Final Project Report: Lost in Translation and Transfer

Kelly Patel

20 December 2022

1 Introduction

Sentiment analysis has been a primary topic of research in numerous areas of application like journalism, marketing, finance, etc. In recent years, data sources rich with opinions have become widely available due to the prominence of social networks. However, some data reflects highly-resourced languages like English, but is found to be lacking in other low-resource languages. As the world becomes more interconnected through global communications, it is becoming increasingly imperative to analyze data from all over the world. Unfortunately, lexicons are not always available in other languages (marking them "low-resource languages") and it remains an expensive task to construct and label them in order to complete a standard sentiment analysis.

This motivates me to build upon some existing approaches and run a sentiment analysis for different languages using relatively small datasets. For the sake of this project, the focus will be on Slavic languages (further discussed in the Data Section).

Previous efforts in sentiment analysis for multilingual corpora entailed the translation of the corpora into English to then be analyzed for opinions. Researchers have used language-specific pre-trained models to vectorize English sentences for analysis (using models such as Word2Vec) (1). These embeddings are then passed through supervised machine learning classifiers such as Random Forest, Support Vector Machines, and Gradient Boosting Trees–which provide decent accuracies on out-of-sample data. In one study by Galshchuk, Jourdan, and Qiu in 2019, an accuracy of about 86% was achieved using translation and Random Forest.

I claim that there are semantic details lost in the translation from a source language to English. It is often hard to capture the full connotative meanings of words between languages even in person-to-person communication. Therefore, I have experimented with transfer learning to see if the nuances of a language can remain in tact while being analyzed. Instead of translating from a source to English and risking the loss of some semantic detail in that translation, I used BERT to transfer knowledge from a pre-trained English classification paradigm and then fine-tuned the model for a source language to then be analyzed for sentiment.

This was a domain-adaptation task in cross-lingual transfer, where I took a pre-trained BERT base and froze several layers (trained on an English corpus) and re-trained the remaining layers on a low-resource language. This differs from multilingual BERT since that model was completely trained in other source languages, using high amounts of data from Wikipedia articles in those languages, whereas I was curious as to see the effects of transfer learning from one high-resource language to one low-resource language. I experimented with differing numbers of frozen layers across tweets of 4 languages (English, Slovenian, Croatian, and Polish). While none of the low-resource languages yielded accuracies higher than English sentiment analysis, I was able to achieve 88.6% accuracy on Croatian tweets (and a 93% accuracy on English tweets using the same model). I was also able to explore how a different amount of tweets could influence the outcome. There are

certainly limitations with my approach, which I will also discuss in section 7.

My code can be found here: GitHub

2 Related Work

Deep learning has revolutionized NLP with powerful models. Efforts in multilingual models have been made in many facets by many groups. Among the most popular are the state-of-the-art mBERT, ELECTRA, XLM and XLM-R (2)(6). While XLM and XLM-R use BERT-large architectures, mBERT uses a smaller architecture (BERT-base). Of these models, XLM-R performs slightly better than the other models on the Cross-lingual Natural Language Inference (XNLI), which is a natural language inference task: given a premise and a hypothesis, does the premise entail or contradict the hypothesis (or is it neutral?). Due to the nature of XNLI and the languages tested in this experiment, I cannot directly compare my own results with the ones explored in Moberg and Lample's papers.

Model	D	#M	#lg	en	fr	es	de	el	bg	ru	tr	ar	vi	th	zh	hi	sw	ur	Avg
Fine-tune multilingual model on English training set (Cross-lingual Transfer)																			
mBERT	Wiki	N	102	82.1	73.8	74.3	71.1	66.4	68.9	69.0	61.6	64.9	69.5	55.8	69.3	60.0	50.4	58.0	66.3
XLM (MLM+TLM)	Wiki+MT	N	15	85.0	78.7	78.9	77.8	76.6	77.4	75.3	72.5	73.1	76.1	73.2	76.5	69.6	68.4	67.3	75.1
XLM-R	CC	1	100	**88.8**	**83.6**	**84.2**	**82.7**	**82.3**	**83.1**	**80.1**	**79.0**	**78.8**	**79.7**	**78.6**	**80.2**	**75.8**	**72.0**	**71.7**	**80.1**
Translate everything to English and use English-only model (TRANSLATE-TEST)																			
BERT-en	Wiki	1	1	88.8	81.4	82.3	80.1	80.3	80.9	76.2	76.0	75.4	72.0	71.9	75.6	70.0	65.8	65.8	76.2
RoBERTa	CC	1	1	**91.3**	82.9	84.3	81.2	81.7	83.1	78.3	76.8	76.6	74.2	74.1	77.5	70.9	66.7	66.8	77.8
Fine-tune multilingual model on each training set (TRANSLATE-TRAIN)																			
XLM (MLM)	Wiki	N	100	82.9	77.6	77.9	77.9	77.1	75.7	75.5	72.6	71.2	75.8	73.1	76.2	70.4	66.5	62.4	74.2
Fine-tune multilingual model on all training sets (TRANSLATE-TRAIN-ALL)																			
XLM (MLM+TLM)	Wiki+MT	1	15	85.0	80.8	81.3	80.3	79.1	80.9	78.3	75.6	77.6	78.5	76.0	79.5	72.9	72.8	68.5	77.8
XLM (MLM)	Wiki	1	100	84.5	80.1	81.3	79.3	78.6	79.4	77.5	75.2	75.6	78.3	75.7	78.3	72.1	69.2	67.7	76.9
XLM-R	CC	1	100	**88.7**	**85.2**	**85.6**	**84.6**	**83.6**	**85.5**	**82.4**	**81.6**	**80.9**	**83.4**	**80.9**	**83.3**	**79.8**	**75.9**	**74.3**	**82.4**

