Second Year Project
Feasibility of fine-tuning on one language to predict sentiment in another using XLM-RoBERTa

Ivan Petrov
ivpe@itu.dk

Jesper Terkildsen
jete@itu.dk

Daniel Ring Hansen
darh@itu.dk

Abstract

A lot of natural language processing today is conducted in English, but it is equally as important to have models that can be used for more than just one language. Problems occur when applying a strictly English model on non-English data, since every language has nuances not available in English. This is where multilingual variations of BERT come in. This report explores using the XLM-RoBERTa model to predict sentiment in 6 languages using the features provided in lang2vec. In addition, we try and determine how good each language is at prediction each others' sentiment, and if there is one language that performs better on all or most of the others. We found that the best lang2vec feature for transferability is phonological. Based on the mean F1-scores of the fine-tuned models we found that the performance of the English fine-tuned model was on par with French (0.914) and only slightly worse than German (0.916). Therefore, we suggest that English is the best language to fine-tune XLM models on for this use case due to greater data availability and insignificant performance difference.

GitHubRepo: [3]

1 Introduction

Natural language processing (NLP) is a field that has boomed in the last couple of years with the invention of models like BERT[5] that consistently yield good performance on a number of diverse NLP tasks. Most of these experiments have been conducted on English data, a language for which there is an abundance of data. Since the benefits of BERT models are equally as useful in other languages as they are in English, multi-lingual variants have been developed, such as XML-RoBERTa[4], a pre-trained model trained on 100 languages by a team from Facebook[7]. While this helps a lot with bringing the BERT and its variants to other languages, we wanted to see if we could fine-tune the XML-RoBERTa model on one language and get good performance on other languages. This could be useful since, instead of training a model for each language, we could save both money and time by only fine-tuning one with good performance in the other language domains, since these models take a lot of time while training. This could be extended to help us in related languages, such as if we were to have plentiful data in Danish, we could use it to predict sentiment in Norwegian and Swedish, since they are related to a high degree.
In this paper we would like to answer the following: What performance can be achieved by fine-tuning XLM-RoBERTa on one language for detecting sentiment in other languages, and what features can help predict transferability? Is there a symmetrical relationship between the languages' performance, and is there a language which works significantly better than others? What is meant by symmetrical performance is whether or not two models fine-tuned on two languages perform similarly at detecting sentiment on each other's languages.
We will go about answering these by fine-tuning XLM-RoBERTa to various languages and testing the produced models on each of the languages to inspect their performances; as well as to correlate them to lang2vec distances[6], a series of typological, geographical, and phylogenetic vectors denoting the differences between languages, in an attempt to find a metric that can be used to predict the performance level of these models.

2 Related works

2.1 Multilingual NLP

In the paper "How multilingual is Multilingual BERT?"[8] they have a pre-trained multilingual variation of BERT called M-BERT. In their re-

search they looked at how good M-BERT was at Named Entity Recognition (NER) and Part of Speech Tagging (POS) across a series of languages, some with different language scripts, such as Hindu. They found that high lexical overlap, which means the languages share a lot of common words, have better cross lingual performance, but that it also works for languages with no lexical overlap. Further they also found that typological features, such as word ordering (e.g. subject, verb, object) have the best performance. While they looked at the language structures themselves, we are using the lang2vec features[6] which, besides typological features, also includes geographic and phonological features.

2.2 Transformer models, BERT, RoBERTa

Instead of directional models which read the input from left to right, such as RNNs, the transformer model[9] uses self-attention in the encoder and decoder parts of itself in order to relate every word to every other word in the input and find a score of how closely they are connected. A way that the BERT model[5] differs is that it also uses an additional type of encodings which exist in order to separate input segments from each other. RoBERTa[7] further differentiates from the BERT model by dropping one of the pre-training steps and increasing the robustness of the model via a change to how the masking is done within the self-attention layers. It is also trained on a larger amount of data points and implements changes to the batch size and training steps. The XML-RoBERTa[4] model even further differs by introducing new pre-training method for cross-lingual language modeling and being trained on a much bigger corpus.

