Machine Learning Engineer Nanodegree Capstone Report

Detecting contradiction and entailment in multilingual text using TPUs

Matthew Soh

3 Aug 2021

1. Domain Background

Natural language processing (NLP) is the intersection of computer science, linguistics and machine learning. The field focuses on communication between computers and humans in natural language and NLP is all about making computers understand and generate human language.

In the past few years, there have been great advancements in the research of NLP models, particularly transformers, to tackle language tasks like question answering, text extraction, sentence generation and many other complex tasks. In this Kaggle Competition, the task is to build an NLP model that can determine the relationships between sentences, which could have profound implications for fact-checking, identifying fake news, analyzing text and much more. The nature of the task is called Natural language inference (NLI), where the model must determine whether a "hypothesis" is true (entailment), false (contradiction), or undetermined (neutral) given a "premise". The current state of the art model for NLI tasks is the RoBERTa model (Liu et al., 2019) that was able to achieve a categorization accuracy of 90.8 on the Multi-Genre Natural Language Inference (MultiNLI) corpus.

I am particularly interested in this domain of Artificial Intelligence, as a model that can successfully determine the relationship between sentences would improve the way we conduct focus group discussions for my company's research projects. Transcripts can be more thoroughly analyzed and we would be able to leverage our NLP models to generate deeper insights.

2. Problem Statement

Taken from Kaggle Competition Page:

Our brains process the meaning of a sentence like this rather quickly.
We're able to surmise:

- Some things to be true: "You can find the right answer through the process of elimination."
- Others that may have truth: "Ideas that are improbable are not impossible!"
- And some claims are clearly contradictory: "Things that you have ruled out as impossible are where the truth lies.

If you have two sentences, there are three ways they could be related: one could entail the other, one could contradict the other, or they could be unrelated. Natural Language Inferencing (NLI) is a popular NLP problem that involves determining how pairs of sentences (consisting of a premise and a hypothesis) are related.

Your task is to create an NLI model that assigns labels of 0, 1, or 2 (corresponding to entailment, neutral, and contradiction) to pairs of premises and hypotheses. To make things more interesting, the train and test set include text in fifteen different languages!

3. Datasets and Inputs

3.1 Dataset Description

The dataset provided was taken from the XNLI data, which is a subset of a few thousand examples from MNLI which has been translated into 15 different languages. As with MNLI, the goal is to predict textual entailment (does sentence A imply/contradict/neither sentence B) and is a classification task (given two sentences, predict one of three labels).

The fifteen different languages, including: Arabic, Bulgarian, Chinese, German, Greek, English, Spanish, French, Hindi, Russian, Swahili, Thai, Turkish, Urdu, and Vietnamese.

3.2 Dataset Example

Taken from Kaggle Competition Page on Data Description:

Let's take a look at an example of each of these cases for the following premise:

He came, he opened the door and I remember looking back and seeing the expression on his face, and I could tell that he was disappointed.

Hypothesis 1:

Just by the look on his face when he came through the door I just knew that he was let down.

We know that this is true based on the information in the premise. So, this pair is related by **entailment**.

Hypothesis 2:
He was trying not to make us feel guilty but we knew we had caused him trouble.

This very well might be true, but we can't conclude this based on the information in the premise. So, this relationship is **neutral**.

Hypothesis 3:
He was so excited and bursting with joy that he practically knocked the door off it's frame.
We know this isn't true, because it is the complete opposite of what the premise says. So, this pair is related by **contradiction**.

3.3 Training and test set

A training set and test set was provided to train the language model.

The training set contains 12,120 entries, with the columns (i) id, (ii) premise, (iii) hypothesis, (iv) lang_abv, (v) language and (vi) label. There are 8,209 unique premises, 12,119 unique hypotheses and 3 unique labels.

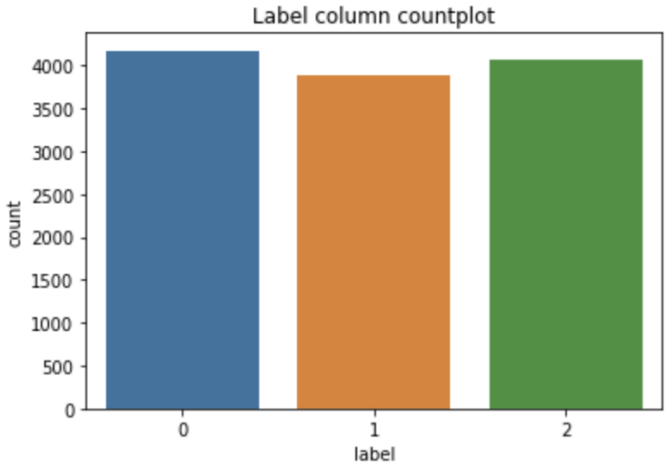
The test set contains 5,195 entries with 4,336 unique premises and 5,195 unique hypotheses. The test set will be used for the competition submission.

