LING 573 Project Report: D4

Levon Haroutunian*, **Yi-Chien Lin***, **Bridget Tyree***, **David Yi***
Department of Linguistics
University of Washington
{levonh, yichlin, btyree, davidyi6}@uw.edu

Abstract

This project considers the task of offensive language detection in English and Greek. To classify tweets in different languages, we utilize cross-lingual pretrained models and fine-tune each model on either OLID: the English data set from OffensEval 2019 (Zampieri et al., 2019b), OGTD: the Greek dataset from OffensEval 2020 (Zampieri et al., 2020), or a combination of both datasets, to classify tweets as offensive or non-offensive. For the 2019 English data (our primary task), we find that fine-tuning XLM-RoBERTa-large (Conneau et al., 2019a) on OLID achieves the best performance. On the other hand, the best performance on the 2020 English data (primary task) and 2020 Greek data (adaptation task) are both obtained by ensemble models that are fine-tuned on varying combinations of the Greek and English datasets.

1 Introduction

The detection of offensive or abusive language has been a rich area of research in NLP, which is overviewed in Waseem et al. (2017). Automated abuse detection can be a helpful tool for content moderation, though it should be noted that such a tool is only one component of a system that can adequately address and prevent toxic behavior (Jurgens et al., 2019).

Our first and second systems focus solely on the primary task, which identifies offensive English tweets. On the other hand, our current system tackles both the primary and adaptation tasks – identifying offensive tweets in English and Greek. In our initial primary task system, we used static GloVe word embeddings (Pennington et al., 2014) trained on Twitter data[1] to form a representation for each tweet, and fed those representations into a logistic regression classifier. This system achieves a Macro F1 score of around 0.598 on the development dataset.

For the second system, we used the same word embeddings but fed them into a BiLSTM classifier instead. We also made significant additions to preprocessing methods including negation detection and emoji representations. We experimented with many different combinations of potential improvements, and our best system achieves a Macro F1 score of roughly 0.732 for the development dataset.

For our current system, we apply language specific pre-processing methods for English and Greek. We then use various "BERT-style" tokenizers and encoders (e.g. mBERT, RoBERTa, XLM-RoBERTa) (Devlin et al., 2019) and fine-tune a randomly initialized classification layer on the relevant dataset(s). Our current system achieves a Macro F1 score of around 0.809 for the 2019 English test data, 0.9202 for the 2020 English test data, and 0.7015 for the 2020 Greek test data.

The rest of this paper is organized as follows. Section 2 details the task description, including both the primary task of detecting offensive language in English tweets and the secondary task of detecting offensive language in Greek. Sections 3 and 4 provide a system overview and describe our approach for the current system in detail. Section 5 provides the results from our current system, which are analyzed in Section 6. Finally, Section 7 provides a conclusion.

* Equal contribution, sorted in alphabetical order.

[1] https://nlp.stanford.edu/projects/glove/

2 Task Description

The tasks in this project are based on OffensEval 2019[2] (Zampieri et al., 2019b) and OffensEval 2020[3] (Zampieri et al., 2020), both of which were SemEval shared tasks. Description regarding the primary task and adaptation task are presented respectively in the following subsections.

2.1 Primary Task

For our primary task, we tackle subtask A of OffensEval 2019 – identifying offensive language in English (i.e. determining whether a given statement is offensive). More specifically, the affect type of this task is interpersonal stance and the target is sentiment. The identification of offensive language is obtained by implementing binary text classification on tweets.

We use the training set from subtask A of OffensEval 2019: the Offensive Language Identification Dataset (OLID)[4] (Zampieri et al., 2019a). OLID contains annotated tweets that are crowdsourced based on a three-layer annotation scheme. This annotation scheme enables developers to use the same dataset for different subtasks. For evaluation, we evaluate our system on the test sets from both subtask A of OffensEval 2019 and that of OffensEval 2020 (Zampieri et al., 2020). The test set from OffensEval 2019 is a part of the OLID dataset while the OffensEval 2020 is a part of the Semi-Supervised Offensive Language Identification Dataset (SOLID) [5] (Rosenthal et al., 2020). SOLID was annotated using the same annotation scheme as OLID but was labelled in a semi-supervised manner.

