# Noisy Student Training to identify Textual Elements in Unsupervised News Data via Argumentative Essay Pieces

A Preprint

Liew Wei Pyn[1] and Prannaya Gupta[1]

[1]NUS High School of Math and Science

## Abstract

*Noisy Student Training and Knowledge Distillation are separate techniques to boost the accuracy and noise-resistance of a given model. Although largely used on Convolutional Neural Networks (ConvNets), prior work has show their legitimacy on Natural Language Processing (NLP) tasks and specifically transformers. We design a system inspired by these techniques to craft a robust model capable of identifying textual elements in argumentative essay pieces and news pieces (particularly opinion editorials), such as the Lead, Position and Concluding Statement. Initially, we use transfer learning on multiple pretrained models retrained on a set of argumentative essays to get a set of Teacher models, after we apply selection and then follow through with data augmentation through word and sentence-based augmentation. Sequences are labelled by a teacher model and the labelled data is passed into a student model based on the RoBERTa architecture, with Layerdrop and dropout implemented, which is now retrained again to produce a student model, which repeatedly replaces the teachers and the steps take place again iteratively, allowing a set of comparable student models. We compare and contrast to find the best model capable of identifying these essays. A User Interface is later designed to visualise these models separately.*

(215 words)

# Contents

# 1 Introduction

Argumentative Essays are one of the key ways by which a student's calibre in language is assessed. Generally, this quantity is assessed based on a hour to one-and-a-half hour paper that prompts the student with any $n$ topics. These topics are based on key sociopolitical concepts, bordering on current affairs topics such as the recent Covid-19 Pandemic. These essays have many different forms, but there is often a key format, as highlighted in the diagram below:

**Lede (Lead)**: Introductory paragraph introducing the concepts, context and definitions held in the entire essay, in an engaging manner.

**Thesis (Position)**: Indication of author's point of view, summarising key points highlighted in the essay.

**Point (Claim)**: Usually statements not yet justified based on data or elaboration which are raised at the start of a body paragraph.

**Evidence, Explanation, Examples, Elaboration (EEEE)**: Statements based around relevant facts and case studies justifying one's point.

**Counterargument (Counterclaim)**: Statements highlighting opposing points of view from other perspectives and onlookers. Usually start with something like *"Critics may say..."* or *"Some may argue that..."*.

**Rebuttal**: A rebuttal aims to refute the point raised in a Counterargument, reaffirming the Position of the author.

**Conclusion (Concluding Statement)**: The author delivers some call for action or some other smaller points to iron out his Position clearly to anyone reading.

Argumentative Essays usually include a plethora of these different essay elements that are quite difficult for the average student to identify.

For instance, given this passage, taken from Leemen Chan et al:

> *Although the history of human invention spans over millennia, there have been relatively few game-changing technologies. From the manipulation of fire back in the primal age to the invention of the electronic transistor in the 1940s, these monumental developments have often been followed up by fervent exploration and a burgeoning of new technologies that stem from that initial spark.*

Here, you can state that this paragraph is the Lede, or the **Lead**. It is the statement of facts prior to a statement of opinion, and this is a very objective set of statements.

However, after this, the author writes the following:

> *While some may say Artificial Intelligence, or AI, is akin to these other inventions, I beg to disagree.*

This is a **Position**, as can be said from the subjective statement of opinion, which has yet to be substantiated by fact or evidence. However, in a different context, this can be classified as a **Rebuttal** as well, since it started by giving an alternate perspective and then admitting that this is not something that matches up with the perspective of the author. The fact is that this Position is identified based on the context laid out by prior sentences.

This contradiction can often cause confusion in modern machine learning models, since they may classify sentences without keeping track of context. Context is quite important in Natural Language Processing (NLP) Tasks, which is something many models would not excel at.

Often, students struggling with or generally writing essays use automated writing feedback tools, which, while numerous, each have their own limitations. Many of these feedback tools are unable to identify writing structures in essays, or are at least very inaccurate in their identification. Many of these tools are proprietary, with algorithms and feature claims that cannot be backed up independently, and more importantly, that are inaccessible to educators and the marginalized populace due to high costs.

In addition, automatically and consistently identifying and highlighting such elements, that too amongst hundreds of scripts, can be a tedious task for teachers, who may wish not to use these tools due to, in summary, their high levels of inaccuracy, lacking features and high costs.

Hence, our aim with this project is to develop a model capable of analysing and identifying the key argumentative elements in an English essay or even a news article[1]. We target the Minimum Viable Product (MVP)[2], particularly a Web User Interface (UI), designed post-modelling to be aimed towards teachers aiming to help their students improve their writing, and students aiming to improve their writing to prepare for the A Level General Paper and University Applications Preparation.

# 2 Methodology

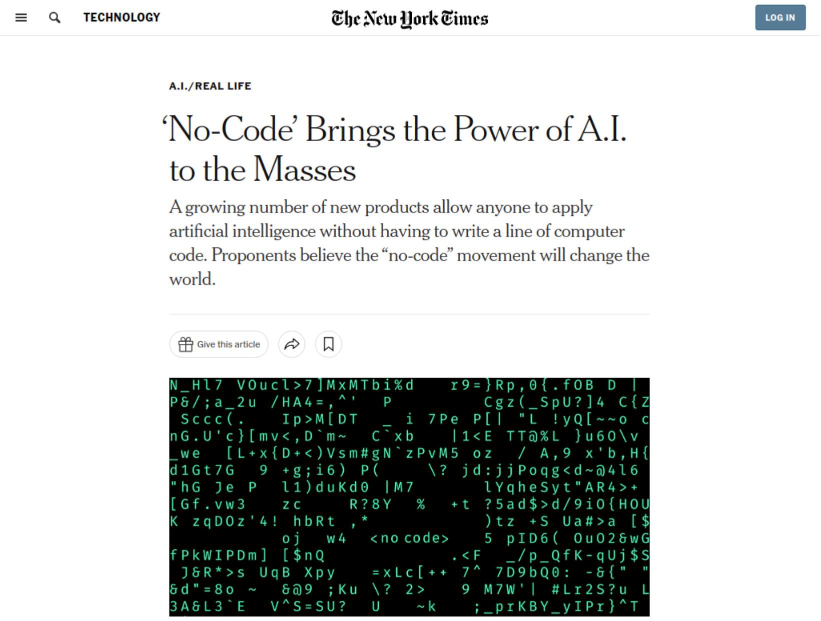
## 2.1 Data Collection

### 2.1.1 The Feedback Prize Dataset

The Feedback Prize Dataset [1] is a textual dataset containing roughly 60,000 samples of argumentative essays written by U.S students in grades 6-12. These essays have been annotated by expert raters for elements commonly found in argumentative writing. Provided by the Georgia State University in a partnership with the Learning Agency Lab, this dataset focuses primarily on student writing, and, having been labelled, is a much better candidate to use with Deep Supervised Learning. This dataset has been retrieved from the Kaggle Competition, "Feedback Prize - Evaluating Student Writing".

A sample of the essay (partial) is shown below:

> *Asking multiple people for advice can very heavily influence how you may act on certain topics. Receiving advice from multiple people is not such a bad thing as it may seem. Receiving advice from multiple people is good because it can influence your decision, teach you more about the topic, and more points of view on the topic.*
>
> *Getting more peoples' opinions can heavily impact your decision. Having more opinions means more choices you can choose. This means that the more options you have, the easier it is to do the right thing. Others' opinions can stop you from doing the wrong thing. When they tell you their end it may impact you and stop you from doing something you wouldn't have wanted to do. Others' opinions can also show you what consequences there might be for your actions. They might tell you the possibilities of what might happen when you say that. People opinions can greatly impact what you do.*



A.I./REAL LIFE

'No-Code' Brings the Power of A.I. to the Masses

A growing number of new products allow anyone to apply artificial intelligence without having to write a line of computer code. Proponents believe the "no-code" movement will change the world.

