CHALMERS
UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

Machine Learning for Detecting Hate Speech in Low Resource Languages

Master's thesis in Computer science and engineering

DAVID RODRIGUEZ
DENITSA SAYNOVA

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2020

Master's thesis 2020

Machine Learning for Detecting Hate Speech in Low Resource Languages

DAVID RODRIGUEZ
DENITSA SAYNOVA

UNIVERSITY OF
GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2020

Machine Learning for Detecting Hate Speech in Low Resource Languages
DAVID RODRIGUEZ
DENITSA SAYNOVA

Supervisor: Richard Johansson, Department of Computer Science and Engineering
Examiner: K V S Prasad, Department of Computer Science and Engineering

Master's Thesis 2020
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2020

Machine Learning for Detecting Hate Speech in Low Resource Languages
DAVID RODRIGUEZ
DENITSA SAYNOVA
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Abstract

This work examines the role of both cross-lingual zero-shot learning and data augmentation in detecting hate speech online for low resource set-ups. The proposed solutions for situations where the amount of labeled data is scarce are to use a language with more resources during training or to create synthetic data points. Cross-lingual zero-shot results suggest some knowledge transfer is occurring. However, results seem greatly influenced by the specific training data set selected. This is further supported by cross-data set experimentation within the same language, where results were also found to fluctuate based on training data without the need for cross-lingual transfer. Meanwhile, data augmentation methods show an improvement, especially for low amounts of data. Furthermore, a detailed discussion on how the proposed data augmentation techniques impact the data is presented in this work.

Keywords: machine learning, natural language processing, BERT, cross-lingual zero-shot learning, data augmentation, hate speech, classification, Twitter

Acknowledgements

The authors would like to thank Richard Johansson for his support and feedback throughout this project and KVS Prasad for the many constructive discussions.

David Rodriguez and Denitsa Saynova, Gothenburg, June 2020

Contents

List of Figures xi

List of Tables xiii

1 Introduction 1

2 Theory 3

2.1 Machine Learning . . . 3

2.1.1 Unsupervised Learning . . . 4

2.1.2 Supervised Learning . . . 4

2.1.2.1 Data Annotation . . . 5

2.2 Feature Representations . . . 6

2.3 Neural Networks for Text Representations . . . 7

2.3.1 Neural Network Basics . . . 7

2.3.2 Transformers . . . 8

2.4 Transfer Learning . . . 10

2.4.1 BERT . . . 11

2.5 Cross-Lingual Representations . . . 12

2.5.1 Multilingual BERT . . . 12

2.6 Data Augmentation . . . 13

2.6.1 TF-IDF Synonym Replacement . . . 14

2.6.2 Word Dropout . . . 15

2.6.3 Back-Translation . . . 16

3 Methods 17

3.1 Hate Speech Data Sets . . . 17

3.1.1 Annotation Guidelines . . . 17

3.2 Baseline . . . 19

3.3 Cross-Lingual Zero-Shot Learning . . . 20

3.4 Effects of Training Set Size on Performance . . . 20

3.5 Data Augmentation . . . 20

3.5.1 TF-IDF Synonym Replacement . . . 21

3.5.2 Word Dropout . . . 21

3.5.3 Back-Translation . . . 21

3.6 Translation of English Training Data . . . 22

3.7 Evaluation . . . 22

3.8 Data Sets Analyses . . . 23

3.8.1 Clustering 23
3.8.1.1 Clustering for Hate Speech 24
3.8.1.2 Clustering English Training Data Sets 25
3.8.2 Classifying English Training Data Sets 25

4 Results 27
4.1 Baseline 27
4.2 Cross-Lingual Zero-Shot Learning 28
4.3 Effects of Training Set Size on Performance 30
4.4 Data Augmentation 30
4.4.1 TF-IDF Synonym Replacement 31
4.4.2 Word Dropout 31
4.4.3 Back-Translation 32
4.5 Translation of English Training Data 32
4.6 Data Set Analyses 33
4.6.1 Clustering 33
4.6.1.1 Clustering of HatEval Data Sets 33
4.6.1.2 Clustering English Training Data Sets 35
4.6.2 Classifying English Data Sets 37