Our training data for the primary task contains 14,100 English tweets, in which 4,640 (33%) are annotated as offensive (OFF) and 9,460 (67%) are labeled as not offensive (NOT). Since no distinguished development set is provided, we designate 10% of the training data as development set. For evaluation, we examine the performance of our system respectively on the 2019 English and 2020 English test sets. The 2019 English test set contains 240 (28%) OFF tweets and 620 (72%) NOT tweets while the 2020 English test set contains 1,090 (28%) OFF tweets and 2,807 (72%) NOT tweets. We compute the Macro F1 score as our evaluation metric, which is the official evaluation metric of OffensEval 2019 (Zampieri et al., 2019b).

2.2 Adaptation Task

For our adaptation task, we extend our system to classify tweets in Greek, which is one of the languages included in OffensEval 2020. We use the same training and test sets as the ones used in the subtask A of OffensEval 2020: Offensive Greek Tweet Dataset (OGTD)[6]. OGTD is a dataset annotated and preprocessed using the same guidelines as OLID (Zampieri et al., 2020).

The training data for the adaptation task contains 8,743 Greek tweets, in which 2,486 (28%) are annotated as OFF and 6,257 (72%) are annotated as NOT. Similar to the dataset for our primary task, since no distinguished development set is provided, we designate 10% of the training data as development set. The test data for the adaptation task contains 1,544 Greek tweets, consisting of 425 (28%) OFF tweets and 1,119 (72%) NOT tweets. The evaluation metric is same as the one used for our primary task – calculating the Macro F1 scores of the classification results.

3 System Overview

This section provides a high-level overview of our model fine-tuning approach for both the primary and adaptation tasks in D4.

3.1 Data Preprocessing

For the primary system, we implement preprocessing methods that handle capitalization, punctuation, and hashtags. Unlike in previous deliverables, we do not incorporate negation detection in preprocessing since BERT-style embeddings are contextual (Devlin et al., 2019) and should inherently be able to capture negation information. In the adaptation system, we additionally explore Greek-specific preprocessing techniques including diacritic removal,

[2] https://sites.google.com/site/offensevalsharedtask/offenseval2019

[3] https://sites.google.com/site/offensevalsharedtask/results-and-paper-submission

[4] https://sites.google.com/site/offensevalsharedtask/olid

[5] https://sites.google.com/site/offensevalsharedtask/solid

[6] https://zpitenis.com/resources/ogtd/

conversion of unicode characters to ASCII, and lemmatization.

3.2 Fine-tuning Pre-trained Models

In our final system, we design a multilingual fine-tuning pipeline that is used for both the primary and adaptation tasks. Here, we leverage large pre-trained language models such as RoBERTa (Liu et al., 2019) and its multilingual variation XLM-RoBERTa (Conneau et al., 2019b) to tokenize and encode tweets, and additionally fine-tune a randomly initialized classification layer on the relevant dataset(s) using HuggingFace[7] to output binary predictions. To further improve our systems, we explore ensembling techniques including ensemble averaging and classifier stacking.

3.3 System Runtime

Table 1 shows the approximate training and prediction runtimes for our D4 condor script for the adaptation task, which fine-tunes XLM-RoBERTa-large on the Greek data (OGTD) and generates predictions. The training portion is commented out in the D4 bash script because fine-tuning XLM-RoBERTa-large on Patas results in memory errors.

4 Approach

This section presents detailed descriptions of the approaches used in our improved system. First we present the preprocessing methods we consider for our improved system. Then, we provide a description of the classification methods we use for our improved system.

4.1 Data Preprocessing

In our improved system, with the aim of reducing out of vocabulary (OOV) tokens, we apply the basic preprocessing methods to both English and Greek datasets. For Greek data, we also implement additional Greek-specific preprocessing methods. Details of each method are as follows.