**Figure 1:** An example of an article considered for this project.

### 2.1.2 The New York Times Article Database

Whilst it is agreeable that the Feedback Prize Dataset (2.1.1) is a suitable choice for training, given the fact that is labelled and, in that way, a pretty unique dataset, the essays are also made up of a relatively broken form of English, given the fact that they are written by younger individuals, hence it possesses quite a bit of noise within the textual samples. The samples are also not ideal for models that are not necessarily very noise-resistant, hence The New York Times Article Database is retrieved for noise removal.

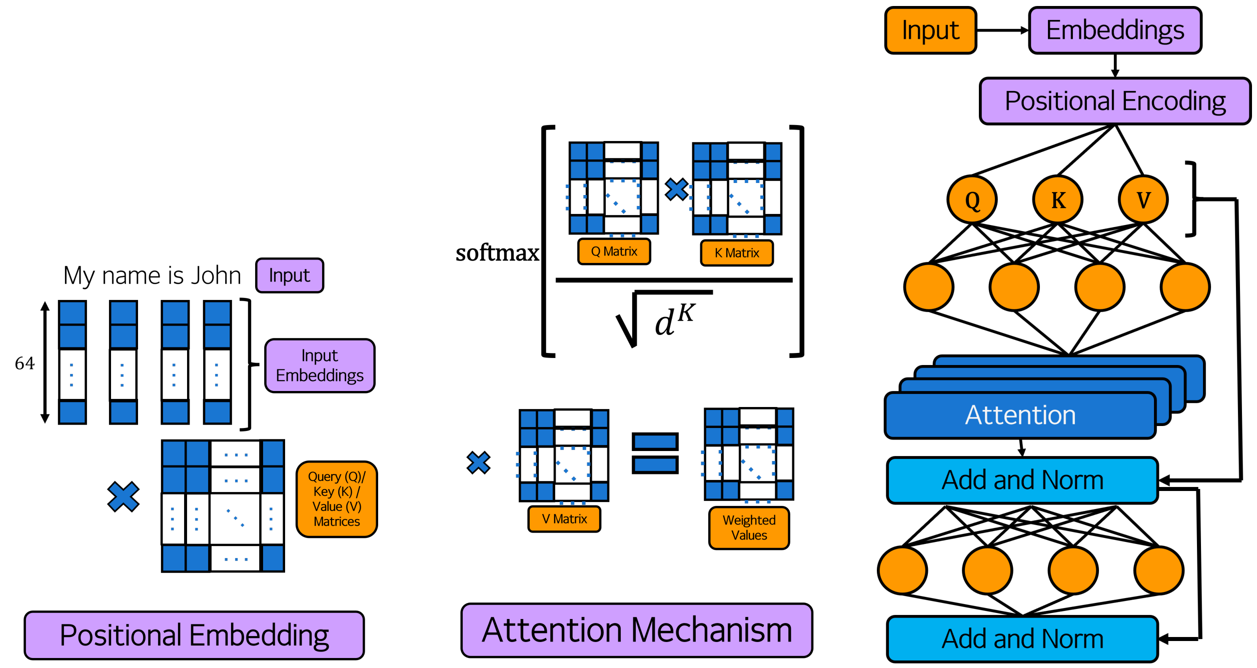
The New York Times Article Database is used to introduce relative noise in terms of the variety of samples, while also sticking to perfect English (since it is a publication). Data is retrieved based on a dynamic set of New York Times article URLs, which is also retrieved programmatically from every topical section referenced in their website. The article texts, in addition to the title, are stored into a database, and the articles are filled within a set of text files in a similar format to the aforementioned Feedback Prize Dataset. There are no labels available for this dataset, and our teacher model will address this issue later. Just as a note, this database is made of texts, not separated into sentences. Hence, NLTK's [2] `PUNKT` Sentence Tokenizer [3] is used to segment the articles into sentences.

## 2.2 Algorithms Employed

## 2.3 The Transformer

The transformer is a well known model architecture introduced by Vaswani et al.[4] to solve a variety of NLP tasks. Because the use of transformers is commonplace and our implementation does not deviate from traditional transformer architectures, we will omit an exhaustive description of the transformer, and instead refer readers to the incredibly informational resources such as the original paper [4] and online guides such as The Annotated Transformer. [3]

[1] You can find our Research Repository at `https://github.com/treeai/writingAnalysis`
[2] You can find our MVP Repository at `https://github.com/treeai/writingAnalyzer`
[3] `http://nlp.seas.harvard.edu/2018/04/03/attention.html`



**(a)** A representation of the embedding done by transformers.

**(b)** A representation of the attention layers in transformers.

**(c)** The overall process of the encoder in a transformer.

**Figure 2:** Three simple graphs

### 2.3.1 LayerDrop

Stochastic Depth is a technique initially introduced by Huang et al. [5] to accelerate training of very deep ResNets [6] for image tasks with depth well beyond 1200 layers. During training time, a random subset of layers are dropped and bypassed, while the entire deep network is used during inference time. This reduces training time substantially as well as improves test error significantly by reducing gradient vanishing and diminishing forward flow.

Inspired by work on Stochastic Depth, the LayerDrop technique was introduced by Fan et al. [7], applied to transformers for sequence modelling tasks. While Stochastic Depth is interested in accelerating the training of very deep ResNets, LayerDrop intends to reduce the final model complexity of transformers. Utilising the same principle of structured dropout of layers in the encoding and decoding blocks of the transformer during training time, the authors found that the transformer models became more resistant to predicting with missing layers, and they were able to prune the larger model during inference time to obtain a smaller sub-network from any depth without fine-tuning with limited impact on performance.

We note that this metric applies to residual layers (depicted with the function $\mathcal{R}$), which can be formally defined as shown below:

$$x_{i+1} = x_i + \mathcal{R}(x_i)$$

In LayerDrop, a drop rate, $p^* \in [0, 1]$, is introduced, based on a set of $N$ groups and pruning set of $r$ groups, based on the following definition:

$$p^* = 1 - \frac{r}{N}$$

On general principle, $p^*$ is set to be either $0.2$ ($r = 0.8 \cdot N$) or $0.5$ ($r = 0.5 \cdot N$), although Fan et al [7] suggests using $p^* = 0.5$ for models aimed for small inference-times.

Now you define a variable $Q_i \in \{0, 1\}$ which represents a Bernoulli Distribution similar to dropout, which is used to ascertain whether or not to skip or use the layer. $Q_i$ is dictated by the following equations (by Wang et al [8]):

$$P(Q_i = 0) = p \implies P(Q_i = 1) = 1 - p$$
$$x'_{i+1} = x_i + Q_i \times \mathcal{R}(x_i)$$

### 2.3.2 Knowledge Distillation and Expansion

Knowledge Distillation [9] refers to the process where a larger, difficult to deploy model transfers its learned knowledge about its task to a smaller, faster model. This is typically used to reduce large sized models into computationally less expensive models for real-world deployment. The larger model is often termed the "teacher" model, and the smaller model the "student". Existing work done in the field of knowledge distillation for NLP is limited and focuses on feature-basead knowledge, wherein the student model tries to replicate the feature embeddings (intermediate layers) of the teacher model. This includes work done by Shin et al. [10] wherein they use knowledge distillation and ensemble methods for word embedding distillation, as well as Jiang et al. [11] that demonstrates knowledge distillation from pretrained BERT models achieves improvements in performance.