3 Methodology

3.1 Data Description

The project pipeline started with getting the dataset of amazon product reviews in English, Japanese, French, German, Chinese, and Spanish from AWS[2]. The reviews without any review text were given empty string values to allow the model to process them. They were not thrown out since empty reviews still had star ratings, which meant that our model might be able to figure out some relationship between a lack of review text and sentiment. Reviews with ratings of 2 star or less were marked as negative sentiment reviews, and those with ratings of 4 stars or more were marked as positive sentiment reviews. Reviews with 3 stars were deemed to be neutral and removed from the data that was to be processed. This was done because the 3 star reviews, accounting for 20% of each language's review dataset, would prolong training time while also being difficult to determine sentiment for automatically.

3.2 Feature engineering

The baseline model that we used had a vocabulary of 3789 words and used whole-word tokenization. The vocabulary consisted of words that appeared in more than 0.005% of the sentences. This cutoof was implemented in order to limit the memory usage of the model.
The state of the art model uses a Byte level Byte-Pair[10] encoding for the creation of the token vocabulary with 256 bytes of base tokens, a special end-of-text token, and the symbols learned with 50,000 merges of the most common symbol combination tokens. This means that, in total, the vocabulary will contain a total of 50257 tokens.

3.3 Model selection

For the baseline model, we went with the naive Bayes method due to its simplicity. There are a lot of things to improve upon with our state of the art model. As for the state of the art model itself, we picked a larger and a smaller pre-trained model; fine-tuned them with the development dataset, which is around 40 times smaller than the actual training set; and recorded the times it took to train. This was to see if the better performance of the larger model came at a reasonable increase in training time. The times can be found in Table 1. We chose to use XLM-RoBERTa since the performance improvement of 5.3% was deemed worth the time difference. In the CPU-only test the model was given 12 cores from either an Intel(R) Xeon(R) Gold 5218R CPU or an AMD EPYC 7352 whereas the CPU+GPU combination was on a desktop with an Intel(R) Core(TM) i7-4790 8 core CPU and a RTX 2070 with 8GB of vRAM.

	XLM-RoBERTa	Distill-BERT-M
CPU	28 min	28 min
CPU+GPU	129 sec	85 sec

Table 1: Training times on development dataset

3.4 Model training

The training of the state of the art models was done on the HPC cluster where the pre-trained model was loaded and fine-tuned for sentiment analysis with the training dataset.
The naive Bayes baseline model was trained on the combined training data from all six languages.
The baseline model training was also done on the HPC cluster using the same i7-4790 8 core CPU as the state of the art models. In training this model, the system memory was the limiting factor due to the way that the naive Bayes model works. We used 99.5% of the 24 GBs worth of memory that the system had with the ≈ 3800 token vocabulary. Using naive Bayes without any constraints would attempt to allocate 12.3 TiB or 13.5 TB of memory.

3.5 Predictability feature

Putting the F1-scores acquired from the fine-tuned models on the y-axis, and the distances of all 6 language to the other 5 on the x-axis, we then calculated linear regression coefficients for each of these regressions. We ended up with 33 five-data point plots as 3 of the regressions were impossible to calculate due to all distance values being the same. An example of this is Japanese being in its own genetic family and therefore having the same distance to all the other languages. We then added up all the R^2 values and took the mean of them for each type of distance. The results reside in Table 2.

4 Experiments

4.1 Model Specifics

The model takes an input of max length 256 tokens. Anything over is truncated to the max length. On top of the pre-trained model, we had two other layers. The first was a dropout layer with a 30% probability for regularization of the model. The second layer was a linear output layer that resized the 768-size output of the model into two neurons since we had two classes. The positive/negative is decided by the index of the highest value of the output vector. The learning rate that we started with was 0.00001 and our training batch size was 8. We used the Adam optimizer and Cross Entropy Loss function. Cross Entropy was easier for us to get working than Binary Cross Entropy due to constant shape mismatches and therefore Cross Entropy was used.