4.1.1 Basic Preprocessing

Because tweets are extremely noisy and often contain typos, irregular text, URLs, emojis, and tags, we find simple preprocessing methods to be extremely beneficial. The four "basic" preprocessing methods we applied to the raw tweets include converting to lowercase, splitting words from punctuation, removing apostrophes from contractions, and removing all hashtag symbols. These methods were chosen with the goal of minimizing the number of OOV tokens in any pretrained embeddings we used in our experiments. Hence, the following raw tweet would be converted in the following way:

1. By the way, I don't agree with your argument. #livid

2. by the way , i dont agree with your argument . livid

4.1.2 Greek-specific Preprocessing

For Greek-specific pre-processing methods, we focus on dealing with diacritics and lemmatization. We implemented two methods to deal with diacritics – removing diacritics and converting unicode data into ASCII characters. For example, given a Greek character $\grave{\alpha}$, the former method converts this character to α whereas the latter method converts the character to $\boldsymbol{a}$.

For lemmatization, we use the spaCy library (Honnibal and Montani, 2017) to lemmatize Greek tweets. Lemmatization refers to converting words to their base forms. For example, in English, lemmatizing inflected words such as *walks*, *walked*, and *walking* refers to converting them into their base form *walk*. Following is an example of lemmatizing a Greek tweet:

1. Πώς τα πάνε οι γυναίκες με το αυτοκίνητο

2. πώς ο πάνες ο γυναίκα με ο αυτοκίνητο

4.2 Fine-tuning Approach

In the previous deliverables, we created word vector representations from sequence tokens based on English GloVe embeddings (Pennington et al., 2014), with additional fine-tuning on the OLID dataset for the BiLSTM that we built in D3. However, since GloVe embeddings are entirely trained on English corpora, this approach cannot be directly extended for the adaptation task of predicting Greek tweets.

Instead, we explore a fine-tuning approach which uses monolingual and cross-lingual pretrained language models including BERT (Devlin et al., 2019) and its variants to create contextual representations for tweets, and add a

[7] https://huggingface.co/docs/transformers/training

Section	**Approximate Runtime (in minutes)**
Training Loop (n=7869, epochs=5)	290
Predictions (n=1545)	2
Total Runtime	292

Table 1: System runtimes for the end-to-end fine-tuning and inference pipeline for XLM-RoBERTa-base on the Greek (OGTD) dataset