In related work in the usage of teacher student models for images, Xie et al. [12] introduces the semi-supervised Noisy Student training approach, demonstrated on ImageNet [13], which while similar to Knowledge Distillation in terms of utilising soft labels and the teacher student model, has some key differences. Knowledge Distillation focuses on model compression, without use of unlabelled data or noise injection. On the contrary, Noisy Student makes use of unlabelled data, noise injection and student models that are not smaller than the teacher model. The authors describe their approach intuitively as "knowledge expansion", where they enable the student model to outperform the teacher model.

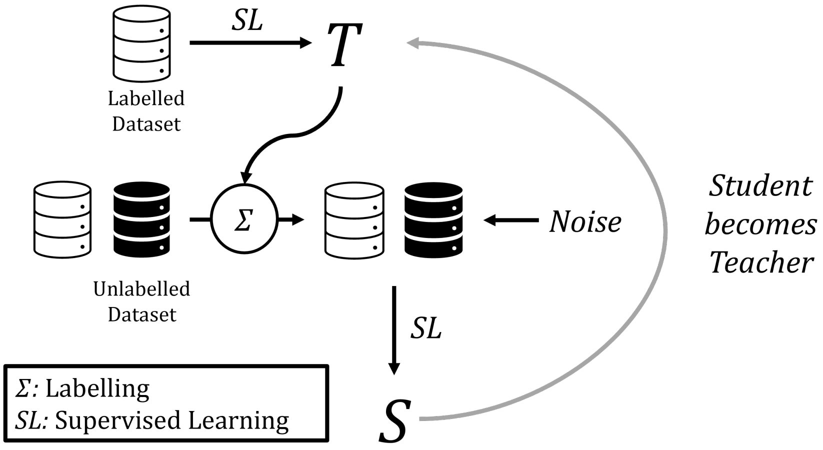


**Figure 3:** How Noisy Student Training works.

Noisy Student injects noise into the dataset and model through methods highlighted for as follows:

**Input Noise**

- Random Corrupting and Augmentation
- Mild Rotation
- Color/Jitter Changes

**Model Noise**

- Structured and Unstructured Dropout (skipping of neurons)
- Stochastic Depth (skipping of residual layers)

Our work expands upon that of Xie et al. in the field of Natural Language instead of Image Classification with some modifications. In terms of input noise, for obvious reasons we cannot rotate natural language (if you find a way do let us know), and we instead perform textual modification via synonym substitution and sentence restructuring. In terms of model noise, we make use of Dropout and LayerDrop [7] in place of Stochastic Depth. The use of stochastic depth in this case intends to introduce model noise, and LayerDrop achieves the same effect during training time for the transformer architecture, while additionally allowing us to prune the final student models without major loss in performance. The resulting student models are hence smaller, in contrast to the Noisy Student technique where student models are intentionally larger than the teacher model.

## 2.4 Training Process

### 2.4.1 Teacher Model

A teacher model is obtained based on prior work done on the project. For instance in Xie et al [12], they perform Noisy Student on the previously established EfficientNet model architecture [14]. However, due to the relative newness of our dataset, not much prior work has been done to justify using a specific architecture. We instead train both the teacher and student models in order to attain full autonomy, and the teacher model trained is based on much larger models which can gain higher accuracy with ease. We hence utilise the following model architectures:

- DistilGPT2[15]
- DistilBERT[16]
- BERT[17]
- RoBERTa Base[18]

These are more openly available models derived by researchers from Google AI, Meta AI, OpenAI and Hugging Face which can be retrained for any specific task. They have been previously pretrained for alternative tasks, and the models are fine-tuned via transfer learning. Transfer learning is a well established technique for improving the performance of machine learning models, and Raffel et al[19] establishes that transformers are no different. Fine tuning transformers trained on rich datasets for downstream tasks achieves state of the art results for dozens of language understanding tasks. Due to their high performance in related language modelling tasks, we utilise these four models and train them using transfer learning.

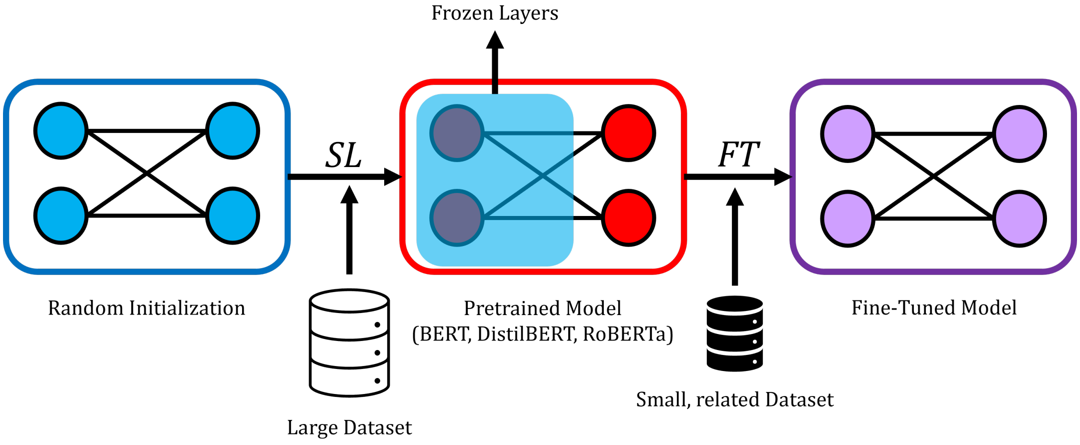


**Figure 4:** How Transfer Learning works.

For the selection of the best model, we utilise the following metrics[20]:

- Accuracy
- Kappa Coefficient, $\kappa$ [21]
- Precision
- Recall
- Macro $F_1$ Score[22]

We use these metrics, in addition to Early Stopping[23], to select the best model (and its best epoch) to use as the teacher model going forward.

#### 2.4.2 Pseudo-labelling and Mock Dataset

We use the selected teacher model's output logits as soft labels each sample given in the Feedback Prize Dataset (2.1.1) and The New York Times Dataset (2.1.2). Instead of relying on one-hot labels, the student model can take advantage of soft labels which contain inherent relationships between classes, helping us a produce a larger mock dataset composed of these pseudo-labelled datapoints. Following Xie et al.[12] original implementation, when calculating metrics to evaluate model performance on validation split of the datasets, we treat the correct label as the highest probability class predicted by the teacher model.

#### 2.4.3 Noise Addition

After this, we inject noise into the given samples. This is done using the `nlpaug` package [24] which we utilise, with the help of reference to Uribe et al [25], wherein `nlpaug` experiments were carried out to identify ideal forms of data augmentation.

| **Contextual Embedding** [26] | |
|---|---|
| Contextual Word Embedding [27] | Feeding surrounding words to BERT [17], DistilBERT [16], RoBERTa [18] or XLNet [28, 29] language models to find out the most suitable word(s) for substitution |
| Contextual Sentence Embedding [30] | Insert sentence according to XLNet [28, 29], GPT2 [31, 32] or DistilGPT2 [15] predictions |
| **Word-Based Augmentation [33]** | |
| Synonym Replacement [34] | Substitute similar word according to WordNet [35, 36] / PPDB [37, 38] synonym |
| **Sentence-Based Augmentation [39]** | |
| Abstractive Summarisation [40] | Summarize article by abstractive summarization method based on predictions from the T5-Base [19] model |

**Table 1:** Possible Augmentation Techniques to use in introducing noise to the dataset.

#### 2.4.4 Heuristic-based Elimination

Since many samples in the New York Times Dataset are picked on random, there are bound to be sample texts which do not fit well with the given task, for instance advertisement articles or interactive articles which are randomly pulled into the dataset. To remove these samples, since their raw logits are not trustworthy and hence the samples can corrupt the model, we utilise simple heuristics to remove the given articles.