4.2 Chosen metrics

Throughout our experiments we kept track of all four of the possible outcomes: true positives, true negatives, false positives and false negatives. From those we could derive any of the metrics needed such as accuracies or F1-scores for the models. Since our classes were balanced, accuracy was a good contender for what metric to report. Ultimately we decided to report the F1-scores as those represent the different types of errors rather than just pure accuracy.

5 Results

Table 3 shows the F1-scores for all models' predictions for the test sets. Table 5 in turn, shows the F1-scores from our count-vectorizer baseline model. Comparing the scores between the baseline and the state of the art, the western languages differ by anywhere from 0.04 to 0.09 F1-score. The baseline model performed terribly on Japanese and Chinese and the accuracies were close to 50%. The state of the art overcomes this limitation and achieves a much better performance on these languages.
Figure 1 shows the difference in the F1-Scores between a model fine-tuned for a language predicting another language's data and the reverse, where the language being fine-tuned for and the language of the data where sentiment is being predicted are swapped. Positive values show that the language being fine-tuned for performed better when predicting sentiment on another language's data than the model fine-tuned on the other language did predicting the first one's language's data, and vice-versa with negative values.
From Table 2 it is seen that out of all the distance features, the phonological had the strongest R^2 values. This suggests that the values on the y-axis are best explained by that type of distance.

6 Analysis

Figure 2 shows a sample confusion matrix of the fine-tuned German model predicting on the English test set. We see there is 105 (2.625%) false positives and 126 (3.15%) false negatives from its predictions.

	Genetic	Geographic	Syntactic	Inventory	Phonological	Featural
Mean R^2	0.256	0.398	0.239	0.451	0.522	0.276

Table 2: Mean R^2 values for each distance type from the URIEL database

	ENG test	JPN test	FRA test	DEU test	CHN test	ESP test
ENG	0.942	0.914	0.928	0.937	0.845	0.918
JPN	0.904	0.932	0.904	0.911	0.861	0.910
FRA	0.929	0.911	0.944	0.934	0.832	0.932
DEU	0.926	0.920	0.929	0.946	0.846	0.928
CHN	0.910	0.887	0.907	0.916	0.881	0.907
ESP	0.916	0.897	0.929	0.927	0.807	0.945

Table 3: Models' predictions on all 6 test sets. The y-axis is the language that the model was fine-tuned for and the x-axis is the language of the test set