4 Evaluation Metrics

The evaluation metric used for this competition is a simple categorization accuracy, which can be calculated with the following equation:

$$Accuracy = \frac{True\ Positive + True\ Negative}{True\ Positive + True\ Negative + False\ Positive + False\ Negative}$$

5 Data Exploration



Class distribution

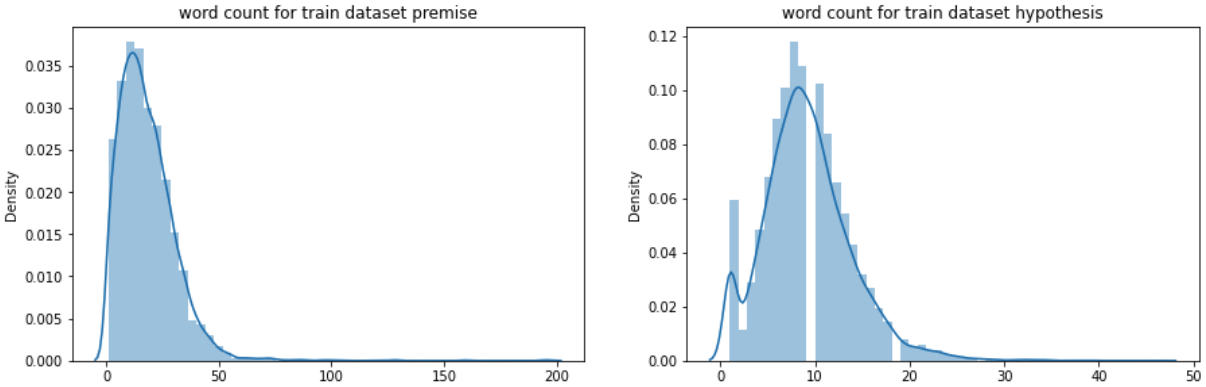
Out of the 12,120 entries in the training set, we observe an equal split in 3 classes (0 for entailment, 1 for neutral, 2 for contradiction).

This would mean that we would not be required to apply techniques to balance the datasets.

Language distribution

That being said, there is an uneven representation in the languages, with English being overrepresented in the dataset. This is expected, as it was explained that the dataset was originally an English dataset that was translated into multiple languages.

Strategies to overcome the uneven representation of languages will be elaborated in the later section.

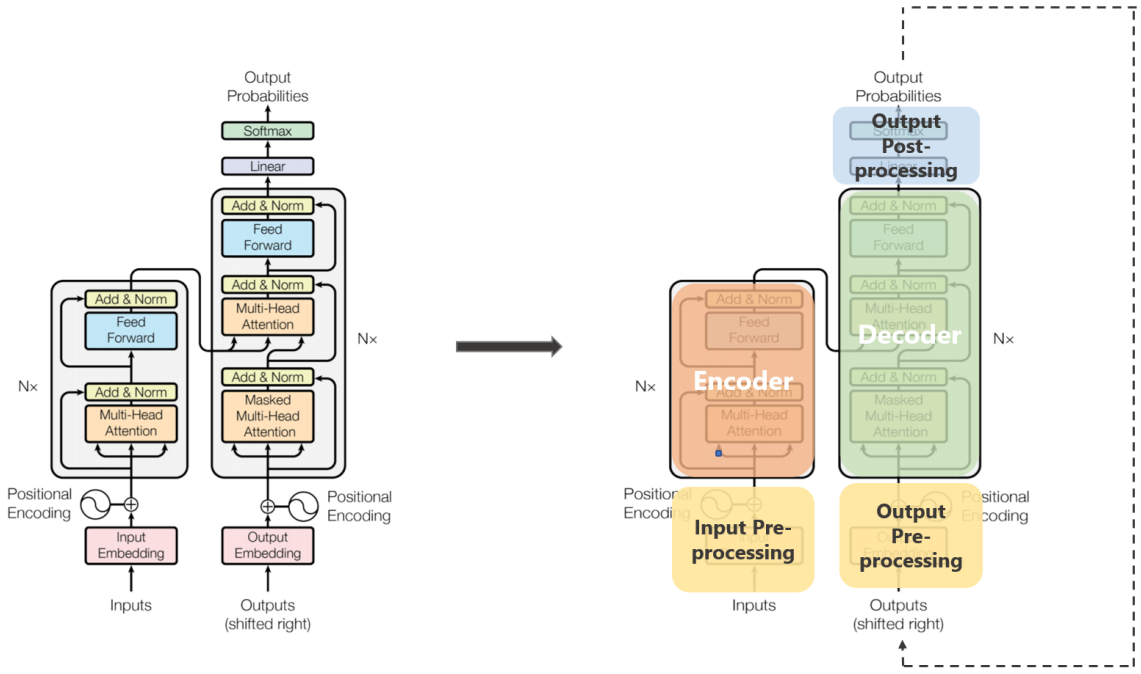


Word count

A histogram of the distribution of word count is also plotted, to understand the nature of the sentences in the dataset. The premise seems to be longer than the hypothesis, with a median of 26 words, while the hypothesis only has a median of 10 words. This is a useful plot that would inform how we design our model hyperparameters later one.

6 Algorithms and Techniques

While the traditional way of approaching such an NLP task would be to convert these sentences into word vectors and pass them through a RNN or LSTM network, we are able to achieve better performance by making use of Transformers. The Transformer in NLP is a novel architecture that aims to solve sequence-to-sequence tasks while handling long-range dependencies with ease.