We do this by applying a Softmax function on the raw logits to derive the final soft logits which are the actual probability distributions of the labels for each sample. We then compute the Standard Deviation of each individual probability distribution of the sample set. This gives rise to samples that the model is unsure about being given a lower score, whereas those that the model is able to identify with high probability is given a higher standard deviation score. It is as depicted below:

$$\begin{aligned} sl_{(j,k)} &= softmax(rl_{(j,k)}) \\ \mu_{(j,k)} &= \frac{1}{N}\sum sl_{(j,k,i)} \\ \sigma_{(j,k)} &= \sqrt{\frac{1}{N}\sum \left(sl_{(j,k,i)} - \mu_{(j,k)}\right)^2} \\ q_j &= \frac{1}{M}\sum \sigma_{(j,k)} \\ disqualified(j) &= q_j \text{in bottom 10\%} \end{aligned}$$

Wherein the following applies:

- $rl$ represents the raw logits
- $j$ is the index of sample paragraphs
- $k$ is the index of the sentence in sample $j$
- $M$ is the number of total sentences in the sample
- $sl$ represents the soft logits
- $i$ is the index of the logit of the sentence $k$ of given sample $j$
- $N$ is the total number of logits/classes
- $q_j$ is the qualifying factor of the sample $j$

We followed on with this with random subsampling in order to equalise the sizes of the Feedback Prize and New York Times datasets, which were disproportionate in volume and hence needed to be configured for this cause to a 1:1 ratio. This allowed a proper comparison of samples pre-labelled and samples that had been pseudo-labelled, which meant we could compare between them with relative ease when actually doing the testing.

This also opened up more samples from the Feedback Prize dataset, which had been prelabelled and hence would be excellent texts to compare against the student model which had been noised.

### 2.4.5 Student Model

The following is the algorithm we employed to iteratively train the Student Model. Conventionally, the teacher model is passed into memory alongside the student model and loss computation involves a forward pass through both models during runtime, at which point the logits of both models are compared. Due to computational limitations, we had the teacher model perform inference on all texts beforehand, then appending the logits of the teacher model as the labels of the texts, turning the training process into a slightly more complicated form of multi-label classification problem.

The loss function of the training is determined by the Kullback-Leibler divergence[41], a statistical measure of the difference between two probability distributions, in our case, the softmax of the output student and teacher logits. We additionally tune loss using temperature ($T$) and alpha ($\alpha$) parameters, and the final loss function is hence given by

$$\begin{aligned} P_{teacher} &= softmax(\frac{logits_{student}}{T}) \\ P_{student} &= softmax(\frac{logits_{teacher}}{T}) \\ \mathcal{L}(P_{teacher}, P_{student}) &= (P_{teacher} \cdot \log\left(\frac{P_{teacher}}{P_{student}}\right) \cdot (T^2)) \cdot \alpha \end{aligned}$$

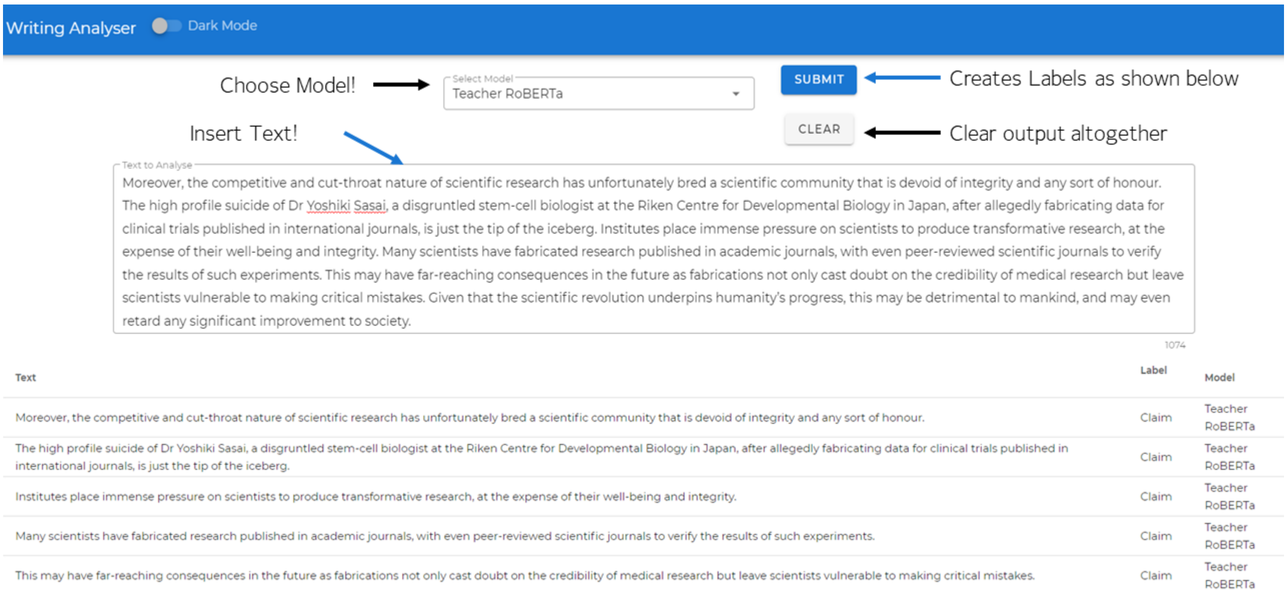
**Algorithm 1** The Student Model Architecture

```
procedure PSEUDOLABEL($\mathcal{M}_{teacher}, fp, nyt$)                ▷ pseudolabel with teacher logits
    $logits_{fp} = t(fp)$
    $logits_{nyt} = t(nyt)$
    while $size(nyt) \neq 0.1 \cdot size(fp)$ do
        $r \leftarrow random(nyt)$
        $nyt \leftarrow augment(r)$
        $logits_{nyt} \leftarrow \mathcal{M}_{teacher}(r)$
    end while
    $data_x \leftarrow scripts_{fp} + scripts_{nyt}$
    $data_y \leftarrow logits_{fp} + logits_{nyt}$
    return $data$
end procedure

procedure TRAINING($iter, fp, nyt, layerdrop$)
    $\mathcal{M}_{student,0} = model("roberta_{pretrained}")$          ▷ Pretrained on Feedback Prize only
    for i=1;i<iter;i++ do
        $\mathcal{M}_{student,i} = train($PSEUDOLABEL$(\mathcal{M}_{student,i-1}, fp, nyt), layerdrop)$
    end for
    return $\mathcal{M}_{student,best}$
end procedure
```

## 2.5 User Interface and Deployment

As part of this project, in order to visualise the models created, we utilise a web application developed in Vue.js (with TypeScript) and Flask (with Python). This web application is designed such that a simple paragraph input can trigger the model to segment it into sentences via NLTK's `PUNKT` tokenizer, then indicate each and every class that the sentences have been classified as.