Figure 1: Multilingual Model Evaluation on XNLI

The limitation of mBERT is that it requires a huge amount of data which is not always available in low-resource languages. Also, according to (7), mBERT's performance improves with language similarity, showing that it is easier for mBERT to map linguistic structures when they are more similar. I expected the same phenomenon to manifest in my own experiments, as this seems to be tied to BERT's architecture. Still, mBERT does have peak performance in terms of sentiment analysis in most languages. For the purposes of this paper, I am not looking to beat mBERT's performance. Rather, I hope to see if transfer learning is a promising route to take in the case of low-resource language NLP tasks, using BERT as a vehicle for my experiments.

Transfer Learning is also a broadly explored topic especially in the realm of NLP tasks. As it helps extract knowledge from a source text form and applies it with relative ease to a different use case, it is often used to shorten the time spent on textual analysis. Named Entity Recognition, Intent Classification, Cross-lingual learning, Sequence Labeling and Sentiment Analysis are all realms where transfer learning has been applied and proven effective.

As for the combination of Transfer learning and multilingual modeling, there have been several attempts made. Kanclerz et al. (3) explore a novel technique for the use of language agnostic sentence representations to adapt the model trained on texts in Polish to recognize polarity in texts in other (high-resource) languages. They focus on creating a language-agnostic representation of each sentence and then predicting the sentiment of the text based on these representations.

Other groups focus on vectorization techniques like Word2Vec in tandem with machine learning methods like random forest, gradient boosting trees, and support vector machines (Galeshchuk et al., 2019). These efforts translate the low-resource language into English to perform the sentiment analysis. My work is a direct response to these methods, exploring whether it is possible that certain semantic information is lost in the translation between languages before sentiment analysis.

3 Background

For this task, I build upon existing methods for sentiment analysis to explore whether transfer learning from a model trained on a high-resource language can be applicable to sentiment analysis on a lower-resource language. In particular, I will be making use of BERT's base architecture (12 layers) to test whether learning is transferable between languages without the need for translation. Further, I will be testing the effects of freezing different amounts of layers. This latter experiment might be useful in saving compute power and potentially gaining accuracy. See section 4 for layer freezing details. Lastly, I check the advantage gained by having more labeled data.

Understanding BERT is imperative for this task. BERT stands for Bidirectional Encoder Representations from Transformers. It is a model that pre-trains deep bidirectional representations from unlabeled text by jointly conditioning on both left and right contexts in all layers (4). It is comprised of a multi-layer bidirectional Transformer encoder and makes use of several special tokens to indicate the separation of tasks, sentences, paddings, and masks for its masked-language learning scheme. Though an older model which appears to be superceded by OpenAI's efforts with GPT-2+ (10), BERT is still fairly reliable at noting trends and it is heavily documented, so insights into its inner workings are a bit more transparent than other, high performing language models.

4 Method

The experiments I run for this project rely on BERT layer freezing. I used BERT-base-uncased for sequence classification, as is used in many sentiment analysis studies (9). After using the BERT-base-uncased tokenizer to tokenize the preprocessed data, I freeze different layers of BERT to see which layers might retain the most semantic information and achieve the highest accuracies. For clarity, freezing layers means disabling gradient computation and backpropagation for the weights of these layers. Typically, we see models frozen from the bottom to top, as the first layers of a neural network are assumed to encode coarser features while the top layers encode finer, task-specific features. I also follow this scheme for my experiments. For instance, if I indicate 4 frozen layers, it means the first 4 layers of the model are frozen and the last 8 are unfrozen (BERT-base has 12 layers).

After the BERT layers, I have a dropout layer and then a simple Dense layer for classification. The parameter breakdown is found in *Table 1* for differing numbers of frozen layers:

Note that 0 frozen layers indicates a fully trainable model and 12 frozen layers indicates a fully frozen model.