Transformer Architecture based on research paper "Attention is All You Need" (left) and an abstracted version (right)
Credit: https://towardsdatascience.com/transformers-89034557de14

The BERT (Bidirectional Encoder Representations from Transformers) architecture was proposed in a paper published by researchers at Google AI Language. BERT's applies the bidirectional training of the Transformer architecture to language modelling. The paper's results show that a language model which is bidirectionally trained can have a deeper sense of language context and flow than single-direction language models. In the paper, the researchers detail a novel technique named Masked LM (MLM) which allows bidirectional training in models in which it was previously impossible.

As the BERT architecture is designed with only the encoder portion of the transformer, it is used for text classification tasks such as sentiment analysis and natural language inference but adding an additional layer on top of the BERT architecture.

For our task at hand, we have a mere 12,000 training samples. As BERT is a big neural network with a large number of parameters, it would be advisable to perform transfer learning and utilise a pre-trained BERT model and fine tune the final layers of the model with our training sample.

We will leverage the huggingface library, which contains a repository of pre-trained transformer models to tackle this task.

7 Benchmark

7.1 Benchmark Procedure

We hope to explore how the Transformer architecture performs on our textual entailment task. To establish a baseline, we will use the "bert-base-multilingual-cased" model available in huggingface. This is a base BERT model trained on the top 104 languages.

We performed a 90-10 split of the training set to form a training and validation set, which will be the same training and validation set used for the rest of our experiments.

We used the standard training parameters as advised in the original BERT paper to fine tune the classification layer of the BERT model, and used the max word length based on the median word count in our dataset (i.e. sentences longer than the max

word length will be truncated while sentences less than the max word length will be padded will zeros).

Model Hyperparameter	**Value**
Max sentence length	30
Batch size	32
Optimizer	AdamW
Learning rate	1e-5
Epsilon	1e-8

7.2 Benchmark Results

We were able to achieve an accuracy of 55% on the validation set after 3 epochs of training. We aim to build a model that outperforms this benchmark.

8 Methodology

8.1 Data Pre-processing

Our dataset consists of two columns of text data, the “premise” and “hypothesis”. As we are using the BERT architecture, we will have to encode our data in the same format that text was trained on BERT. We perform the following steps of data pre-processing:

1. Each sentence will be separated into individual words, which are then encoded into integers that correspond to their respective index in the vocabulary used to train the BERT model.
2. Add special tokens at the start and end of sentence, to distinctly separate sentences.
    a. At the end of each sentence, we add a “[SEP]” token.
    b. Since we are performing a classification task, we also have to add a “[CLS]” token at the start of the sentence.
3. We also have to pad sentences that are too short and truncate sentences that are too long.
    a. We pad the end of short sentences with a specific padding token till they are a certain “max_length”.
    b. For sentences that are too long, we truncate them when they have reached the “max_length”.

4. We prepare an attention mask, a list of 0s and 1s which helps to explicitly differentiate real and padded tokens. This attention mask informs the BERT model which word to focus on when decoding.

As we are dealing with a multilingual text dataset, checking for spelling and grammar mistakes was not possible. Thus we did not revise the content of the textual data at all, we merely prepared them to be ingested by the BERT model.

8.2 Experiment 1: Modelling

While the BERT architecture has provided a good benchmark for us to work on, it would be useful to also explore the different transformer models designed for such classification tasks. Due to the limited time given for this project, 3 other models will be explored, DistilBERT, RoBERTa (base), RoBERTa (large). A table below consolidates the specifications of the model.

	Multilingual BERT (cased) (benchmark)	Multilingual DistilBERT (cased)	XLM RoBERTa (base)	XLM RoBERTa (large)
No. of parameters	179 mil	134 mil	270 mil	550 mil
Architecture	12-layer, 768-hidden, 12-heads	6-layer, 768-hidden, 12-heads	12-layers, 768-hidden-state, 3072 feed-forward hidden-state, 8-heads	24-layers, 1024-hidden-state, 4096 feed-forward hidden-state, 16-heads
Data	Wikipedia in 104 languages	Wikipedia in 104 languages	Trained on 2.5 TB of newly created clean CommonCrawl data in 100 languages	Trained on 2.5 TB of newly created clean CommonCrawl data in 100 languages

As seen clearly from the specifications of the various models, not all models are built the same, neither are they designed for the same intent. While the aim of XLM RoBERTa is clearly to optimise for performance, DistilBERT aims to improve on the inference speed while maintaining some level of performance.

DistilBERT vs BERT

DistilBERT learns a distilled version of BERT, and is able to retain an approximate 97% of the performance while using significantly lesser parameters. In its architecture, it retains only half of the layers compared to BERT and uses a