Moreover, the competitive and cut-throat nature of scientific research has unfortunately bred a scientific community that is devoid of integrity and any sort of honour. The high profile suicide of Dr Yoshiki Sasai, a disgruntled stem-cell biologist at the Riken Centre for Developmental Biology in Japan, after allegedly fabricating data for clinical trials published in international journals, is just the tip of the iceberg. Institutes place immense pressure on scientists to produce transformative research, at the expense of their well-being and integrity. Many scientists have fabricated research published in academic journals, with even peer-reviewed scientific journals to verify the results of such experiments. This may have far-reaching consequences in the future as fabrications not only cast doubt on the credibility of medical research but leave scientists vulnerable to making critical mistakes. Given that the scientific revolution underpins humanity's progress, this may be detrimental to mankind, and may even retard any significant improvement to society.

| Text | Label | Model |
|---|---|---|
| Moreover, the competitive and cut-throat nature of scientific research has unfortunately bred a scientific community that is devoid of integrity and any sort of honour. | Claim | Teacher RoBERTa |
| The high profile suicide of Dr Yoshiki Sasai, a disgruntled stem-cell biologist at the Riken Centre for Developmental Biology in Japan, after allegedly fabricating data for clinical trials published in international journals, is just the tip of the iceberg. | Claim | Teacher RoBERTa |
| Institutes place immense pressure on scientists to produce transformative research, at the expense of their well-being and integrity. | Claim | Teacher RoBERTa |
| Many scientists have fabricated research published in academic journals, with even peer-reviewed scientific journals to verify the results of such experiments. | Claim | Teacher RoBERTa |
| This may have far-reaching consequences in the future as fabrications not only cast doubt on the credibility of medical research but leave scientists vulnerable to making critical mistakes. | Claim | Teacher RoBERTa |

**Figure 5:** How the User Interface Looks Like.

# 3 Results and Conclusion

## 3.1 Teacher Model and Selection

In this section, we note the results of the Teacher Model Selection. The evaluation metrics considered have been depicted as shown in **Figure 6**.

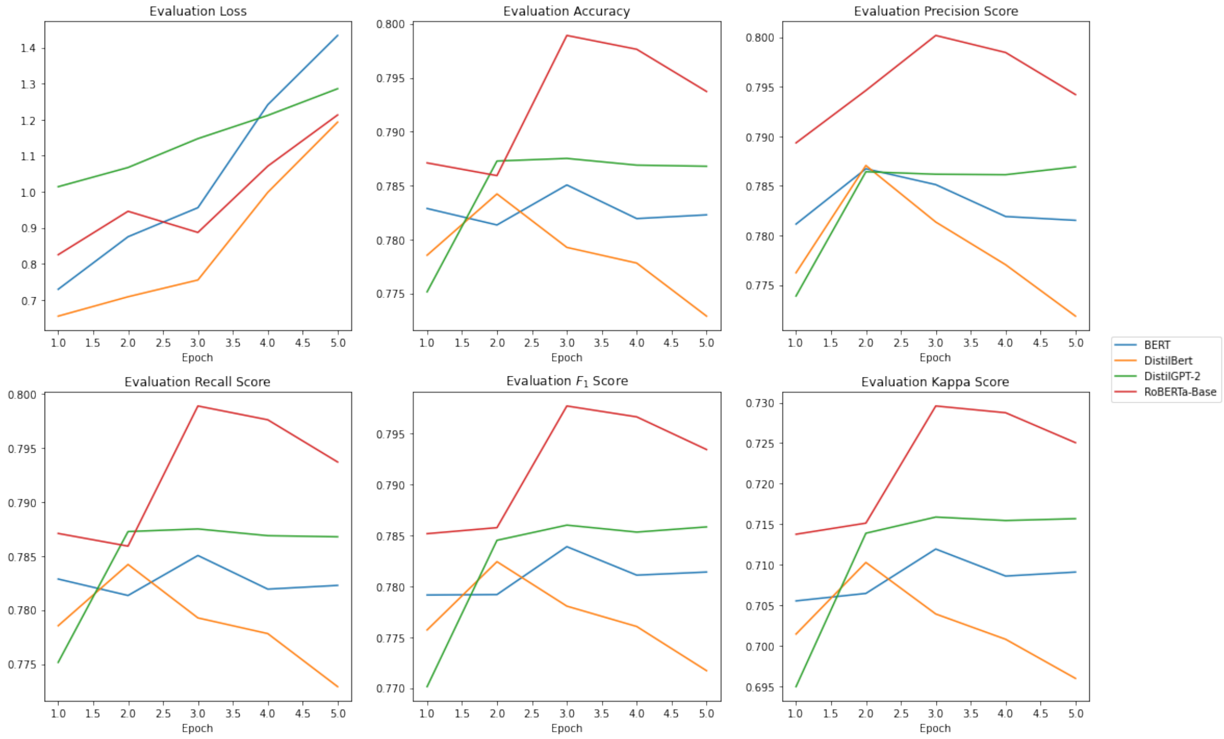


**Figure 6:** Evaluation Metric Values (per Epoch) of each of the Teacher Models.

Based on our assessments, we have concluded using the best few epochs, based on Validation-based Early Stopping, and hence we produce the data as shown in **Table 2**.

| Model | Epoch | $\mathcal{L}_{train}$ | $\mathcal{L}_{eval}$ | Accuracy | Precision | Recall | $F_1$ | $\kappa$ |
|---|---|---|---|---|---|---|---|---|
| BERT | 3 | 0.516 | 0.956 | 0.785 | 0.785 | 0.785 | 0.784 | 0.712 |
| DistilBERT | 2 | 0.469 | 0.708 | 0.784 | 0.787 | 0.784 | 0.782 | 0.710 |
| DistilGPT-2 | 5 | 0.837 | 1.286 | 0.787 | 0.787 | 0.787 | 0.786 | 0.716 |
| RoBERTa-Base | 3 | 0.619 | 0.887 | 0.799 | 0.800 | 0.799 | 0.798 | 0.730 |

**Table 2:** The results of all prospective Teacher Models.

We note that the **`RoBERTa-Base`** model at Epoch **3** is the best-performing Teacher, hence we decide to continue with this teacher model for our project.

## 3.2 Student Model

### 3.2.1 Identifying the best LayerDrop parameter

To identify the best LayerDrop parameter, we utilise only the New York Times dataset, which is comprised of pseudolabelled New York Times articles, to train 4 different models, each different based on the value $p^*$. The evaluation metrics considered have been depicted as shown in **Figure 7**. Based on the values in Epoch $5$, we have the following results as shown in **Table 3**.