5 Experiments

5.1 Data and Preprocessing

I used datasets in Slavic languages and English from CLARIN to perform my studies. These are standard .csv files with the tweet ID (Tweet text to be fetched using Twitter API) and a label (-1,

Num Frozen Layers	Trainable Params	Untrainable Params
0	109,483,778	0
2	95,308,034	14,174,744
4	81,132,290	28,351,488
6	66,956,546	42,527,232
8	52,780,802	56,702,976
10	38,605,058	70,878,720
12	24,429,314	85,054,464

Table 1: Trainable Parameters by Frozen Layers

0, 1) to indicate negative, neutral, and positive tweets respectively.

To compare to Galeshcuk's findings (1), I will be using corpora in Polish, Slovenian, and Croatian. These languages belong to the Indo-European family, but are members of the Slavic branch, which make them share fewer ties to English than the Latin based languages like Spanish or French. I used only Slavic languages that use a Latin-based alphabet, rather than a Cyrillic alphabet. This latter case is reserved for future study.

An interesting note, Slovenian and Croatian are further categorized into the South Slavic languages, while Polish is West Slavic. This sets the expectation that Slovenian and Croatian should exhibit more similar results while Croatian may deviate, as earlier mentioned, BERT improves performance as similarity to English increases. This seems to hold in my experiments (shown in Section 6).

Once the data is acquired from the Twitter API, it is preprocessed as follows:

1. All neutral tweets are removed (keeping only positive and negative tweets)
2. Positive and Negative Tweets are balanced
3. URLs are removed
4. Emoji's are removed
5. Punctuation is removed
6. Dashes and extra spaces are removed
7. Duplicate Tweets are removed

For English sentiment analysis, it is also customary to remove "stop words", however, as it is unclear what stop words might be in other languages, I refrain from doing so in these experiments. The number of tweets in each language can be found in *Table 2*. Note, due to class balancing, the number of positive tweets is equal to the number of negative tweets, meaning that each class gets an even split of the full dataset in that language.

For my preliminary experiments, I reduce the number of tweets for each language to 20,000 to match English, which I will be using as a control for comparison. I further split these sets into 15,000 tweets for training and 5,000 for testing.

Later, I increase the number of tweets for further studies, as explored in section 5.4.

Language	Number of Tweets
Slovenian	40,850
Croatian	63,118
Polish	98,920
English	20,000

Table 2: CLARIN Datasets

5.2 GPU and Resource setup

My personal laptop contains an NVIDIA GeForce RTX 2070 GPU, which has a shared system memory of 8073 MB, 2304 CUDA Cores, and a memory bandwidth of 352.06 GB/s. To mitigate an array of memory allocation errors, I had to set my batch size to 4 and ranged my learning rate from $1e-5$ to $2e-5$. For my 20,000 tweet runs, the model took between 20-40 minutes per epoch. For the larger runs (98,000 tweets), the model took 1.5-2 hours per epoch.

5.3 Workflow

After acquiring the data through Twitter API, I preprocess the data and split it into training and validation sets. After this, the number of layers to freeze is chosen, and then the model is run. Through preliminary testing, I found that the loss is minimized at 2 epochs. I started with a learning rate of $2e-5$, but for the freezing layers 0-6, I use a learning rate of $1e-5$, as the model starts to overfit. As mentioned in section 5.2, I use a batch size of 4. Additionally, I have a max sequence length of 512 and pad to that max length, and I use the Adam optimizer with an epsilon value of $1e-8$ to avoid division by zero errors.

In terms of loss, I use sparse categorical cross entropy as is standard for binary class classification tasks.

$$L = \frac{-1}{n}\sum_{i=1}^{n} y_i log\hat{y}_i + (1-y_i)log(1-\hat{y}_i)$$

I run this process for 4 languages (English, Slovenian, Croatian, and Polish). For my base experiments, I use 20,000 tweets from each language, breaking this down to a 15,000-tweet training set size and a 5,000-tweet validation set size. For each language, I freeze 0, 2, 4, 6, 8, 10, and 12 layers on separate runs. For later experiments that explore the benefit of adding more data in the low-resource language, I repeat the process, but with more tweets. The results are compiled in section 5.4.

5.4 Results

20,000 Tweets

Figure 2 highlights the validation results from running the BERT-base model with different numbers of frozen layers with 20,000 tweets from each of the four languages. Training accuracies can be found in Appendix A.