5 Discussion 39
5.1 Cross-Lingual Zero-Shot Learning 39
5.2 Data Augmentation 40
5.2.1 TF-IDF Synonym Replacement 40
5.2.2 Word Dropout 41
5.2.3 Back-Translation 42
5.3 Translation of English Training Data 43
5.4 Data Set Analyses 44
5.4.1 Clustering of HatEval Data Sets 44
5.4.2 Clustering English Training Data Sets 44
5.5 Classifying English Data Sets 45

6 Conclusion 47

Glossary 49

A Appendix 1 I

List of Figures

2.1 Examples of less useful (left) and more useful (right) feature representations. 6
2.2 An example of a neural network. 7
2.3 An example of a recurrent neural network [29]. 8
2.4 An example of the transformer architecture where both on the left and on the right side, several of the components are stacked on top of each other. 9
2.5 An example of transfer learning. The main model is used to identify the breed of dogs. The knowledge gained could be useful for identifying breeds of cats, however, it is probably not useful for identifying healthy brains in MRI images. 11
2.6 BERT input representation, where the input is a sum of three levels of embedding [8]. 12
2.7 The original image. 13
2.8 Original image blurred. 13
2.9 Original image blurred too much. 13

3.1 Example of true positive and negative elements versus selected positives and negatives. The green and red points refer to positive and negative true labels, respectively. While the green and red backgrounds refer to what the model considers positive and negative, respectively. 22
3.2 Purity and inverse purity calculation for the three clusters above. Majority class for each cluster: cluster A: ▲, 5; cluster B: ⧫, 4; cluster C: ★, 3. Purity = $\frac{1}{20}(5+4+3) \approx 0.6$. Majority label for each class: ▲: cluster A, 5 ⧫: cluster B, 4 ★: cluster A, 4. Inverse purity = $\frac{1}{20}(5+4+4) \approx 0.65$. 24
3.3 Distribution of data compared to labels when perfect correlation is observed. 25

4.1 Spanish base grid search. Darkest green indicates highest accuracy value – 83.73. 27
4.2 English base grid search. Darkest green indicates highest accuracy value – 80.75. 27
4.3 Size dependence of performance for Spanish data. 30
4.4 Distribution of tweets within all available English data sets in each cluster. 36

4.5 Distribution of hate and non-hate tweets within all available English data set in each cluster. 36

List of Tables

2.1 Example of data that requires annotation. The tweet's text is the only thing that is automatically acquired. The label is obtained through manual annotation (neg = negative; pos = positive). 5
2.2 Example of multiple annotations. The label requires the decisions from each individual annotator – A1, A2 and A3 (neg = negative; pos = positive) and is based on the majority vote. 5
2.3 Values of TF-IDF scores for example given above. 14

3.1 A confusion matrix with two classes 26

4.1 Performance of the English Base (EN) and Spanish Base (ES). 28
4.2 Performance of English and Spanish base on other languages. Baselines with no cross-lingual learning are shown in gray. 28
4.3 Performance of different English models. Baselines on own test sets are shown in gray. WH refers to [26], F refers to [11] and HatEval refers to [3]. 29
4.4 Performance of Waseen & Hovy data set on other languages. Baseline with no cross-lingual learning are shown in gray. 29
4.5 Performance of Founta data set on other languages. Baseline with no cross-lingual learning are shown in gray. 30
4.6 Validation set TF-IDF results. Baselines where no DA is applied are shown in gray. 31
4.7 Results obtained using word dropout. Baselines where no DA is applied are shown in gray. 32
4.8 Results obtained using back-translation. Baselines where no DA is applied are shown in gray. 32
4.9 Translated vs cross-lingual zero-shot results. 33
4.10 Clustering results for hate speech in English training HatEval data. . 34
4.11 Clustering results for hate speech in Spanish training HatEval data. . 34
4.12 Clustering results for data source in English training data. 35
4.13 Clustering results for hate speech in English training data. 35
4.14 Data source classification results. Showing number of data points having a particular label vs the model prediction. Standard deviation on those counts are also shown in parentheses. 37

A.1 Performance of English Base (EN) and Spanish Base (ESP). I
A.2 Performance of English and Spanish base on other languages. I