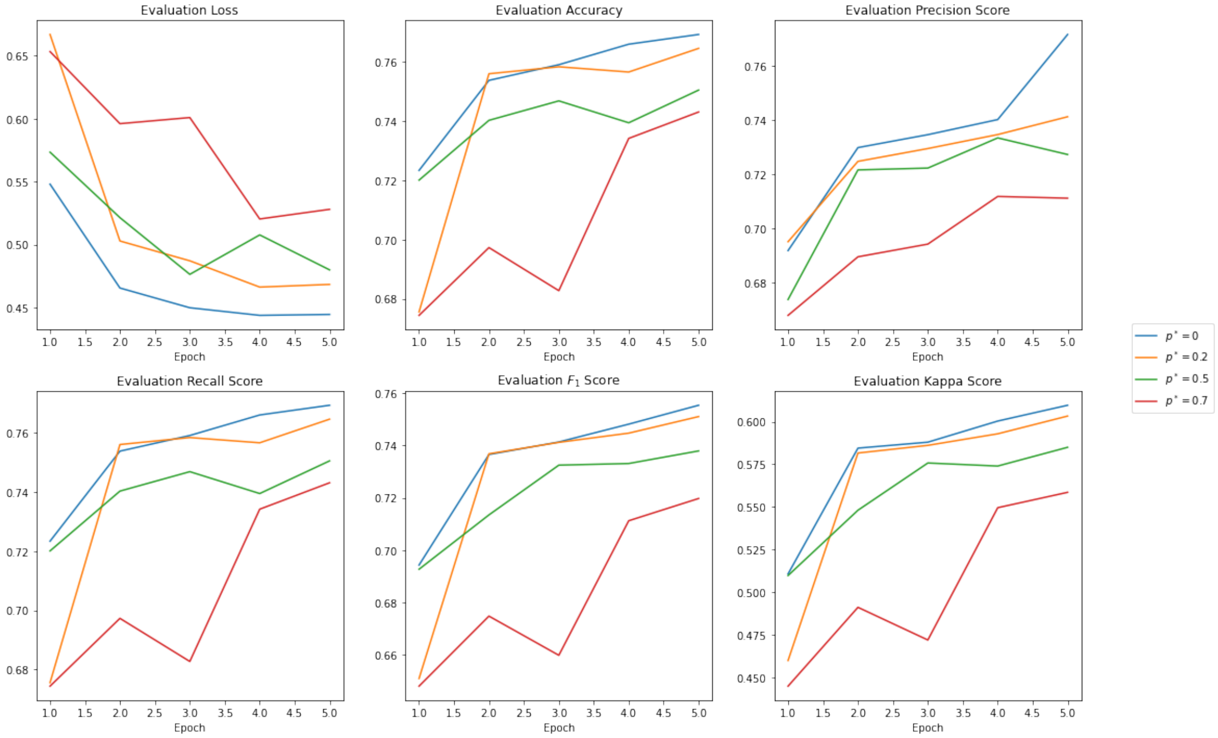


**Figure 7:** Evaluation Metric Values (per Epoch) of each of the Student Models, based on their $p^*$ values.

| $p^*$ | $\mathcal{L}_{train}$ | $\mathcal{L}_{eval}$ | Accuracy | Precision | Recall | $F_1$ | $\kappa$ |
|---|---|---|---|---|---|---|---|
| 0 | 0.421 | 0.445 | 0.769 | 0.772 | 0.769 | 0.755 | 0.610 |
| 0.2 | 0.467 | 0.468 | 0.765 | 0.741 | 0.765 | 0.751 | 0.603 |
| 0.3 | 0.373 | 0.495 | 0.780 | 0.768 | 0.780 | 0.770 | 0.616 |
| 0.5 | 0.526 | 0.480 | 0.751 | 0.727 | 0.751 | 0.738 | 0.585 |
| 0.7 | 0.592 | 0.528 | 0.743 | 0.711 | 0.743 | 0.720 | 0.559 |

**Table 3:** The results of Epoch $5$ based on the different $p^*$ values.

While it is clear that Epoch $5$ of the Noisy Student Model with no LayerDrop seems to work the best, we note that $\mathcal{L}_{train} < \mathcal{L}_{eval}$, which indicates a high amount of variance in the dataset, since it seems to overfit on the training set. Hence, we instead use Epoch **5** of the Noisy Student Model with LayerDrop Rate $\mathbf{p^* = 0.2}$. We also plan to experiment using Epoch **5** of the Noisy Student Model with LayerDrop Rate $\mathbf{p^* = 0.3}$.

### 3.2.2 Identifying the best Iteration

We now move on to identifying the best iteration, based on our trained student models. Due to time limitations, we only trained 3 iterations, and utilised LayerDrop values of $p^* \in \{0.2, 0.7\}$, as was concluded in the previous section. The evaluation metrics considered have been depicted as shown in **Figure 8**. This gave us a good understanding of the iteration process. Based on the values in Epoch $5$, we have the following results as shown in **Table 4**.

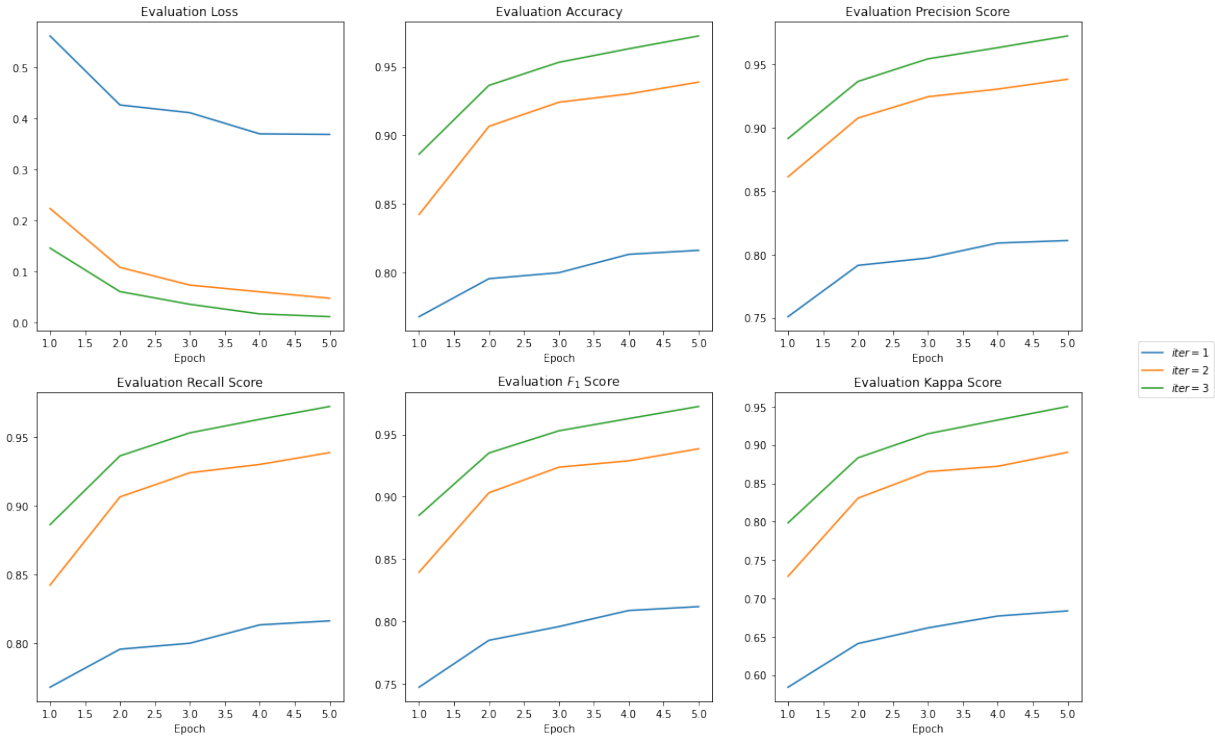


**Figure 8:** Evaluation Metric Values (per Epoch) of each of the Student Models, across different iterations ($p^* = 0.2$).

| Iteration | $\mathcal{L}_{train}$ | $\mathcal{L}_{eval}$ | Accuracy | Precision | Recall | $F_1$ | $\kappa$ |
|---|---|---|---|---|---|---|---|
| 1 | 0.446 | 0.369 | 0.816 | 0.811 | 0.816 | 0.812 | 0.684 |
| 2 | 0.159 | 0.047 | 0.939 | 0.938 | 0.939 | 0.938 | 0.890 |
| 3 | 0.145 | 0.011 | 0.972 | 0.973 | 0.972 | 0.972 | 0.950 |

**Table 4:** The results of Epoch $5$ across different iterations ($p^* = 0.2$).

It can be clearly seen that the performance of the model improves with each iteration of the teacher student model training, and hence the ideal iteration count, in this case, for Noisy Student Training is the **3rd Iteration**.

### 3.3 Pruning of Student Models

We used a student model with pruning value of $p^* = 0.7$ so as to showcase the results of pruning at the most extreme values. Following the "every other" pruning method suggested by Fan et al.[7], we simply drop every layer at depth $d$ where $d = k(\frac{1}{p})$ where $k$ is an arbitrary integer. The results are as described in 9.

As we can see, there is not a significant drop in performance for the pruned model in all metrics other than loss, and in fact the pruned model has a parameter count more than 26 times smaller than the normal model.

## 4 Conclusion

One thing to note in the above sections is that the metrics computed are based off the labels of the previous teacher model. In order to conclude that the model has actually improved in performance, we compare the model's performance to our labelled Feedback Prize dataset(2.1.1) and a subset of manually labelled New York Times Dataset(2.1.2) articles, as depicted in **Table 5**.