As expected, English has the highest accuracies across all frozen layers, peaking at 93.28% accuracy with 0 frozen layers. None of the other languages can beat this benchmark, though we don't expect it to without fine-tuning, prompting, or some other forms of meta-data use. However, the model evidently is still able to make some sense of tweets in other languages. This probably

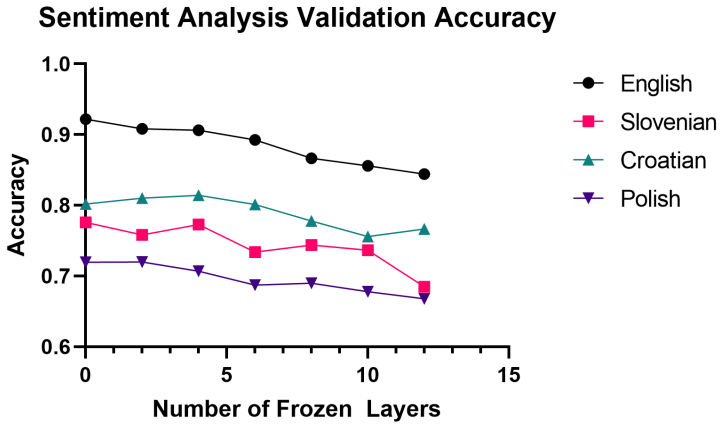

Figure 2: Validation Accuracies for 20,000 Tweets Across 4 Languages

indicates that a large amount of BERT's sentiment analysis stems from the structure of the text, rather than the actual words.

Slovenian and Croatian have better performance than Polish, which may be due to a larger language similarity between these two languages (See section 5.1 for discussion on Slavic Language Similarities). Croatian exhibits the best validation performance of all the Slavic languages, reaching approximately 82% accuracy at 4 frozen layers. Slovenian's peak performance also occurs at 4 frozen layers (about 79%). Polish's best performance is at 0 frozen layers with about 72% accuracy.

An interesting note is that for the three Slavic languages, performance dips most significantly after 4 frozen layers, but until that point, accuracy stays relatively constant. This may imply that BERT learns structural information in the preliminary layers of the model and textual/word meanings in the later layers. These results may show that if using transfer learning between languages, freezing four layers is ideal for saving compute, avoiding overfitting, and achieving good accuracy.

The worst performance occur at 10 and 12 frozen layers, which makes sense as the model is not trained on much of the low-resource language with just 2 or 0 layers. It still does better than guessing (50% Accuracy), which was quite interesting. One can infer that it does apply some linguistic understanding even when the target language is unknown, though this might be more of an indication on the similarity of English to these Slavic languages than anything about the model itself.

To contradict the findings in (8), I do not observe high variability between runs with the same settings. The author of this blog post claims that BERT is known for instability due to vanishing gradient problems, but I did not encounter such variable findings in a small experiment I ran (See *Table 3*). I ran my 20,000 Tweet experiment in Slovenian with 0 Frozen layers 5 times and was able to achieve consistent results. Due to time constraints, I did not complete these experiments with more runs, more languages, or more tweets, but it did seem worth mentioning, though I do not claim these findings to be sufficient to draw concrete conclusions.

More Tweets

For some languages, the CLARIN datasets provided more tweets. As discussed in section 5.1, Slovenian had about 40,000 tweets, Croatian had about 60,000, and Polish had nearly 100,000

Run	Train Accuracy	Validation Accuracy
1	77.38	77.58
2	77.12	77.43
3	77.80	77.89
4	76.87	77.32
5	77.90	78.18

Table 3: Running Slovenian with 0 Frozen Layers and 20,000 Tweets

tweets (after preprocessing). Thanks to this and some extra time, I was able to run my model with these larger amounts of tweets. The figures can be found in Appendix C.

Of course, increasing the number of tweets does increase the accuracy substantially.

Slovenian had about 40,000 tweets that I could test with (about double the original number of tweets I tested with). In 0-4 frozen layers, accuracy increased to about 82-83%, which shows an increase of about 5 accuracy points. More impressively, with a completely frozen model, accuracy increased to about 77% (an increase of nearly 13 points)! Slovenian had the smallest improvement of the three Slavic languages, but this is to be expected, given that Slovenian has the smallest increase in number of tweets.

Croatian also exhibited quite a jump in accuracy of $+10(79\% \rightarrow 89\%)$ for 0 frozen layers. This language showed more consistency and smoother performance increases. See Figure 6.

Polish, which had the lowest accuracies in the 20,000 tweet run, improved by nearly 13 accuracy points in a 0 frozen layer paradigm, though having 100,000 labeled tweets could be expensive to attain, further solidifying that BERT performs best with languages most related to the language it was trained on (in this case, English).

6 Comparison to Galeshchuk et. al

Galeshchuk et. al (1) achieved peak accuracies of 78.46% in Polish with a Gradient Boosting Tree, 76.10% accuracy for Slovenian with a Gradient Boosting Tree, and 86.32% in Croatian with a Random Forest. As a reminder, they first translate the low-resource language into English and then test their models.