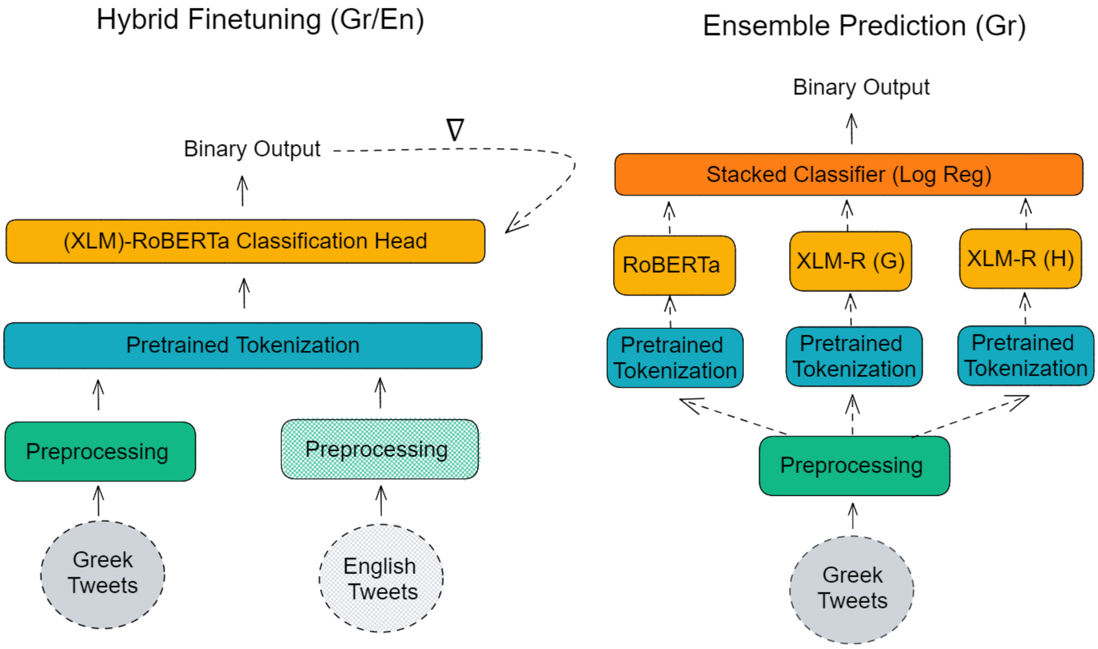

Figure 1: System Architecture Example: Hybrid fine-tuning and Greek Inference

fully-connected classification layer at the top of each model which is fine-tuned on the OLID English dataset and/or the OGTD Greek dataset. To tokenize the tweets, we use the innate tokenizer for each model which is a byte-level BPE for RoBERTa (Liu et al., 2019), a SentencePiece encoder (Kudo and Richardson, 2018) for XLM-RoBERTa (Conneau et al., 2019b), and a WordPiece tokenizer for multilingual BERT (mBERT) (Devlin et al., 2019).

4.2.1 Model Selection and Hypertuning

We perform several fine-tuning experiments with different configurations that explores several parameters, including the choice of pre-trained model, which data to fine-tune the classification layer on, and a select number of hyperparameters. For each task, we experiment with both monolingual fine-tuning (e.g. training only on OGTD for the adaptation task) and hybrid fine-tuning (i.e. training with both OLID and OGTD for the adaptation task). For model selection, we experiment with several pre-trained models including RoBERTa (base and large), which is a monolingual English model, as well as XLM-RoBERTa (base and large) and mBERT ((BERT-BASE-MULTILINGUAL-CASED) which are both cross-lingual models.

In terms of model hyperparameters, we search over *batch_size* $\in \{2, 4\}$, $LR \in \{1 \times 10^{-6}, 5 \times 10^{-6}\}$, and *epochs* $\in \{5, 10\}$ with the random seed set to 11 and chose the best performing combination for each model.[8]

4.2.2 Ensembling

To address potential shortcomings of each individual model, we also explore relatively simple ensembling methods including “Ensemble Averaging” and “Stacked Classification”. For the Ensemble Averaging model, we compute the

[8] We couldn’t run certain combinations (i.e. XLM-RoBERTa Large with batch_size=4) because of memory constraints on Patas and Google Colab

predicted offensive probability as:

$$P(OFF) = \frac{\sum_{i=0}^{k} P(OFF)_i}{k}$$

for an ensemble of k models where $P(OFF)_i$ is the predicted offensive probability of model i.

For the "Stacked Classification" approach, we concatenate the predicted probabilities of each model into a feature vector for each tweet and train a "combiner" Logistic Regression model over the validation data which predicts the final output. More formally,

$$P(OFF) = \sigma(W\vec{v} + \vec{b})$$

for an ensemble of k models where σ is the standard logistic function, $\vec{v} \in R^{2\times k}$, and W and b are the learned weights of the Logistic Regression model.

5 Results

This section presents the evaluation results of the primary and adaptation tasks. The primary task is evaluated respectively on the English development set from OffensEval 2019, the English test set from OffensEval 2019, and the English test set from OffensEval 2020. Note that the datasets from OffensEval 2019 are parts of the OLID dataset while the datasets from OffensEval 2020 are parts of the SOLID dataset. For the adaptaion task, we evaluated our system respectively on the development and test sets from OffensEval 2020 (i.e. OGTD). Details of the results evaluated on each dataset are as follows.