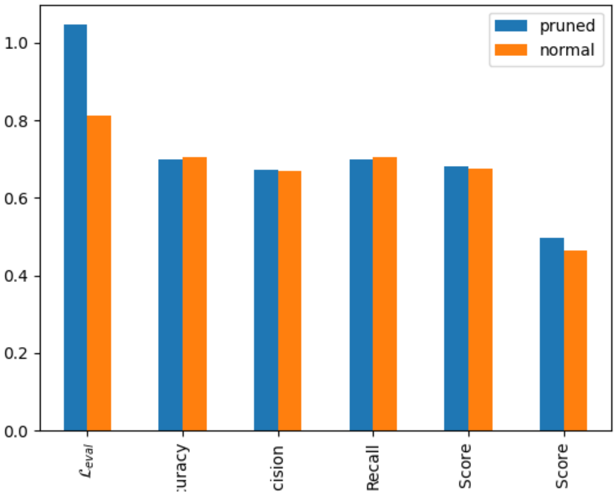


**Figure 9:** How Pruning Helps.

| Metric | $\mathcal{M}_{teacher}$ | $\mathcal{M}_{student,0}$ | $\mathcal{M}_{student,1}$ | $\mathcal{M}_{student,2}$ |
|---|---|---|---|---|
| Accuracy | 0.860 | 0.740 | 0.730 | 0.726 |
| Mean IoU | 0.695 | 0.491 | 0.480 | 0.482 |
| Weighted Precision Score | 0.863 | 0.753 | 0.744 | 0.746 |
| Weighted Precision Score | 0.863 | 0.753 | 0.744 | 0.746 |
| $\kappa$ Score | 0.811 | 0.641 | 0.627 | 0.622 |
| Weighted $F_1$ Score | 0.858 | 0.732 | 0.721 | 0.718 |

**Table 5:** Comparison of multiple metrics between the models ($p^* = 0.2$) on the Feedback Prize dataset.

This result is not ideal, as we can see that for every score, the value has dropped significantly in the cases of any $M_{student,i}$. This is unideal, and although the comparison over the teacher model bore positive results for these models, we must unfortunately conclude that Noisy Student Training does not perform well, nor improve the scores of the models.

However, after running and evaluating a separate model run for 15 epochs, we reached much higher values for these metrics, which suggests that our models did not train for enough epochs, and are possibly undertrained, thus further improvement can be made for the model to ensure that the accuracy improves.

# 5 Discussion and Future Work

## 5.1 Novelty of Methods used

In our project, we utilised Noisy Student Training, an algorithm traditionally used in Image Processing and Computer Vision tasks, on a Natural Language Processing (NLP) task, which made this task all the more novel. Whilst this sort of training has been performed before on NLP tasks, it has been relatively scarce, and our work, albeit showing negative results, is a testament to how this technique is still relatively usable in other fields other than Computer Vision, although it may not bear the monumental results achieved in Xie et al [12].

## 5.2 Technical and Conceptual Limitations

One of the main limitations in this project was the limited amount of processing power and time, since only one of possessed GPUs. Even if we had used Google Colaboratory, the training time, in addition to preprocessing and acquisition, would not be able to sustain the duration as configured in the free version. Due to monetary limitations, we did not proceed to purchase licenses for AWS SageMaker of Google Colaboratory Pro, which would have definitely helped in speeding up this project.

In terms of conceptual limitations, this was our first Natural Language Processing (NLP) project, and even though one of us had delved into Topic Modelling before, this was a much more complex project for us to tackle, especially given the lack of topical info taught in the module with regards to NLP. This indeed posed a challenge for us, since the documentation for HuggingFace was not that great, and understanding Transformers was a bit difficult with all of this novel research, of sorts.

## 5.3 Future Work

Our current model does not use the influence of other textual elements around the sentence to change the decision of the model, which can be solved if we use Sequential Sentence Classification (SSC)[42], which allows us make sentence classifications based on the surrounding local context, which would likely improve our classification accuracies as many labels are dependent on the content of the previous and next sentences.

Another interesting exploration would be the usage of alternative sentence augmentation techniques. In our project, we decided to dial down to abstractive summarisation methods over a subset of our data due to time considerations, but realistically more noise can be added to the dataset to ensure greater robustness against noisy samples.

Additionally, we could explore the possibly of text style transfer [43, 44], training a model to convert from informal to formal writing or vice versa, as the styles of the two datasets that we used varied vastly due to the nature of how the data was obtained, and this would have been an interesting avenue of exploration.

# 6 Reflection

## 6.1 Wei Pyn

Through the course of the project I learnt a lot more about transformers and implementation of large libraries, having to subclass and look through the source code of the HuggingFace library to implement the Noisy Student architecture. I believe that if we had more time, we could possibly perfect the process and more thoroughly explore the possibilities. Perhaps we did not properly train the teacher model for enough epochs, and that lead to the slow degradation of student model quality with each iteration, although that is left to future work.

## 6.2 Prannaya

In this project, I worked on many things that I had never even learnt before, such as transformers and novel techniques such as noisy student training and knowledge distillation. These have helped me gain a greater appreciation over the way that AI infrastructure is usually handled, and I am more motivated to pursue AI research in the future. While the task itself was relatively challenging, I think with the time that we have gotten, we achieved the best that we could. With more time, we could have certainly done a brilliant project, but even as it stands, I think we have achieved what we aimed for, and that, I believe, is what makes our project good. As for what I learnt, I think we both have worked through a lot of struggles and for me, that was the User Interface, which was difficult to format given my limited understanding of Flask and only recent introduction to Vue as part of AppVenture. However, I think this has given me a great purview into how many companies automate AI-powered websites.

# 7 Work Distribution Matrix

| | Prannaya | Wei Pyn |
|---|---|---|
| Project Brainstorm and Planning | ✓ | ✓ |
| Dataset Sourcing and Acquisition | ✓ | |
| Data Cleaning and Preprocessing | ✓ | ✓ |
| Training Teacher Models | | ✓ |
| Pseudo-Labelling and Noise Injection | ✓ | ✓ |
| Training Student Model | | ✓ |
| User Interface (UI) | ✓ | |
| Presentation Slides | ✓ | ✓ |
| Report Writing | ✓ | ✓ |
| Video Presentation | ✓ | ✓ |

# 8 References

[1] Georgia State University and Learning Agency Lab. *Feedback Prize 2021 Dataset*. URL: https://www.kaggle.com/competitions/feedback-prize-2021/data/.

[2] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python*. O'Reilly, 2009. URL: https://www.nltk.org/book/.

[3] Tibor Kiss and Jan Strunk. "Unsupervised Multilingual Sentence Boundary Detection". In: *Computational Linguistics* 32.4 (Dec. 2006), pp. 485–525. ISSN: 0891-2017. DOI: 10.1162/coli.2006.32.4.485. eprint: https://direct.mit.edu/coli/article-pdf/32/4/485/1798345/coli.2006.32.4.485.pdf. URL: https://doi.org/10.1162/coli.2006.32.4.485.

[4] Ashish Vaswani et al. *Attention Is All You Need*. 2017. DOI: 10.48550/ARXIV.1706.03762. URL: https://arxiv.org/abs/1706.03762.

[5] Gao Huang et al. *Deep Networks with Stochastic Depth*. 2016. DOI: 10.48550/ARXIV.1603.09382. URL: https://arxiv.org/abs/1603.09382.

[6] Kaiming He et al. *Deep Residual Learning for Image Recognition*. 2015. DOI: 10.48550/ARXIV.1512.03385. URL: https://arxiv.org/abs/1512.03385.

[7] Angela Fan, Edouard Grave, and Armand Joulin. *Reducing Transformer Depth on Demand with Structured Dropout*. 2019. DOI: 10.48550/ARXIV.1909.11556. URL: https://arxiv.org/abs/1909.11556.