With 0-4 frozen layers, my model beats all these results, though my Polish SA only beats Galeshchuk's findings with the larger amount of tweets. To be fair, this could be attributed to the fact that Galeshchuk et. al do not use deep learning models at all, and rely on more standard machine learning techniques, but there most likely still is semantic information lost when a low-resource language is translated to English first. There are many differences between our model designs that could lead to the gap in performance, but my studies have shown that transfer learning may be a viable technique to sentiment analysis on low-resource languages and may mitigate some semantics lost in translation.

7 Limitations and Future Work

Primarly, the drawback of using BERT in particular for transfer learning is that it seems to benefit from language similarity. For example, Slovenian and Croatian appear to be more similar to English than Polish is, which seems to explain why Polish does not perform as well using this cross-lingual transfer.

Another point is that in order to achieve very high accuracies, you still need a large amount of labeled data. This may be unrealistic to achieve, and if you have a large amount of labeled data, one may argue simply designing a new model to train one the low-resource language from scratch.

In terms of data, it is unclear whether some of the things eliminated in preprocessing affect the analysis of sentiment in low-resource languages. As mentioned in section 5.1, for English SA, it is traditional to remove stop words, but it is not always clear what the stop words or stop-word-equivalents are in other languages. Due to this, it is more difficult to build a generalizable model that handles preprocessing.

Lastly, it is always difficult to evaluate how well the model is doing when you are relying on the validity of the labeled data. For example, it would be easy for any human to verify whether labels are correct for a dog/cat image classifier, but in terms of multiple languages, it is hard for one person to verify that the sentiment labels are correct, especially if that person does not speak the languages being tested.

In the future, it would be nice to explore transfer learning with other models like ELECTRA or mBERT–maybe even GPT-3/Chat-GPT. But in dealing with this model, I would have liked to analyze which tweets the model is misclassifying and seeing if there is a trend. I did a preliminary glance through and found that tweets that were emoji-heavy before pre-processing are often misclassified, but I do not have any quantitative data to back up this claim.

8 Conclusion

Transfer Learning is a viable way to analyze the sentiment of text in low-resource languages. It may not be sufficient to use out-of-the box methods in order to achieve state-of-the-art results, but the BERT in particular appears to still grasp some semantics and structural sentiment from low-resource language tasks. It also learns efficiently with 4 frozen layers. BERT benefits from using languages similar to that it was originally trained on, which may support the claim that the initial layers of BERT learn more about the structure of a text than the wording/context. Additionally, we see increased performance with more data, as expected. I theorize that transfer learning in tandem with some use of meta-data may boost performance, though it is unclear whether it will be able to compete with bigger models that have access to larger compute resources that will inevitably be created in the future.

References

[1] Galeshchuk, Jourdan, Qiu. *Sentiment Analysis for Multilingual Corpora*, Aclanthology. code 2019

[2] Moberg, John. *A Deep Dive into Multilingual NLP Models*, Peltarion. 23 Feb. 2020

[3] Kanclerz, Milkowski, Kocon. *Cross-lingual deep neural transfer learning in sentiment analysis*, ScienceDirect. 2020.

[4] Devlin, et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*, arxiv. 11 Oct 2018.

[5] Clark et al. *ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators*, arxiv. 23 Mar 2020.

[6] Lample, Conneau. *Cross-lingual Languaage Model Pretraining*, arxiv. 22 Jan 2019.

[7] Pires, Schlinger, Garette. *How multilingual is Multilingal BERT?*, arxiv. 4 Jun 2019.

[8] B. Raphael *How many layers of my BERT model should I freeze?* Raphael B.'s Blog 10 May 2021.

[9] Tran, Chris. *Fine-Tuning BERT for Sentiment Analysis* SkimAI

[10] Radford, et. al *Language Models are Unsupervised Multitask Learners* Paper 2018

