A Bilingual BERT Model Ensemble for English-based Multitask Fine-tuning

Stanford CS224N Default Project

Ryan Dwyer
Department of Computer Science
Stanford University
rydwy01@stanford.edu

Abstract

BERT has become a popular architecture that only becomes more popular when cleverly fine-tuned for particular tasks. In the area of multi-task fine-tuning, these design decisions are more important since we have one model trying to make a prediction on tasks that may have very different formats and embeddings. There are versions of BERT that are pretrained on just English text and ones that are pretrained on multiple language datasets. A single language's text is often used to train models performing tasks in that same language, but this paper investigates the utility of polyglotism for language models to see if some transfer-learning benefit can come from fine-tuning on multilingual data in addition to English data. I use the ensembling of two models: an uncased English minBert model pretrained on Sentiment analysis and an uncased multilingual (top 104 most common languages) one pretrained on multilingual sentiment analysis. I use their combined output and pass to the final multitask classifier and evaluate its predictions on English datasets for Sentiment Analysis, Paraphrase Detection, and Semantic similarity. I also make architectural decisions in this final classifier in an attempt to increase performance based on accuracy and Pearson Correlation.

1 Acknowledgement

- Mentor: Josh Singh

2 Introduction

The most straightforward way to perform well on a specific NLP task is to use a single dataset and split it into train, validation, and test sets in order to get the best results. However, as large language model technology is becoming increasingly prevalent not just in the research space but also commercially, there is increasing demand for models that can complete various NLP-related tasks. Though it poses some challenges around finding the best architecture and weight-structure to optimally perform across all tasks, many papers have arisen discussing various techniques to make models increasingly powerful and capable.

One of the methods that many researchers have tried in order to improve multi-task learning abilities is the use of an Ensemble– incorporating multiple language models together (often with multiple datasets) in order to develop more robust embeddings. This may seem too obvious to be useful, as it seemingly just applies the principle of "two is better than one" to language modeling, however there are many scenarios in which bringing together two models can effectively make up for each others short-comings; one may notice a feature in training data that another may not, so finding some way to represent both their outputs in a final prediction leading to increased resilience can lead to reduced error frequency at test time. As stated in Banerjee (2020), a simple yet effective method for taking

into account both model outputs is to put each of their respective embeddings into a linear layer of the form $y = xA^\top + B$:

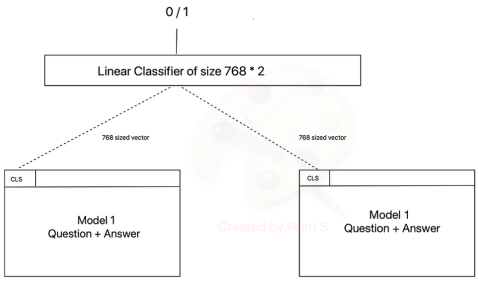

Figure 1: An example diagram for how models working on Q+A tasks can combine their outputs into a linear classifier Banerjee (2020)

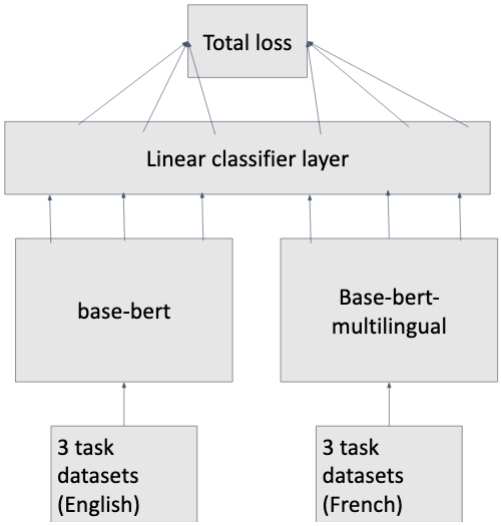

Figure 2: My model's diagram, which is similar to figure 1 but involves each of the tasks datasets being fed into separate bert models and separately into classifiers until they are combined using the total loss function specified in