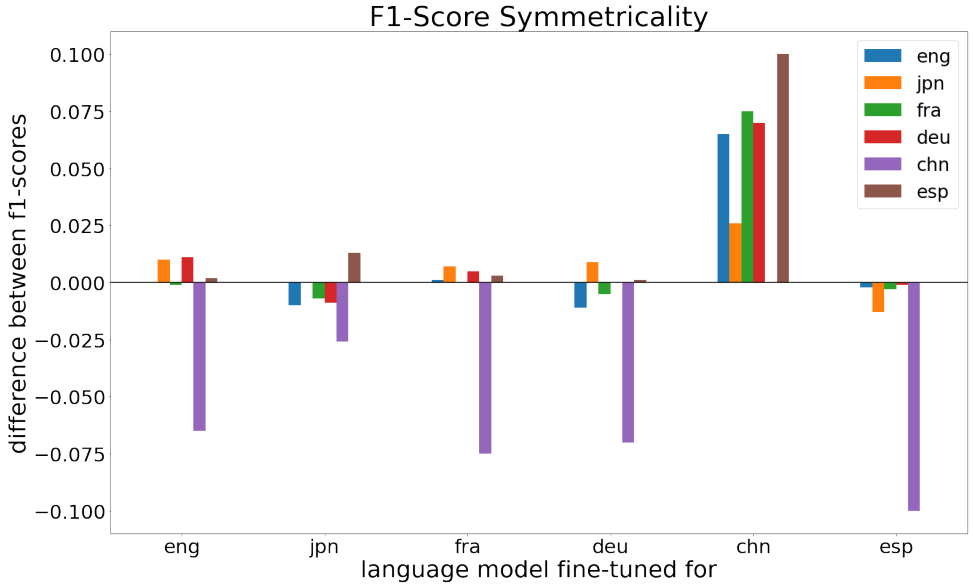

Figure 1: Numerical difference between F1-scores of the fine-tuned models on each language and their symmetrical counterparts, that being the reverse of the language being fine-tuned for and language tested on

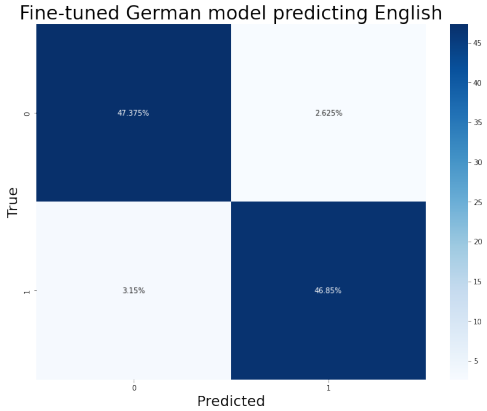

Figure 2: Confusion matrix of German model on English test data

Of those 231 falsely predicted cases we looked at 100 positive and negative reviews to find patterns in each. There were some instances that would be hard even for a human to classify. For a lot of the ground truth negative reviews they just wrote a statement like *"Have not used it yet"* or *"Smells a little oily..."*, no words to really drive a sentiment from the person. It is understandable how the model might confuse these cases.

As for the ground truth positive reviews there were primarily two types. A good amount included a positive thing but with a small complaint at the end like *"Tasted very good for a vitamin. However, did not see any difference in hair growth."*. The other type were just complaints or closer to being negative reviews. It seemed like the models were right in more cases than it appeared, and that some

	ENG	JPN	FRA	DEU	CHN	ESP
Mean F1-Score	0.914	0.904	0.914	0.916	0.901	0.904
Standard Deviation	0.032	0.021	0.038	0.032	0.013	0.046

Table 4: Mean F1-scores and Standard Deviations for language the model was fine-tuned for

	ENG	JPN	FRA	DEU	CHN	ESP
F1	0.837	0.156	0.849	0.864	0.167	0.839

Table 5: F1-scores for baseline model tested on each language

mistakes were made when the review was marked by the user.
One thing that stood out was that 79 of the 231 (34,2%) falsely classified reviews contained the word "but". The use of this word might indicate some level of nuance or complexity to the sentiment of the reviewer, where they initially are negative but comes with a positive aspect at the end, or vice-versa.

7 Discussion

7.1 Summarization of key findings

The F1-score analysis shows that German, English, and French performed the best with F1-scores of around 0.915, with German having the best performance by a small margin. Japanese, Spanish, and Chinese also performed well, with scores also lying above 0.900, but still 0.010 below the other three languages.
From our symmetry analysis on the languages, we found that the model fine-tuned for Chinese was performing better on other languages than the models fine-tuned for the other languages performed on Chinese test data. This is likely due to all the fine-tuned models, the one fine-tuned for Chinese included, struggling on the Chinese test data, showing exclusively F1-scores below 0.9. This granted the model fine-tuned for Chinese an easier time beating its symmetrical counterparts despite its low F1-scores. Aside from that, the models fine-tuned for French and English performed better than 4/5 and 3/5 of their symmetrical counterparts, respectively, though oftentimes by very small margins.
Since English had a insignificantly lower F1 performance relative to German, there is a significantly larger amount of data available to work with in English as it makes up a majority of the internet[1].
Looking into the feature to best predict transferability, we found that the phonological distance outperformed the other distances in terms of the F1-score.