9 Appendix

Appendix A

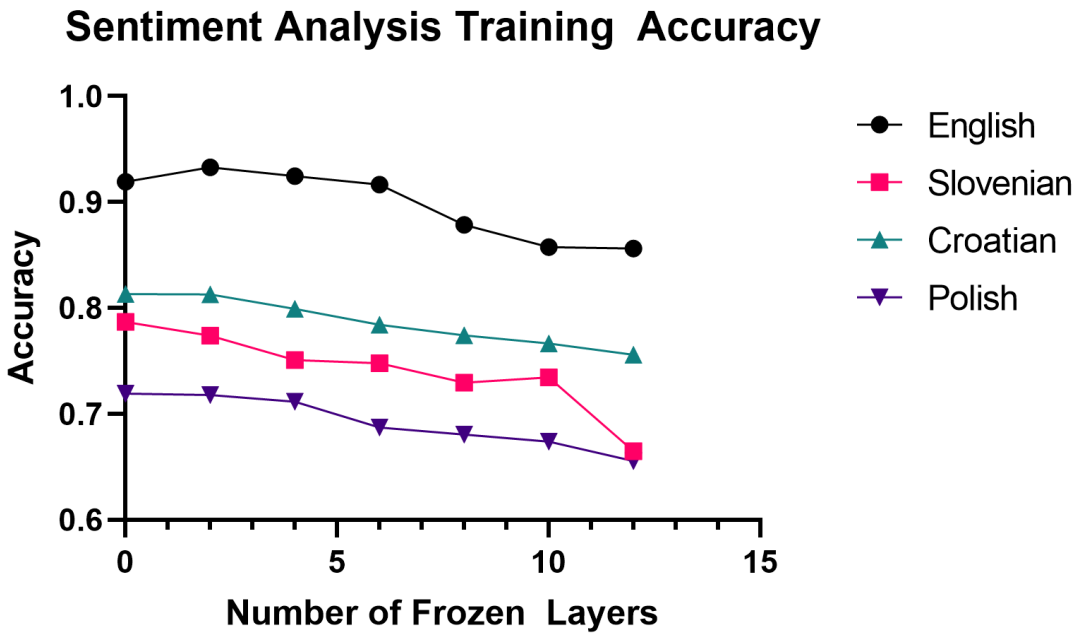

Figure 3: Training Accuracies across all Languages with 20,000 Tweets

Appendix B

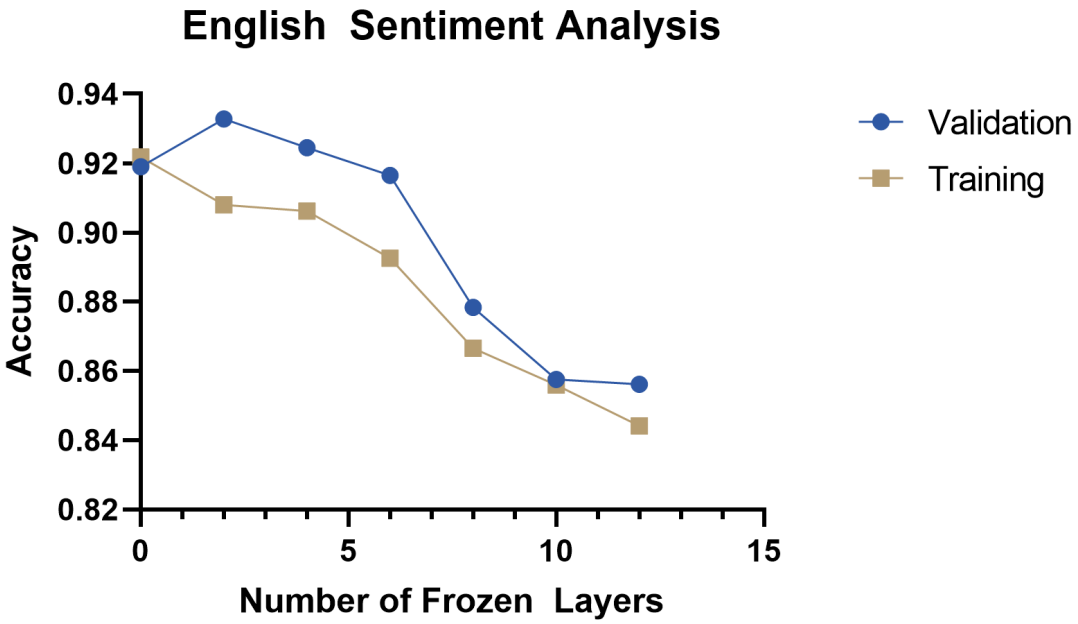

Figure 4: English SA with 20,000 Tweets

Appendix C

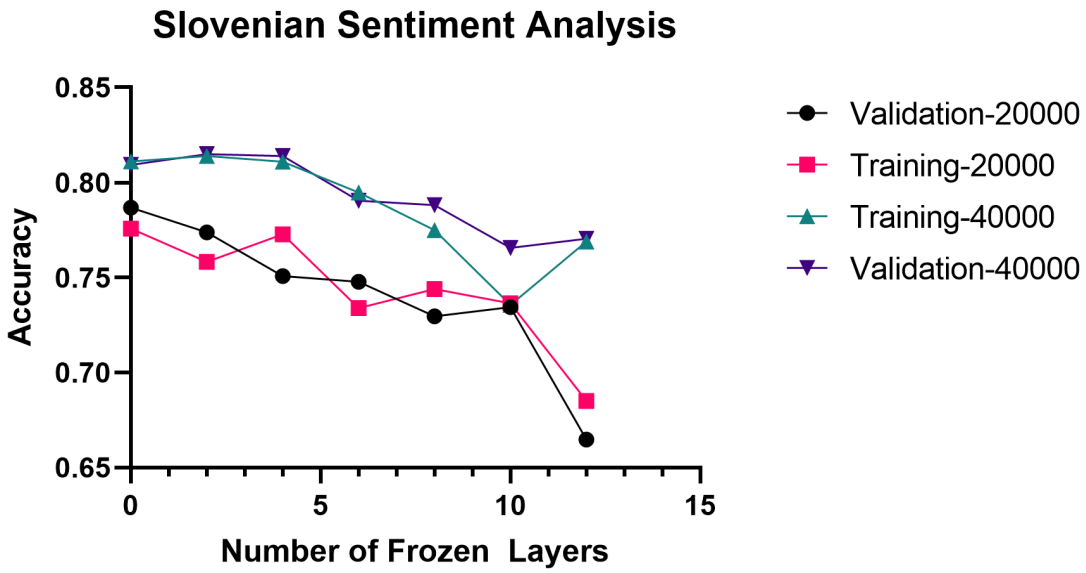

Figure 5: Slovenian SA

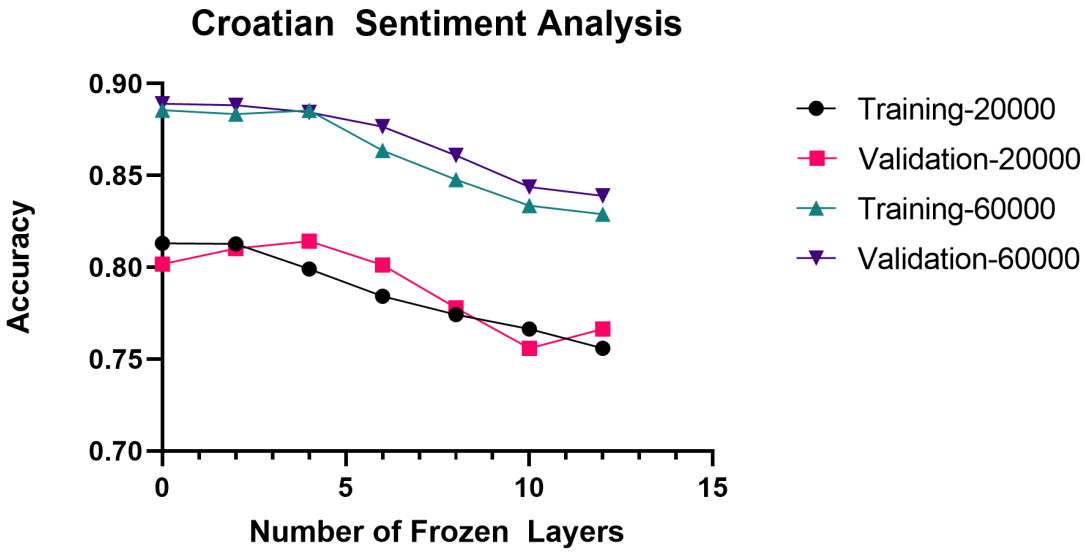

Figure 6: Croatian SA

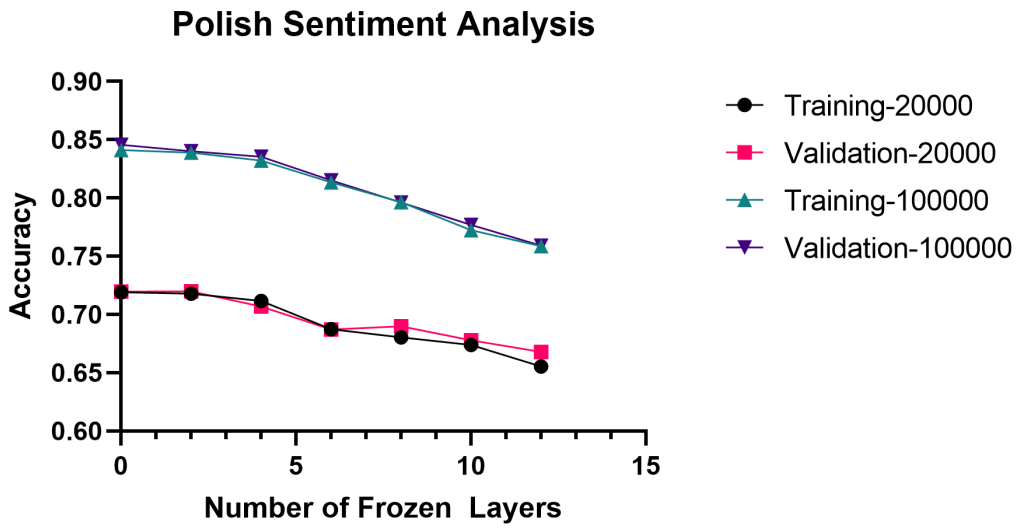

Figure 7: Polish SA

Appendix D

Language	# of Frozen Layers	# Trainable Params	# Untrainable Params	20000 Train	20000 Loss	20000 Val	20000 Loss2	Training Acc	Training Loss	Val Acc	Val Loss	Time for Full
English	0	109,483,778	0	0.919	0.21804	0.9218	0.2129	0.919	0.21804	0.9218	0.2129	50m
English	**2**	**95,308,034**	**14,174,744**	**0.9328**	**0.1844**	**0.908**	**0.2766**	**0.9328**	**0.1844**	**0.908**	**0.2766**	1h
English	4	81,132,290	28,351,488	0.9245	0.2024	0.9062	0.2892	0.9245	0.2024	0.9062	0.2892	50m
English	6	66,956,546	42,527,232	0.9165	0.2229	0.8926	0.3198	0.9165	0.2229	0.8926	0.3198	50m
English	8	52,780,802	56,702,976	0.8784	0.2975	0.8666	0.3149	0.8784	0.2975	0.8666	0.3149	1h