5.1 Primary Task Results 2019

For the evaluation, since the OLID dataset does not contain a distinguished development dataset, we designated 10% of our training data to be our development dataset.

Table 2 shows the performance of our D2, D3 and D4 models on the development and test sets against the baseline models (**ALL OFF** and **ALL NOT**) as well as the top scoring model for the task (**BERT-based-uncased default params**). As shown in Table 2, our best scoring model, which is a XLM-Roberta-Large model fine-tuned on OLID, achieves a Macro F1 score of 0.809 which is only .02 lower than that of the top scoring team (Zampieri et al., 2019b).

Tables 3 and 4 show that while we score relatively well on our Macro F1 score, we still struggle with class imbalance. Our top scoring model correctly classified non-offensive tweets roughly 93% of the time for the test set and roughly 85% of the time for the development set. However, our model only correctly identifies offensive tweets roughly 67% of the time for the test set and roughly 73% of the time for the development set. While additional efforts to fix this class imbalance through oversampling and undersampling methods did not work, this is actually a substantial improvement from our D3 model which only classified the offensive tweets correctly roughly 68% of the time on the development set.

5.2 Primary Task Results 2020

Table 5 shows the results of our D4 models on the 2020 primary task against the majority baseline and the top scoring model for the task (**UHH-LT**) (Zampieri et al., 2020). As shown in Table 5, our highest scoring model, which is a stacked ensemble containing RoBERTa (fine-tuned on Greek and English data), XLM-RoBERTa-base (fine-tuned on English data), and XLM-RoBERTa-Large (fine-tuned on English data) achieves a Macro F1 score of 0.9202 which is only 0.002 lower than the highest scoring model which achieved a Macro F1 score of 0.9204.

Table 6 shows that for the 2020 test data, our model classifies both offensive and non-offensive tweets with over 90% accuracy. It classifies non-offensive tweets correctly roughly 92% of the time and offensive tweets roughly 98% of the time.

5.3 Adaptation Task Results

Similar to the OLID dataset, since the OGTD dataset does not contain a distinguished development dataset, we designate 10% of our training data to be our development dataset.

Table 7 presents the Macro F1 score of a XLM-RoBERTa-Large model fine-tuned on Greek data using different combinations of preprocessing methods. As shown in Table 7, the model fine-tuned on data using only basic preprocessing method obtains the best Macro F1 score for test data.

Primary Task 2019		
Method	Dev Macro F1	Test Macro F1
ALL OFF	N/A	0.220
ALL NOT	N/A	0.420
D2 Model - Log Regression + GloVe	0.598	N/A
D3 Model - BiLSTM + GloVe	0.723	N/A
D4 Model - RoBERTa (en/gr)	0.790	0.787
D4 Model - XLM-RoBERTa-Large (en)	0.782	**0.809**
D4 Averaging Ensemble	0.789	0.808
BERT-based-uncased default params	N/A	0.829

Table 2: Table showing our model's performance against the baseline models and models created by other teams for the 2019 primary task provided in (Zampieri et al., 2019b). Our highest score is denoted in **bold**.

Confusion Matrix 2019 Primary Test		
	Predicted Label	
True Label	Negative	Positive
Negative	574	80
Positive	46	160

Table 3: Confusion Matrix based on the results from the D4 primary model on the 2019 test set

Confusion Matrix 2019 Primary Dev		
	Predicted Label	
True Label	Negative	Positive
Negative	765	112
Positive	134	313

Table 4: Confusion Matrix based on the results from the D4 primary model on the 2019 development set

Table 8 shows the performance of the models implemented in D4 on the 2020 Greek subtask A against the majority baseline as well as the top scoring model for the task (**NLPDove**) (Zampieri et al., 2020). As shown in 8, our highest scoring model, a stacked ensemble model, achieves a 0.7051 Macro F1 score on the test data which is lower than the top scoring model of the task (NLPDove) but significantly higher than the majority baseline. For the stacked ensemble model, we concatenate predicted probabilities of two models – an XLM-RoBERTa-Large model fine-tuned solely on Greek data and an XLM-RoBERTa-Large model fine-tuned on both English and Greek data.