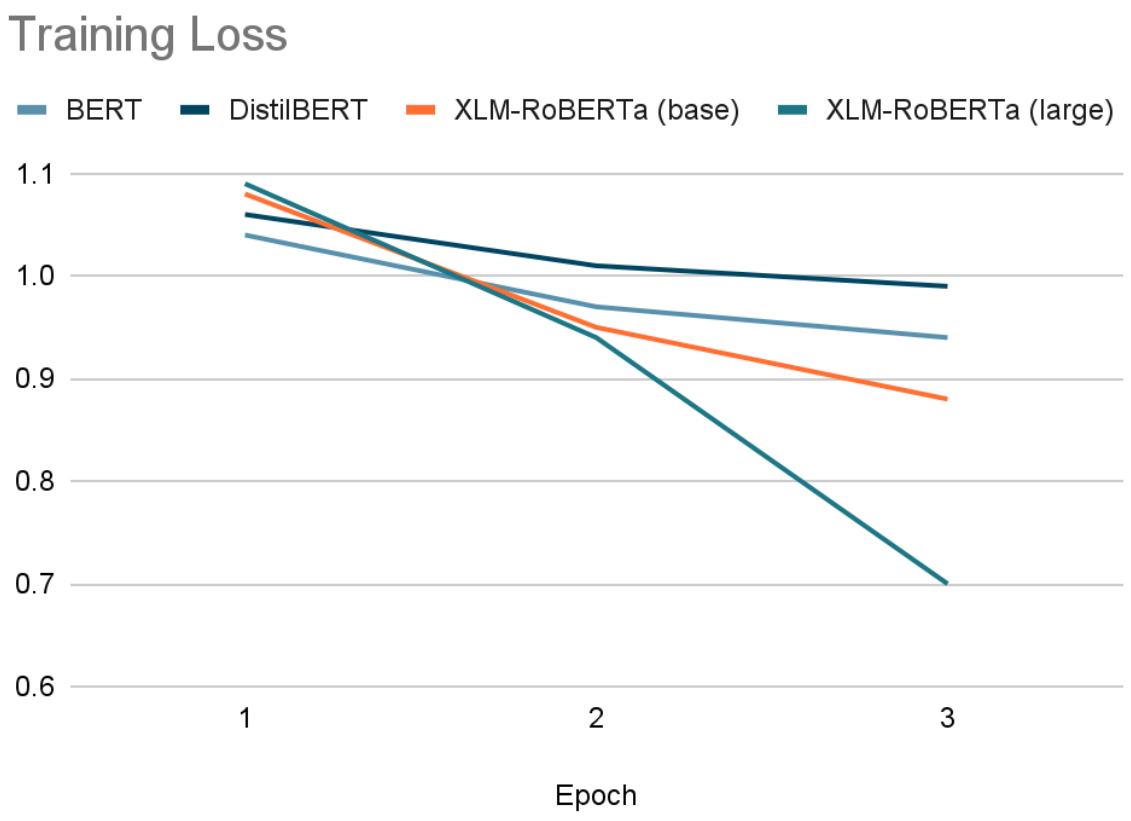
technique called distillation to approximate a larger network with a small network. Details of distillation are outside the scope of this report, but DistilBERT was included in our analysis out of curiosity -- to understand the effect of shrinking the network on the performance of the model, to verify that the network is able to retain 97% of its performance and to understand the trade offs.