[8] Qiang Wang, Tong Xiao, and Jingbo Zhu. *Training Flexible Depth Model by Multi-Task Learning for Neural Machine Translation*. 2020. DOI: 10.48550/ARXIV.2010.08265. URL: https://arxiv.org/abs/2010.08265.

[9] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. *Distilling the Knowledge in a Neural Network*. 2015. DOI: 10.48550/ARXIV.1503.02531. URL: https://arxiv.org/abs/1503.02531.

[10] Bonggun Shin, Hao Yang, and Jinho D. Choi. "The Pupil Has Become the Master: Teacher-Student Model-Based Word Embedding Distillation with Ensemble Learning". In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, July 2019, pp. 3439–3445. DOI: 10.24963/ijcai.2019/477. URL: https://doi.org/10.24963/ijcai.2019/477.

[11] Yidi Jiang et al. *Knowledge Distillation from BERT Transformer to Speech Transformer for Intent Classification*. 2021. DOI: 10.48550/ARXIV.2108.02598. URL: https://arxiv.org/abs/2108.02598.

[12] Qizhe Xie et al. *Self-training with Noisy Student improves ImageNet classification*. 2019. DOI: 10.48550/ARXIV.1911.04252. URL: https://arxiv.org/abs/1911.04252.

[13] Jia Deng et al. "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[14] Mingxing Tan and Quoc V. Le. "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks". In: (2019). DOI: 10.48550/ARXIV.1905.11946. URL: https://arxiv.org/abs/1905.11946.

[15] *DistilGPT2*. Hugging Face, 2019. URL: https://huggingface.co/distilgpt2.

[16] Victor Sanh et al. *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter*. 2019. DOI: 10.48550/ARXIV.1910.01108. URL: https://arxiv.org/abs/1910.01108.

[17] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2018. DOI: 10.48550/ARXIV.1810.04805. URL: https://arxiv.org/abs/1810.04805.

[18] Yinhan Liu et al. *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. 2019. DOI: 10.48550/ARXIV.1907.11692. URL: https://arxiv.org/abs/1907.11692.

[19] Colin Raffel et al. *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*. 2019. DOI: 10.48550/ARXIV.1910.10683. URL: https://arxiv.org/abs/1910.10683.

[20] Margherita Grandini, Enrico Bagli, and Giorgio Visani. *Metrics for Multi-Class Classification: an Overview.* 2020. DOI: 10.48550/ARXIV.2008.05756. URL: https://arxiv.org/abs/2008.05756.

[21] Jacob Cohen. "A Coefficient of Agreement for Nominal Scales". In: *Educational and Psychological Measurement* 20.1 (1960), pp. 37–46. DOI: 10.1177/001316446002000104. eprint: https://doi.org/10.1177/001316446002000104. URL: https://doi.org/10.1177/001316446002000104.

[22] Juri Opitz and Sebastian Burst. *Macro F1 and Macro F1.* 2019. DOI: 10.48550/ARXIV.1911.03347. URL: https://arxiv.org/abs/1911.03347.

[23] Lutz Prechelt. "Early Stopping — But When?" In: *Neural Networks: Tricks of the Trade: Second Edition.* Ed. by Grégoire Montavon, Geneviève B. Orr, and Klaus-Robert Müller. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 53–67. ISBN: 978-3-642-35289-8. DOI: 10.1007/978-3-642-35289-8_5. URL: https://doi.org/10.1007/978-3-642-35289-8_5.

[24] Edward Ma. *NLP Augmentation.* https://github.com/makcedward/nlpaug. 2019.

[25] Juan Uribe and Mandy Korpusik. *Mapping of exercise logs to a database using Neural Networks and data augmentation techniques.* https://github.com/juanuribe28/research-f2020. 2020-2021.

[26] Varun Kumar, Ashutosh Choudhary, and Eunah Cho. *Data Augmentation using Pre-trained Transformer Models.* 2020. DOI: 10.48550/ARXIV.2003.02245. URL: https://arxiv.org/abs/2003.02245.

[27] Sosuke Kobayashi. *Contextual Augmentation: Data Augmentation by Words with Paradigmatic Relations.* 2018. DOI: 10.48550/ARXIV.1805.06201. URL: https://arxiv.org/abs/1805.06201.

[28] Zhilin Yang et al. *XLNet: Generalized Autoregressive Pretraining for Language Understanding.* 2019. DOI: 10.48550/ARXIV.1906.08237. URL: https://arxiv.org/abs/1906.08237.

[29] Aman Rusia. *XLNet-Gen: XLNet for Generating Language.* https://github.com/rusiaaman/XLNet-gen. 2019.

[30] Robin Jia and Percy Liang. *Adversarial Examples for Evaluating Reading Comprehension Systems.* 2017. DOI: 10.48550/ARXIV.1707.07328. URL: https://arxiv.org/abs/1707.07328.

[31] Alec Radford et al. "Language Models are Unsupervised Multitask Learners". In: (2019).

[32] Ranto Sawai, Incheon Paik, and Ayato Kuwana. "Sentence Augmentation for Language Translation Using GPT-2". In: *Electronics* 10.24 (2021), p. 3082.

[33] Mao Ye, Chengyue Gong, and Qiang Liu. *SAFER: A Structure-free Approach for Certified Robustness to Adversarial Word Substitutions.* 2020. DOI: 10.48550/ARXIV.2005.14424. URL: https://arxiv.org/abs/2005.14424.

[34] Tong Niu and Mohit Bansal. *Adversarial Over-Sensitivity and Over-Stability Strategies for Dialogue Models.* 2018. DOI: 10.48550/ARXIV.1809.02079. URL: https://arxiv.org/abs/1809.02079.

[35] George A. Miller. "WordNet: A Lexical Database for English". In: *Commun. ACM* 38.11 (Nov. 1995), pp. 39–41. ISSN: 0001-0782. DOI: 10.1145/219717.219748. URL: https://doi.org/10.1145/219717.219748.

[36] Christiane Fellbaum, ed. *WordNet: An Electronic Lexical Database.* Language, Speech, and Communication. Cambridge, MA: MIT Press, 1998. ISBN: 978-0-262-06197-1.

[37] Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. "PPDB: The Paraphrase Database". In: *NAACL.* 2013.

[38] John Wieting et al. "From Paraphrase Database to Compositional Paraphrase Model and Back". In: (2015). DOI: 10.48550/ARXIV.1506.03487. URL: https://arxiv.org/abs/1506.03487.

[39] Yi Zhang et al. "Sequence-to-sequence pre-training with data augmentation for sentence rewriting". In: *arXiv preprint arXiv:1909.06002* (2019).

[40] Diyah Puspitaningrum. *A Survey of Recent Abstract Summarization Techniques.* 2021. DOI: 10.48550/ARXIV.2105.00824. URL: https://arxiv.org/abs/2105.00824.

[41] Solomon Kullback. *Information theory and statistics.* Courier Corporation, 1997.

[42] Arman Cohan et al. “Pretrained Language Models for Sequential Sentence Classification”. 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)*. Association for Computational Linguistics, 2019. DOI: 10.18653/v1/d19-1383. URL: https://doi.org/10.18653%2Fv1%2Fd19-1383.

[43] Martina Toshevska and Sonja Gievska. “A Review of Text Style Transfer using Deep Learning”. In: *IEEE Transactions on Artificial Intelligence* (2021), pp. 1–1. DOI: 10.1109/tai.2021.3115992. URL: https://doi.org/10.1109%2Ftai.2021.3115992.

[44] Di Jin et al. *Deep Learning for Text Style Transfer: A Survey*. 2020. DOI: 10.48550/ARXIV.2011.00416. URL: https://arxiv.org/abs/2011.00416.