Using code similar to the format:

```
self.bert_model_1 = BertModel(config)
self.bert_model_2 = BertModel(config)
self.cls = nn.Linear(self.n_models * self.config.hidden_size, 2)
```

Figure 3: Simple code setup for an ensemble Banerjee (2020)

Ensembling models can prove to be a useful tool for bridging the human language disparity that is all too common in natural language modeling applications, wherein English is often over-represented in most corpuses. In this project, I use an ensemble-like method to incorporate two different pre-trained BERT models, one multilingual and one English based, to train on the target datasets (SST for

sentiment analysis, Quora for paraphrase detection, and STS for semantic similarity) as well as French-language datasets performing the same tasks, and measure performance to see if multilingual structural context may have any benefit on a model's ability to create meaning from text.

3 Related Work

The initial inspiration for the use of multilingual data comes from a paper on cross-lingual information retrieval (Jiang et al. (2020)), which discusses the usefulness of fine-tuning the BERT language model using non-English datasets with weak supervision, and shows promise in the domain of internet search queries being able to highlight relevant corpora even if the query is in a different language than the document (in their study, Lithuanian). Though in a different task domain, the success of both English and a non-English finetuning on a BERT model showed a large amount of transferability without the model having been explicitly pretrained on Lithuanian, which could show promise for other tasks that may not even have to do with any cross language aspect.

Other non-Bert model research had also resulted in significant success such as in the case of XLM, which actually outperformed multilingual BERT (mBERT) on a number of metrics Conneau et al. (2020). It was shown to be beneficial to low-resource representation in language models such as Swahili and Urdu, but still maintained comparable performance on monolingual tasks.

The potential of taking successes from a domain such as cross-lingual NLP and applying it to improve the tasks of English-based sentiment analysis, paraphrase detection, and semantic similarity, therefore, appeared worth experimenting with. Transfer learning as described in the slides associated with Ruder et al. (2019) can allow for weights from one task to adapt well to another through the use of fine-tuning. A great way to further diversify the method of fine-tuning is to ensemble different models and different language datasets.

Risch and Krestel (2020) discuss in their paper up with an excellent method of a model ensemble for the application of detecting harmful/hate speech across multiple languages that they describe as "bagging" (also known as bootstrap aggregation). This takes into account soft voting on the output classification from 25 different BERT models. The results were promising with an increase in F1 score in four out of 5 of their specific NLP tasks, and overall showed the capability of the bagging methodology for successful ensemble performance in a domain not necessarily centered around multilingualism (i.e. hate speech), but one that could benefit from its introduction to the algorithm.

My experimentation takes into account research surrounding BERT, using multiple models for single and multiple tasks, and using more than one language to diversify datasets and applies a new method that attempts to transfer understanding of French language for our three tasks using multilingual base BERT and apply its weights during fine-tuning on the mostly English-based base BERT. Thus, unlike prior work, my methodology ensembles two different models (similar in architecture but different in their pretraining corpora), using two different fine-tuning dataset languages (French and English), to attempt to enhance performance on their shared three tasks.

4 Approach

My approach was to initially create a baseline without the introduction of another BERT model or data from outside of the three tasks that were to be tested. The initial baseline setup was a simple pretraining/fine-tuning of a minBERT implementation using the Stanford Sentiment Treebank (SST) as well as the CFIMDB dataset, both of which used to predict sentiment on different scales (the former with 5 possible buckets for sentiment while the latter was a binary positive or negative). This performed well in pretrain mode but even better in fine-tune, which showed BERT's potential for single-task classifying.