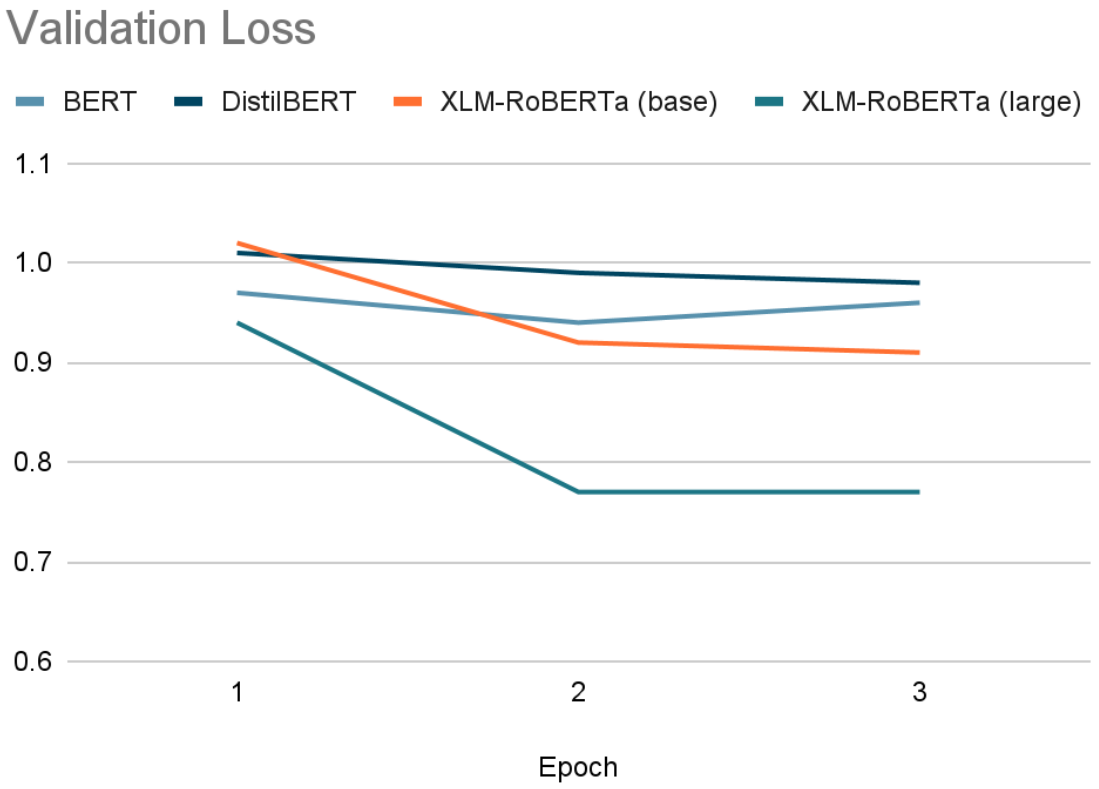
RoBERTa vs BERT

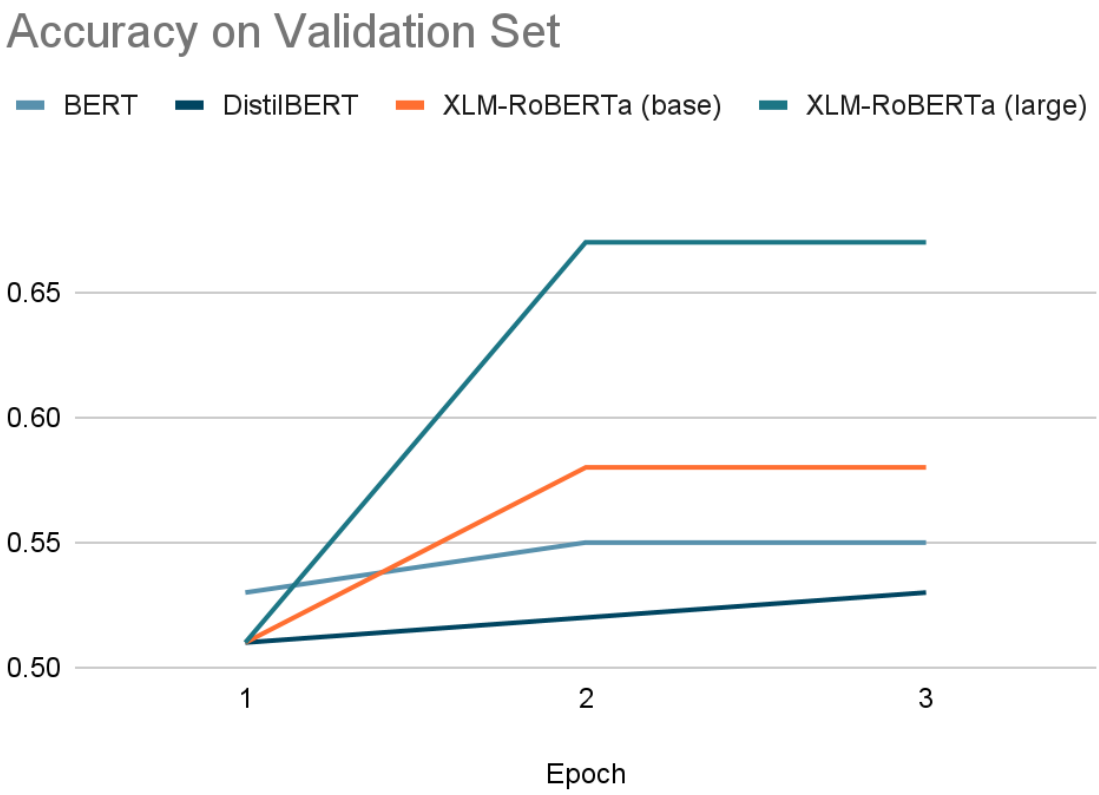
RoBERTa is a newer iteration of BERT that attempts to increase the performance by improving the training methodology and expanding the training dataset. In the paper, it is stated RoBERTa uses about 10 times more text data than BERT and it also attempts to improve the training process by removing the Next Sentence Prediction task (one of the tasks used for training a BERT model) and introduces a dynamic masking task so that the masked token changes during the training epochs. The base and large RoBERTa model only differ in terms of the number of layers and parameters, but are trained in the same manner, on the same dataset. RoBERTa was considered as a possible candidate for our experiment due to the sheer size of the model and the amount of text data that it has been pre-trained on.

Experiment Setup

In our first experiment, we used the same training hyperparameters to train the 4 different models and report the training and validation loss, as well as the accuracy on the validation set.