We present confusion matrices based on the results from the Greek Stacked Ensemble model on the development and test sets respectively in Table 9 and 10. Our top scoring model has a 86% accuracy on our development set and a 85% accuracy on our test set. More specifically, for the development set, our top scoring model correctly classifies the NOT tweets roughly 91% of the time and the OFF tweets roughly 73% of the time. For the test set, similar to the model's performance on the NOT tweets in the development set, our model correctly classifies NOT tweets 92% of the time. However, our model correctly classifies the OFF tweets in the test set only 46% of the time.

6 Discussion

In this section, we analyze the results described above. We first compare the performance of our models on development and test sets for each of the datasets (OLID, OGTD, and the SOLID test set). Then, we discuss the effectiveness of our adaptation approach. Finally, we present a qualitative error analyses on the English tweets.

6.1 Comparing Dev and Test Performance

In this subsection, we present the comparison between the development and test sets in our primary and adaptation tasks. Detailed descriptions are as follows.

6.1.1 Primary Task 2019 (OLID)

As shown in Table 2, two out of three of our models (i.e. XLM-RoBERTa-Large (en) and Averaging Ensemble) performed better on the OLID test set than they did the OLID development set. More notably, the models which scored highest on the test set scored lower on

Primary Task 2020		
Method	Dev Macro F1	Test Macro F1
Majority Baseline	N/A	0.4193
D4 Model - RoBERTA (en/greek)	0.790	0.9156
D4 Model - Stacked Ensemble	N/A	**0.9202**
UHH-LT	N/A	0.9204

Table 5: Table showing our model's performance against the baseline models and models created by other teams for the 2020 primary task provided in (Zampieri et al., 2020). Our highest score is denoted in **bold**.

Confusion Matrix Primary 2020 Test		
	Predicted Label	
True Label	Negative	Positive
Negative	2571	27
Positive	236	1053

Table 6: Confusion Matrix based on the results from the D4 primary model on the 2020 test set

the development set. We initially chose to move forward with the model RoBERTa (en/gr) because out of all the models we hyptertuned on the OLID development set, it scored the highest. However, we later ran our other models on the test set and realized that choosing the model that works best for the development set does not necessarily result in the best performance in generalizing unseen data.

6.1.2 Primary Task 2020 (SOLID)

Table 5 shows the results of running our models which scored highest on the 2019 OLID development set and the 2020 SOLID test set. Both models were finetuned on the OLID training set. Since the Stacked Ensemble model described in section 5.2 was trained on the OLID development set, we only show the Macro F1 score on the OLID development set for the RoBERTa (en/greek) model. The score for the OLID development set for that model was 0.790 and the score for the test set was 0.9156. We speculate that the surprising discrepancy in performance between the validation and test sets can be attributed to either the SOLID test set being very similar to the OLID training set, there being less imablance in the SOLID test set, or it could be that the SOLID tweets were chosen in a semi-supervised manner so they are easier to classify.

6.1.3 Adaptation Task (OGTD)

As shown in Table 8, there was a rather large difference between the Macro F1 score on the OGTD development set and the OGTD test set. Once again, we do not have the score for the development set for the Stacked Ensemble Model because it was trained on the development set. As for the XLM-RoBERTa-Large model, it achieves a Macro F1 score of 0.825 on the development set, but only a score of 0.6815 on the test set. Since no one on our team speaks Greek, we are unable to run an error analysis on our results, but we suspect either the valiadtion set is much easier than the test set or that the XLM-RoBERTa-Large model overfitted to the training set.