English	10	38,605,058	70,878,720	0.8576	0.3319	0.856	0.3289	0.8576	0.3319	0.856	0.3289	50m
English	12	24,429,314	85,054,464	0.8562	0.3564	0.8442	0.3467	0.8562	0.3564	0.8442	0.3467	40m
Slovenian	0	109,483,778	0	0.7867	0.4682	0.7758	0.4927	0.8111	0.4273	0.8093	0.4269	2h30m
Slovenian	2	95,308,034	14,174,744	0.7738	0.4847	0.7582	0.5537	0.8139	0.4197	0.815	0.4285	2h15m
Slovenian	4	81,132,290	28,351,488	0.7507	0.5087	0.7728	0.4874	0.8109	0.4217	0.814	0.4157	2h15m
Slovenian	6	66,956,546	42,527,232	0.7477	0.5134	0.734	0.5204	0.7948	0.4437	0.7904	0.4496	2h
Slovenian	8	52,780,802	56,702,976	0.7296	0.5338	0.744	0.5226	0.7749	0.4743	0.7881	0.449	2h
Slovenian	10	38,605,058	70,878,720	0.7345	0.5324	0.7364	0.5239	0.7356	0.5277	0.7656	0.4894	2h
Slovenian	12	24,429,314	85,054,464	0.6648	0.6133	0.685	0.5844	0.769	0.4836	0.7704	0.4927	1h30m
Croatian	0	109,483,778	0	0.8131	0.4287	0.8018	0.4318	0.8856	0.2876	0.889	0.298	3h30m
Croatian	2	95,308,034	14,174,744	0.8128	0.429	0.8102	0.426	0.8834	0.2894	0.8882	0.296	3h30m
Croatian	4	81,132,290	28,351,488	0.7991	0.4263	0.8142	0.4263	0.8854	0.279	0.8845	0.287	3h30m
Croatian	6	66,956,546	42,527,232	0.7843	0.427	0.8012	0.4134	0.8636	0.3276	0.8766	0.3087	3h30m
