identified what type of inputs the models were having troubles with. Although Flan T5 has a lot more parameters than DistilBERT, we learned that it doesn't necessarily translate to good zero-shot and few-shot performance, and fine-tuning and training DistilBERT showed better overall results on this specific task.

# 8 Contributions

Hong Wei Chen - Zero-Shot and Few-Shot approach, input analysis, writing report

Ronith Mudhuganti - Fine Tuning DistilBERT approach, writing report

# References

Mohaddeseh Bastan, Mahnaz Koupaee, Youngseo Son, Richard Sicoli, Niranjan Balasubramanian. 2020. Author's Sentiment Prediction. *arXiv:2011.06128*