6.2 Effectiveness of Adaptation

As described in Section 4, we experiment with both monolingual and hybrid fine-tuning in order to compare the effectiveness of pre-training a model on both English and Greek or on only one language. Our motivation in conducting these experiments is to leverage the additional data offered by OLID to supplement the somewhat smaller OGTD. The two datasets overlap in their use of emojis and in the use of a few English words in OGTD, but otherwise have disjoint vocabularies.

Ultimately, hybrid pre-training did not offer much improvement over monolingual pre-training for Greek. Our best-performing single model on the Greek development set used an XLM-RoBERTa-large encoder with a classification layer trained exclusively on OGTD.

However, it does appear that our approach transfers well to Greek, even if the English fine-tuning does not. With only the basic pre-processing methods, we are able to attain reasonable performance on OGTD. The Greek-specific pre-processing methods do not seem

Method	Dev Macro F1	Test Macro F1
Basic	0.825	**0.682**
Basic + Diacritics Removal	0.829	0.672
Basic + Unicode Conversion	0.720	0.671
Basic + Lemmatization	0.815	0.673
Basic + Lemmatization + Diacritics Removal	0.822	0.658

Table 7: Table showing the development and test Macro F1 of XLM-RoBERTa-Large model trained on Greek data using different combinations of pre-processing methods. The pre-processing method with the highest test Macro F1 score is denoted in **bold**

Adaptation Task		
Method	Dev Macro F1	Test Macro F1
Majority Baseline	N/A	0.4202
D4 Model - Greek Stacked Ensemble	N/A	**0.7015**
D4 Model - XLM-RoBERTa-Large (greek)	0.825	0.6815
NLPDove	N/A	0.8522

Table 8: Table showing our model's performance against the baseline models and models created by other teams for the 2020 Greek adaptation task provided in (Zampieri et al., 2020). Our highest score is denoted in **bold**.

Confusion Matrix Adaptation 2020 Dev		
	Predicted Label	
True Label	Negative	Positive
Negative	573	57
Positive	65	180

Table 9: Confusion Matrix based on the results from the D4 adaptation model on the 2020 development set

Confusion Matrix Adaptation 2020 Test		
	Predicted Label	
True Label	Negative	Positive
Negative	1200	102
Positive	130	112

Table 10: Confusion Matrix based on the results from the D4 adaptation model on the 2020 test set

effective in improving our system performance. Improving our performance further would likely require additional resources, such as more data in Greek or in a language more closely related to Greek than English.

6.3 Error Analysis

Below is a qualitative analysis of the errors our model made on English test data. Since none of us are speakers of Greek, we are unable to perform similar analyses for the Greek test data. The examples below reflect errors that our model made on both the OLID test set (Zampieri et al., 2019b), which are labelled '2019', and on the 2020 OffensEval test set for English (Zampieri et al., 2020), which are labelled '2020'.

False Negatives Our model failed to detect some offensive tweets that were specifically targeted insults. Notably, many false negatives were tweets that include zero or few words that are explicitly insulting by themselves, such as the tweets below.

- ConfirmKavanaugh ✅ Kavanaugh SCOTUS POTUS45 PresidentTrump AmericaFirst MAGAKAG 🚫 Anti Trump liberal activist Accuser Ford makes accusation but not willing to be subject 2 B question ? Suffer the consequence of filing a false charge LockHerUp URL (2019)
- Ted always has a feel for the most appropriate public behavior . But, often, he is unable to control himself . GOPDysfunction URL (2019)
- @USER literally one of the worst comeback singles of all time, such a letdown (2020)
- men who voted yes are childish, sorry are you 17 ??Men, would you have a problem

if a girl said if shes not receiving head shes not giving head ? (2020)

- @USER do the two of you sit at home and practice your talking points ? Do you feel sad that your ratings suck ? (2020)
- @USER @USER Dehumanize ? He barely has a reflection of human (2020)

Our model also failed to detect offensive tweets with less common or abbreviated swear words:

- I shouldnt be up and hungry this early 😩 like stomach empty as a mf (2020)
- Im a whore for concerts/festivals, Ill literally go and see anyone even if idk their songs 🤷‍♀️ 🤷‍♀️ (2020)
- @USER He knows his days are numbered so going to cause as much trouble as possible now . Twat that he is !! (2020)

False Positives On the other hand, the presence of insulting words that are not directed at any entity caused our model to mis-classify non-offensive tweets as offensive:

- Conservatives are going to make hay out of Trudeaus answer but the truth is theres criminals everywhere and people who will commit heinous acts of violence . Whether theyre from Syria and have brownskin or are white Canadian born its tragic all around . URL (2019)
- @USER ...and how very irrelevant . But if they didnt find weed, they would have dug up an 8th grade picture of him wearing a hoodie and talk about what a thug" he is ." (2019)
- @USER @USER @USER @USER Evil ? Stupid, self serving but evil ? What a ridiculous thing to say (2020)
- @USER MY HEADER NOW IS THE STUPID BAG AOCO GAVE ME (2020)
- Ugly being this much up early Im literally brain dead (2020)

Additionally, our model incorrectly classified self-deprecating language (ie, the use of insulting words applied to the speaker) as offensive:

- In conclusion, im ugly and horrible and have the worst personality . (2020)
- Im fat and ugly): (2020)
- Girls call me ugly until they find out how much money I have ... then they call me poor and ugly (2020)

A third category of false positives is tweets that discuss sexuality without any explicit or profane language.

- NewVideo 🍆 🍑 💦 Toys, Anyone ? URL URL (2019)
- @USER @USER Its a sexual act right ? And all sexual acts except wife and slave are forbidden (2020)
- men moaning during sex is so sexyyyy . dirty talk is so sexyyyyyyy, i love it (2020)

These examples suggest that the classification of a tweet as offensive may be tied to the presence of words that tend to be offensive, particularly words that are typically used in insults. Our model does not seem to be able to detect more subtle insults, or non-offensive uses of insulting language. Additionally, it does not appear that our model is able to differentiate between insulting words directed at no specific entity, directed at the speaker, or directed at another entity.

7 Conclusion

This paper presents a model for offensive language detection in English and Greek. Our current system uses large pre-trained language models and a stacked ensemble classifier. For the primary task, our system attains a 0.809 Macro F1 score on the 2019 test set using a XLM-RoBERTa-Large model and a 0.9202 Macro F1 score on the 2020 test set using a stacked ensemble model. For the adaptation task, our system achieves a 0.7015 Macro F1 score using a stacked ensemble model.

For future improvement, focusing on detecting usage of offensive language (i.e. whether it is insulting) and differentiating the target of insulting words might a potential direction to work on for our primary task. Additionally, for the improvement of our adaptation task, including more Greek data or languages closely related to Greek than English might be helpful in improving the performance.

References

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019a. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Unsupervised cross-lingual representation learning at scale.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

David Jurgens, Libby Hemphill, and Eshwar Chandrasekharan. 2019. A just and comprehensive strategy for using NLP to address online abuse. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3658–3666, Florence, Italy. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing.

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

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Marcos Zampieri, and Preslav Nakov. 2020. A large-scale semi-supervised dataset for offensive language identification. *arXiv preprint arXiv:2004.14454*.

Zeerak Waseem, Thomas Davidson, Dana Warmsley, and Ingmar Weber. 2017. Understanding abuse: A typology of abusive language detection subtasks. In *Proceedings of the First Workshop on Abusive Language Online*, pages 78–84, Vancouver, BC, Canada. Association for Computational Linguistics.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019a. Predicting the Type and Target of Offensive Posts in Social Media. In *Proceedings of NAACL*.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019b. Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval). In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 75–86.

Marcos Zampieri, Preslav Nakov, Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Hamdy Mubarak, Leon Derczynski, Zeses Pitenis, and Çağrı Çöltekin. 2020. Semeval-2020 task 12: Multilingual offensive language identification in social media (offenseval 2020). *arXiv preprint arXiv:2006.07235*.