7.2 Limitations

More languages would have allowed for more data points to be analyzed in regards to the relationship between the models' performances and the lang2vec features. Including more languages beyond the dataset that we used would have introduced a very large pre-processing and processing overhead as combining the dataset with another would likely cause class imbalance and format clashes. All this would be on top of the time spent finding more data.
Having an expert's input on the features of the languages could have been useful in regards to giving potential explanations for why we obtained certain results, such as the significance of phonological features of these languages, as well as specifics as to why Chinese both performed so poorly when used for fine-tuning and when having sentiment predicted on its text.

8 Conclusion

Being able to save resources and time on training models and gathering data is very important for any business that uses ML. Finding the optimal language to fine-tune a pre-trained model on will be beneficial to this cause. By looking into cross-lingual knowledge transfer we found that a satisfactory performance level can be achieved by using a single pre-trained model which is further fine-tuned for the task on even only one language. Further, before committing to a model, it is possible to estimate how well a multi-lingual model fine-tuned on one language will perform on another language using a language feature. In our case, we found that, out of the distances between languages from the URIEL database, the phonological distance returned the best result for this task.

References

[1] Most common languages used on the internet as of january 2020, by share of internet users. https://www.statista.com/statistics/262946/share-of-the-most-common-languages-on-the-internet/. Accessed: 2022-5-18.

[2] The multilingual amazon reviews corpus. https://registry.opendata.aws/amazon-reviews-ml. Accessed: 2022-05-03.

[3] Xlm-roberta sentiment repository. "https://github.itu.dk/ivpe/XLM-ROberta-Sentiment", 2022.

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

[5] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186, 2019.

[6] Patrick Littell, David R. Mortensen, Ke Lin, Katherine Kairis, Carlisle Turner, and Lori Levin. URIEL and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 8–14, Valencia, Spain, April 2017. Association for Computational Linguistics.

[7] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[8] Telmo Pires, Eva Schlinger, and Dan Garrette. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy, July 2019. Association for Computational Linguistics.

[9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[10] Changhan Wang, Kyunghyun Cho, and Jiatao Gu. Neural machine translation with byte-level subwords. *arXiv e-prints*, pages arXiv–1909, 2019.

A Appendix

Group Contributions:
There was no significant imbalance in group contribution.

B Appendix

Results from phase 1:
Our baseline model from phase 2, a small BERT(small_bert/bert_en_uncased_L-4_H-512_A-8), achieved an accuracy of 94.62% This was done on a strictly English dataset of music reviews.

Results from phase 2:
The same baseline model achieved an accuracy of 73.54% this phase included some custom made difficult cases from the original data.