Croatian	8	52,780,802	56,702,976	0.7743	0.487	0.778	0.4998	0.8478	0.3565	0.861	0.3335	3h30m
Croatian	10	38,605,058	70,878,720	0.7665	0.506	0.756	0.496	0.8336	0.3772	0.8437	0.3619	3h20m
Croatian	12	24,429,314	85,054,464	0.7559	0.5068	0.7664	0.492	0.8289	0.3954	0.8389	0.3785	3h30m
Polish	0	109,483,778	0	0.7191	0.5539	0.7196	0.5642	0.8412	0.3661	0.8456	0.357	4h
Polish	2	95,308,034	14,174,744	0.7178	0.5481	0.7198	0.5436	0.8388	0.372	0.8402	0.3668	4h
Polish	4	81,132,290	28,351,488	0.7116	0.5619	0.707	0.5612	0.8321	0.389	0.8354	0.3728	4h
Polish	6	66,956,546	42,527,232	0.6873	0.5858	0.6872	0.5872	0.8134	0.3977	0.8152	0.3946	4h
Polish	8	52,780,802	56,702,976	0.6805	0.5897	0.69	0.5806	0.7964	0.4228	0.796	0.423	4h
Polish	10	38,605,058	70,878,720	0.6738	0.6003	0.678	0.5931	0.7723	0.4786	0.7768	0.4794	4h
Polish	12	24,429,314	85,054,464	0.6554	0.6212	0.6678	0.603	0.7587	0.5672	0.7592	0.5548	4h

Figure 8: Full Experiment Data Collected in One Table

All columns with 20000 prefix indicate results from the 20,000 Tweet Experiments. The last 5 columns all relate to the larger data experiments.