To create a multitask baseline, I implemented a simple forward function which retrieved embeddings from BERT after plugging in the appropriate input for the task that called it, and passed that after

applying dropout layers to increase randomness and decrease the chance of overfitting, passed the embeddings to a linear layer to get logit(s). For each epoch (I ran 10), these logits were calculated for each of the three tasks, each of which got its own loop to work with a batch of 8 examples. I used cross entropy to calculate the loss and Using cross entropy loss functions and an Adam optimizer, and had each of the tasks contribute to the same total loss for a given epoch, in a similar manner to the multitask additive loss mentioned in Bi et al. (2022). One thing I had to significantly take into account was the differences in the structures of my dataset inputs as well as the desired output label from each. The paraphrase and semantic similarity tasks had a single logit output (i.e. is or is not a paraphrase, and senetence A and B have a 0.96 similarity rating) while sentiment analysis had the 5 logit outputs. More importantly, however, is that in order to get an embedding from BERT it is necessary to create a single input string, and the paraphrase and similarity datasets deal with sentence pairs as inputs. Thus, I concatenated each sentence's BERT embedding together after dropout together and then passed that concatenation to the linear activation layer to get the necessary logits.

For the ensemble and non-English language data experiment, my approach was to use Huggingface's bert-base-multilingual-uncased to generate emebddings on three French language datasets I also found on hugging face. I chose French because I theorized that a language with similar root words and grammatical overlap with English (such as a Romance language like French) may yield better results.

I gave read in each of the French language datasets and processed them in their own designated loop for each epoch, just as I had with the original 3 English datasets, and incorporated their losses into the total loss to be averaged.

Once all the data was processed, the pipeline was set up to train and test on the English data.

5 Experiments

I implemented my planned approach in a series of experiments that involved tweaking training and testing structure of the model-fine-tuning file, bringing in non-English from HuggingFace and preprocessing it to allow it to be plugged into the model in the same way as the English dataset, and comparing results between runs to get a picture of whether any benefit may have come from this method of fine-tuning and whether anything could be done to improve it. My total number of separate experimental runs were threefold– once for pretraining and fine-tuning the minBert model with sentiment analysis data (from part 4 of the default assignment), once with neither non-English data nor fine-tuning to serve as a baseline multitask BERT predictor, and once with bilingual data and multilingual bert being used all in the same implementation.

5.1 Data and Models

My experiments involved use of the following datasets:

- CFIMDB dataset: made up of 2,434 highly polarized movie reviews from the popular website IMDb. Analyzing more extreme sentiment examples may make it easier for the model to further discretize the more in-between sentiment categories it sees in a larger, more sentimentally neutral, corpus. These had a simple binary label of positive or negative due to its polarizing content. It has a train-dev-test split of 1,701-245-488 examples.

- Stanford Sentiment Tree (SST) dataset: made up of 11,855 movie review sentence examples and 215,154 sentences annotated by 3 human judges for sentiment. This dataset had 5 different buckets for output: negative, somewhat negative, neutral, somewhat positive, positive. It has 8,544 training, 1,101 dev, and 2,210 test examples.

- Quora Paraphrase dataset: made up of 400,000 sentence pair examples used to detect whether sentence 2 is a paraphrase of sentence 1. Thus, the output is a binary label of 1 or 0.

I operated on a subset of the large dataset, using 200,000 total examples split into a training set of 141,506, validation set of 20,215, and a test set of 40,315

- SemEval Semantic Similarity dataset (STS): made up of 8,628 sentence pairs that vary on a scale of 0 (unrelated) to 5 (equivalent meaning). Instead of using a simple accuracy measurement to analyze results, it made more sense to use Pearson Correlation . This had a train-dev-test split of 6,401-864-1,726Eneko Agirre and Guo. (2013)