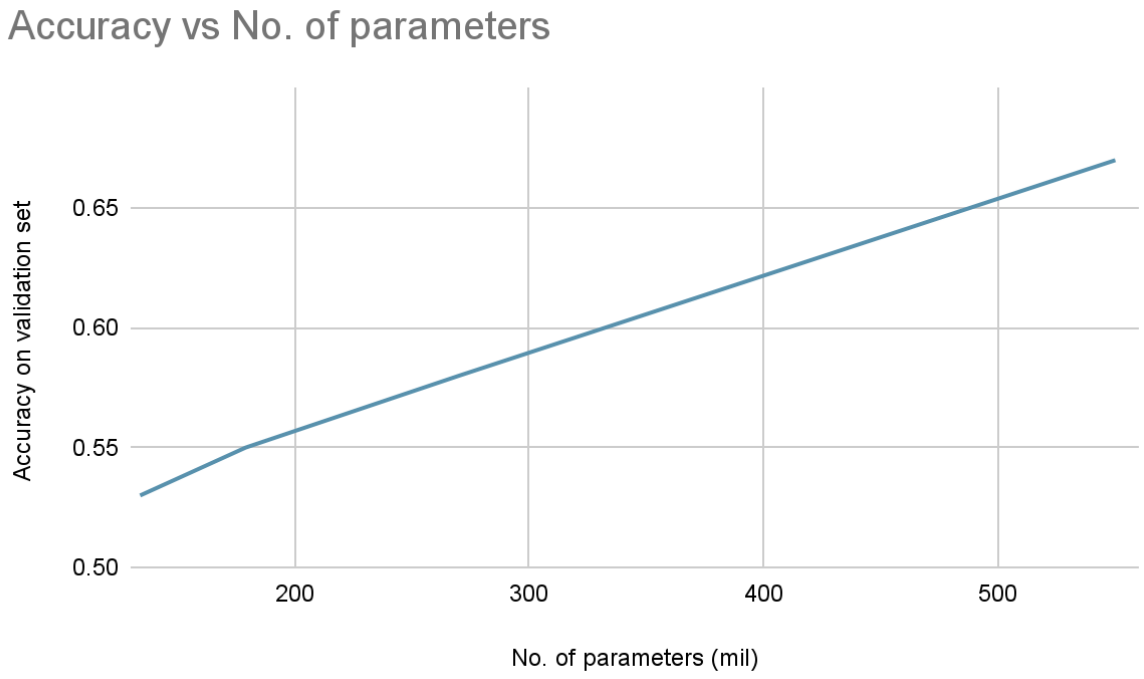




Observations

As expected, we see models with larger parameters performing better on the validation set, with XLM-RoBERTa (large) able to attain a 67% accuracy, 12% better than our baseline.

It may also be worth looking at a plot of the performance vs parameters, to see that there seems to be a linear relationship between the number of parameters and accuracy. Based on the graph plotted, for every additional 100 million parameters trained, it would give a 3.33% increase in the validation accuracy.

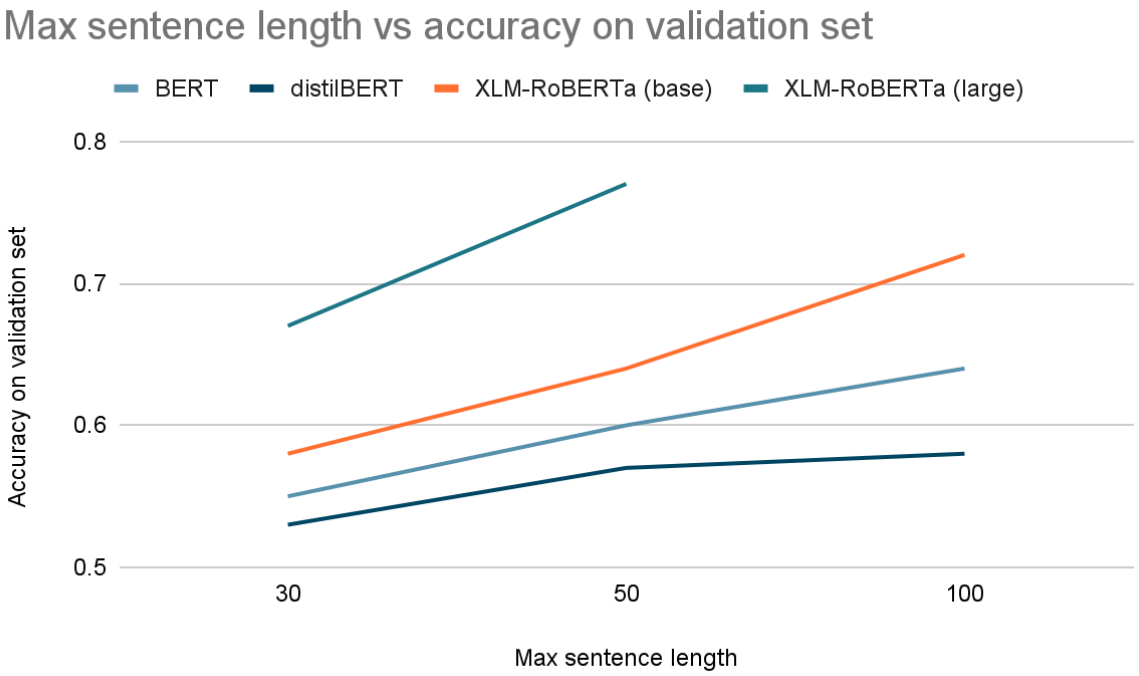


8.3 Experiment 2: Changing the max length of sentences

From the graph of the validation and training loss of the models, it could be observed that the validation loss did not drop as consistently as the training loss, a clear sign of overfitting. In order to counter this overfitting, a simple method of changing the max length of the sentences was used to supply more data into the model.

During the data pre-processing step, a 30 word cap was initially set on sentences, and truncated sentences that were longer than 30 words. This was initially done due to the insight that the median length of words of a “premise” was about 26 words, while the median length of a “hypothesis” was 10 words.

While the initial consideration was to conserve RAM and computation time, and avoid sparsity in our dataset, it would be interesting to see how variation of sentence length may affect our model performance.