C Appendix

Enlarged versions of the figures:
Figure 3
Figure 4

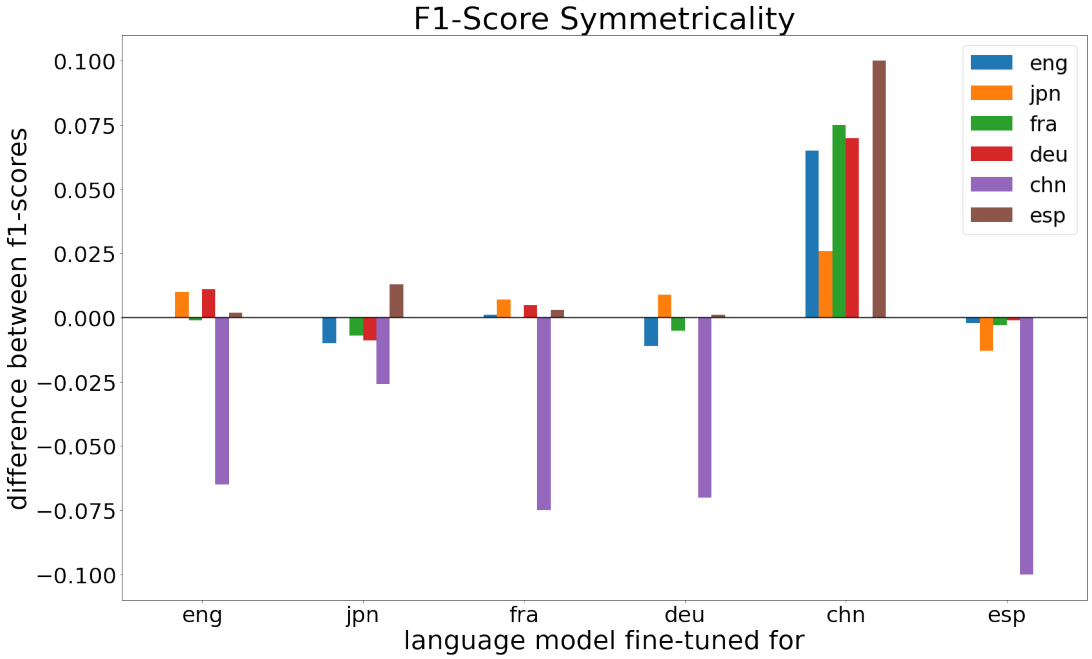

Figure 3: Numerical difference between F1-scores of the fine-tuned models on each language and their symmetrical counterparts, that being the reverse of the language being fine-tuned for and language tested on

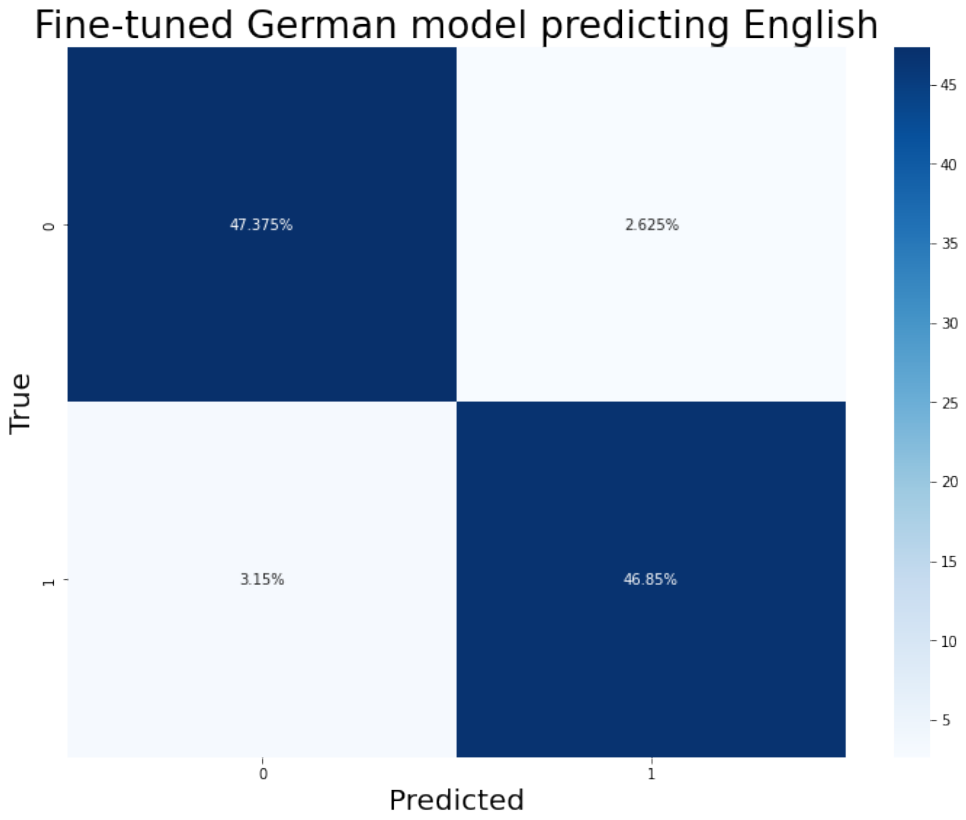

Figure 4: Confusion matrix of German model on English test data