- The French book reviews for prompt sentiment analysis dataset (HuggingFace) Centre Aquitain des Technologies de l'Information et Electroniques (2023a): This dataset consists of french language book reviews that are labeled for sentiment analysis tasks with a 'pos' or 'neg' string for positive or negative sentiment (thus requiring the labels to be changed to 0 for 'neg' and 1 for 'pos' in order to be processed by the model). This dataset also comes with an LLM style prompt associated with it such as "Comment" or "Opinion" (written in French) and draws the actual book review example another french dataset from Abir ELTAIEF (2023), but I chose the one with prompts in case they may be helpful to the model in flagging down the correct sentiment label. It comes with 270,423 examples that I manually split into train-test-dev of 189,296, 40,563, 40,563 (70/15/15%)
- PAWSX paraphrase dataset Yang et al. (2019): made up of 23,659 evaluation pairs for paraphrase detection and 296,406 machine translated examples in multiple different languages. I chose the subset of this dataset that worked just in French, which had 49,401 training examples, 2,000 dev, and 2,000 test. The dataset is made up of a sentence 1, sentence 2, and binary label pair for whether sentence 2 is a paraphrase of sentence 1.
- The stsb multi mt fr prompt sentence similarity databaseCentre Aquitain des Technologies de l'Information et Electroniques (2023b), like the French book reviews database, is a prompt-based subset of the DFP, with the original data coming from **?**. It is made up of French sentence similarity examples that contain a sentence 1, sentence 2, and a similarity score on a scale from 0 to 1. It is made up of 103,482 train, 27,000 dev, and 24,822 test examples.
- The bert-base-multilingual-uncased modelDevlin et al. (2018) uses the basic base bert architecture and has been pretrained on the top 102 languages on Wikipedia using an unsupervised masked labeling model (MLM) objective. multilingual data. It doesn't distinguish between cases (which I didn't believe was necessary for understanding of the three tasks for this project.
- The bert-base-uncased Devlin et al. (2018) which is just a simple implementation of bert that ignores case, trained on multilingual data but on an English language corpus in a self-supervised MLM fashion.

5.2 Evaluation method

Performance was evaluated quantitatively for all datasets during training in order to actively improve the weights at runtime. For all sentiment and paraphrase tasks (both English and French), accuracy was the measure of success, while in the case of the Similarity measurements Pearson Correlation was used, because it's more useful to know how close a prediction was to the correct label rather than to know whether or not the exact similarity score was predicted due to the large range of possible scores.

5.3 Experimental details

I began by creating a baseline multitask classifier that used bert-uncased as it was used in part 1 of the default project, with simple fine-tuning on each of the 3 tasks, summing together and averaging their losses. Each epoch cycled through the datasets in their entirety, and ended up taking around 9 hours to train on a T4 GPU. I used a learning rate of 1e-05 and ran with the finetune option, and chose AdamW as my optimizer.

For my fully implemented extension, I rewrote the same code structure from the baseline in a new file I called multilingual.py and worked in the new data and model. I cycled linearly through each task in

batches of 8 for each of 10 epochs, and added in 3 more for loop for the 3 language datasets. I adjusted my forward and predict functions to have the French language datasets call the multilingual bert model to get their embeddings, and the English language datasets to call the regular base-bert-uncased. I used the finetune option when runnig my code because I did not want to pretrain and change the original weights of each respective BERT model. I used a learning rate of 1e-05, and AdamW as my optimizer. Overall, I kept the underlying implementation as similar to the baseline as I could to ensure the results I was getting were due solely to the inclusion of multilingualism in my training, to analyze it's particular effect on accuracy and Pearson correlation across the tasks.

5.4 Results

The performance was decent on the dev set for an overall accuracy of 0.63, though it showed evidence of potential overfitting given the much higher training accuracy of around 0.93. On multiliingual model, there was overall subpar performance as only the sentiment task had a (slight) increase above the baseline, while the other two did worse. As can be seen in the graph, the dev set accuracies overall increased during training , with the paraphrase dataset being the most unsteady.