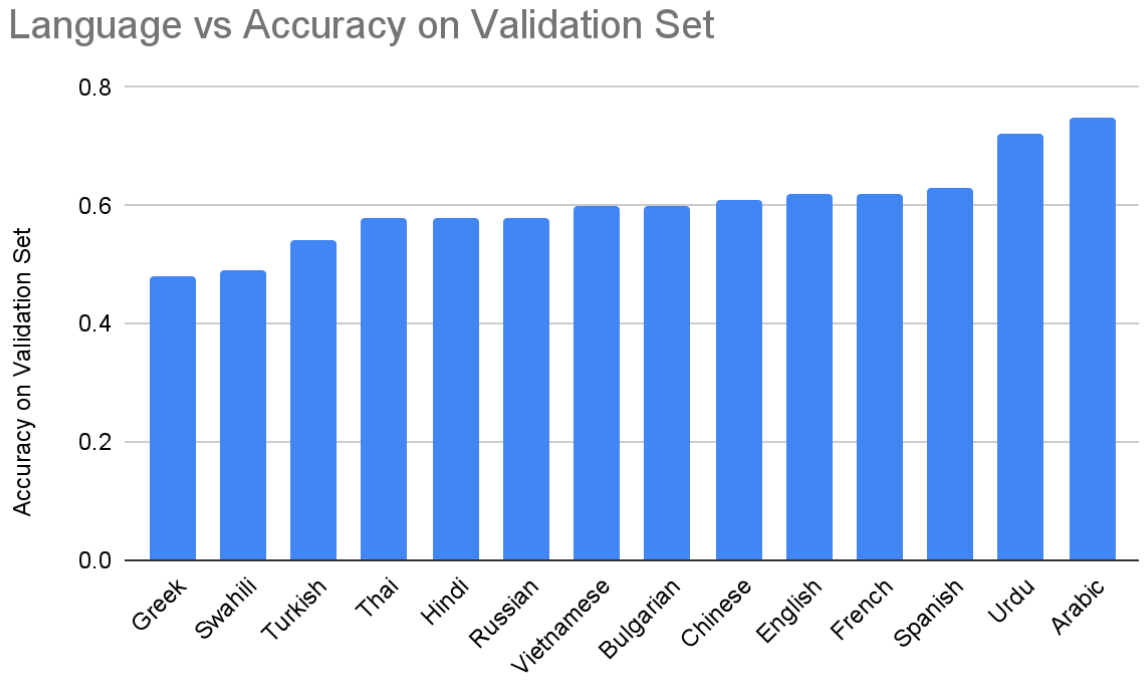
** XLM-RoBERTa (large) was not able to compute the max sentence length of 100 due to limited vRAM*

From the graph, we can see how different models handle the changes in maximum sentence length given. Interestingly, we see that models with larger parameters benefitted the most from the increase in max sentence length.

8.4 Experiment 3: Data Augmentation

Another concern from our analysis of the text data was that there was an unbalanced representation of the english language compared to the rest of the languages. This would inevitably lead to a difference in model performance by language.

Using the benchmark model, the accuracy on the validation set was computed by language in the graph below.



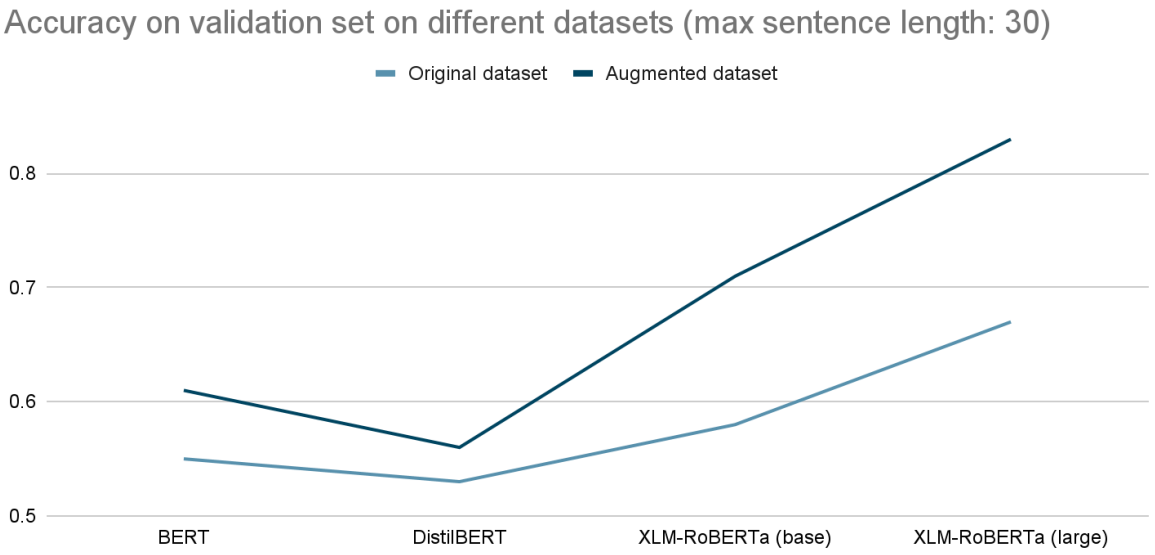
As seen from the chart, the model performs differently for different languages. Interestingly, even though English was overrepresented in the dataset, it did not perform the best in the validation dataset. A reason for this could be the complexity of languages, and how it may be easier to identify contradictions or entailments for certain languages.