	Dev Accuracy			Test Accuracy		
	SST	Quora	STS	SST	Quora	STS
Baseline Fine-tune	0.477	0.753	0.347	0.476	0.755	0.284
Bilingual Fine-tune	0.510	0.467	0.334	0.526	0.466	0.2780

Figure 4:

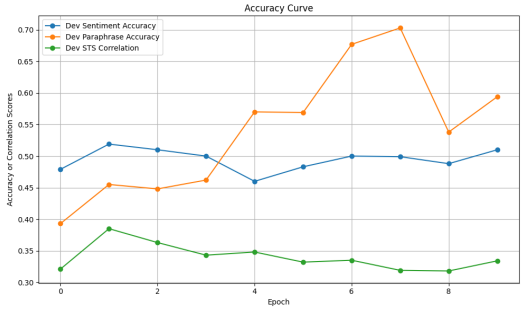

Figure 5:

6 Analysis

It is possible that the poor performance of my model over the baseline can have to do with the principle of ensembling bert models only working sometimes. Particularly, there are often tasks in which some gradients from one model and go well with another and result in more robust weights, while in other scenarios the gradients may conflict and actually have negative effects on performance. I imagine that this is along the lines of what occurred with my model, because in general though the French and English language have lots of overlap and similar structures, some grammatical formatting that may have resulted in one outcome in the French dataset may have been a negative one on the English dataset. Additionally, the fact that this was not traditional ensembling in the sense of multiple models acting on the exact same datasets may have led to some of the conflict that negated performance.

7 Conclusion

Summarize the main findings of your project and what you have learned. Highlight your achievements, and note the primary limitations of your work. If you'd like, you can describe avenues for future work.

Despite the somewhat poor accuracy and Pearson correlation results, it appears that the theory behind a Bilingual BERT model ensemble can lead to better results with some tweaks– namely working in tandem with a few architectural schema that could negate the effects of gradients crashing.

One thing to be considered in the future could be gradient surgery, which projects gradients that conflict onto the same axis during to help work toward reducing loss. Yu- mentions this method, and I believe it could apply well to the problem of getting transfer learning to work when languages and models predictions may conflict.

I also would like to use more than 2 models, to get a better measurable effect from the ensembling. In the Risch and Krestel (2020) paper, they used over 25 BERT models to ensemble and train on multilingual data that included lower resource languages like Tamil and found promising results in the area of hate speech detection.

Overall, this project shows that the incoproation of multiple languages in NLP is important not only for social reasons such as broader societal representation and decreasing bias in training data, but also may very well have some benefits in more monolingual task settings.

References

Abir ELTAIEF. 2023. french$_b ook_r eviews(revision534725e)$.

Rishab Banerjee. 2020. Ensembling huggingfacetransformer models. *Medium*.

Qiwei Bi, Jian Li, Lifeng Shang, Xin Jiang, Qun Liu, and Hanfang Yang. 2022. MTRec: Multi-task learning over BERT for news recommendation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2663–2669, Dublin, Ireland. Association for Computational Linguistics.

Centre Aquitain des Technologies de l'Information et Electroniques. 2023a. Dfp (revision 1d24c09).

Centre Aquitain des Technologies de l'Information et Electroniques. 2023b. Dfp (revision 1d24c09).

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Mona Diab Aitor Gonzalez-Agirre Eneko Agirre, Daniel Cer and Weiwei Guo. 2013. Semantic textual similarity. in second joint conference on lexical and computational semantics. In *volume 1: proceedings of the Main conference and the shared task: semantic textual similarity, pages 32–43,*.

Zhuolin Jiang, Amro El-Jaroudi, William Hartmann, Damianos Karakos, and Lingjun Zhao. 2020. Cross-lingual information retrieval with BERT. In *Proceedings of the workshop on Cross-Language Search and Summarization of Text and Speech (CLSSTS2020)*, pages 26–31, Marseille, France. European Language Resources Association.

Julian Risch and Ralf Krestel. 2020. Bagging BERT models for robust aggression identification. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, pages 55–61, Marseille, France. European Language Resources Association (ELRA).

Sebastian Ruder, Matthew E. Peters, Swabha Swayamdipta, and Thomas Wolf. 2019. Transfer learning in natural language processing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials*, pages 15–18, Minneapolis, Minnesota. Association for Computational Linguistics.

Yinfei Yang, Yuan Zhang, Chris Tar, and Jason Baldridge. 2019. PAWS-X: A Cross-lingual Adversarial Dataset for Paraphrase Identification. In *Proc. of EMNLP*.