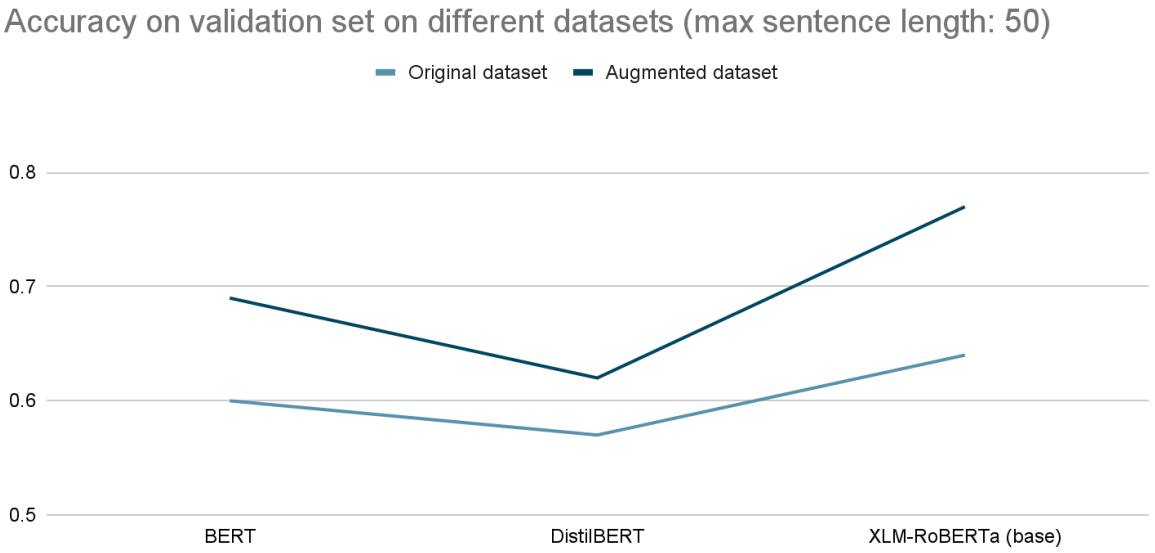
In order to combat the uneven representation of languages, data augmentation was introduced into the data pipeline that would help to increase the diversity of the dataset and also to increase the absolute size of the dataset.

1. The entire dataset was translated to English using the available open source models in huggingface's repository. This meant that an english dataset of 12k rows was created.

2. This english dataset was translated into every language in our dataset. Unfortunately, an English to Thai translation model was not available in huggingface's repository, thus we retained the original Thai text samples to train our model. Apart from Thai, the other languages could be translated from English, thus increasing our dataset size to around 160k.

Training our model on this larger dataset allowed us to improve the performance, as seen from the graph below.





** XLM-RoBERTa (large) with max sentence length of 50 was too computationally expensive to train.*

9 Results

Through the various experiments, the following have been achieved:

1. Through the testing of different models, the model with the larger parameters was able to perform better than a model with smaller parameters.
2. In the testing of different maximum sentence lengths, a longer sentence provided more information to the model (which goes against the initial hypothesis that longer sentences would generate a sparse dataset and hurt the model performance).

3. Data augmentation is able to improve the model performance by providing a more diverse and larger dataset for the model to use.

The highest accuracy that was achieved was using the following model and hyperparameter:

Model	XLM-RoBERTa (large)
Data	Translated dataset (160k samples)
Max sentence length	30 (limited by vRAM available)
Epochs	3
Accuracy on validation set	**83%**

We were able to improve our accuracy by 28% compared to the benchmark initially established.

10 Conclusion

10.1 Reflections

The most time consuming part of the project was learning about the transformers architecture from scratch, and reading the documentation of the huggingface transformers library. I felt that I was able to produce a satisfactory standard of performance and I managed to achieve what I set out to do with the amount of time given to do the project.

From the various experiments, it seems that I am also limited by the amount of compute power to achieve better performance. That being said, I am proud that I was able to innovate and think of interesting ways to augment the dataset to create more training data.

10.2 Improvements

More experiments could be done to finetune the model and hyperparameters. An interesting blog I came across also talked about finding the perfect learning rate for model training. These are tweaks that could be done to the model to improve its current performance.

Above the tweaking of the existing model, supplementing our dataset with external datasets (e.g. The Stanford Natural Language Inference (SNLI) Corpus) should also improve the performance of the model.

11 References

Conneau, A. (2019, November 5). *Unsupervised Cross-lingual Representation Learning at Scale*. ArXiv.Org. https://arxiv.org/abs/1911.02116

Devlin, J. (2018, October 11). *BERT: Pre-training of Deep Bidirectional Transformers for Language. . .* ArXiv.Org. https://arxiv.org/abs/1810.04805

G. (2018). *bert/multilingual.md at master · google-research/bert*. GitHub. https://github.com/google-research/bert/blob/master/multilingual.md

Kulshrestha, R. (2020, November 22). *Transformers in NLP: A beginner friendly explanation | Towards Data Science*. Medium. https://towardsdatascience.com/transformers-89034557de14

Sanh, V. (2019, October 2). *DistilBERT, a distilled version of BERT: smaller, faster, cheaper. . .* ArXiv.Org. https://arxiv.org/abs/1910.01108

Vaswani, A. (2017, June 12). *Attention Is All You Need*. ArXiv.Org. https://arxiv.org/abs/1706.03762