A.3 Results obtained using different English Sets. I
A.4 Cross-lingual zero-shot results obtained using Founta. II
A.5 Cross-lingual zero-shot results obtained using Waseem & Hovy. II
A.6 Results obtained using word dropout. II
A.7 Results obtained using back-translation. II
A.8 Translated vs cross-lingual zero-shot results. II
A.9 Performance of English trained model for sentiment analysis. III

1

Introduction

Hate speech online is increasingly becoming a bigger problem in recent years [16] and has long been a conductor towards different types of hate crimes.[1] Even though there is no exact definition that is universally accepted, hate speech tends to be broadly defined as a type of communication where a person or a group of people gets denigrated based on race, gender or sexual orientation amongst other factors.

In recent years, most social media companies have been heavily criticized for their handling of hate speech within their platform.[2] As hate speech is expressed through language, applying natural language processing (NLP) techniques can be very beneficial for managing it. NLP is a set of computational tools for language understanding, which can be helpful for automating the detection of hate speech. Given the complexity of patterns in hate speech, it is not feasible to develop a purely procedural approach for detecting it. An important component of a solution for similar problems is to also apply machine learning (ML). ML is an approach to building models that change through experience, i.e. automatically learn patterns from example training data. Therefore the performance of ML models is tightly related to the amount and quality of the example data provided. In this case a classification model is used, which outputs a label given a piece of text as input.

Since the task at hand is concerned with classifying data, the examples needed for training the model have to be annotated. This is typically done by multiple people giving each data point a label and choosing the majority. Most of these available examples that are labeled tend to be in English. However, social media is multilingual and the question of moderating all media content is an important one. Therefore, the current work focuses on solutions for hate speech discovery online in languages where resources available are scarce.

Two solutions that are explored are data augmentation (DA) and cross-lingual zero-shot learning. The former is a method for producing additional data examples by transforming existing ones. Thus, increasing the amount of available examples without the need for additional collection or labeling. The latter trains a model using a language different than the one the model is applied to. For instance, the model is trained on English hate speech examples in order to detect Spanish hate speech. This leads to the possibility of using a more populous data set for training.

The development of these techniques comes with a unique set of challenges.

[1] https://www.nytimes.com/2018/10/15/technology/myanmar-facebook-genocide.html

[2] https://www.businessinsider.com/jack-dorsey-twitter-abuse-ted-2019-4?r=US&IR=T

One of the main issues is related to the specific details of the hate speech definition. This is because each data set uses slightly different guidelines when labeling the data. Secondly, as the labels are determined by a group of people and hate speech is complex, there tends to be low annotation agreement – i.e. the annotators tend to disagree on the correct label. Additionally, there might be cultural differences when perceiving hate speech which could impair the transferability of annotations from one culture to another.

Furthermore, some of the issues stem purely from the nature of the data. For example, social media text presents unique challenges since it has a distinct language structure like abbreviations, slang, typos, punctuation issues, etc. [9]

Since this is a complex task, there are several aspects that are not explored in full detail. One of these is the role of the similarity between the training and testing language in the cross-lingual zero-shot experiments. It can be assumed that languages from within the same family work better than two from different ones. However, testing the role of language proximity on performance is very time-consuming. Another issue with the current set-up is the quality of the annotations. Since these are manual, the quality could vary substantially between data sets. The effects of this issue are discussed in the current work. However, developing possible solutions is outside the scope of this project.

In order to best present the project, its components are discussed in the following order: Chapter 2 introduces the main theoretical concepts used; Chapter 3 outlines the experimental set-up; Chapter 4 presents the obtained results; Chapter 5 consists of a discussion of said results; Chapter 6 summarizes the project and lays out possible direction for further works.

2

Theory

In this chapter, some background of the theoretical framework needed is outlined. This includes the key ideas concerning both data annotation and machine learning. A deeper background is given in particular for transfer learning and the model used in this project. Finally, a description of the two methods used for dealing with scarce data – cross-lingual zero-shot learning and data augmentation – is presented.

2.1 Machine Learning

Machine learning is a field that utilizes data to build models for automatic decision making. A model can be seen as a black box that is fed values of an input x (called features) and produces an output decision y. In reality, that black box can have a variety of different architectures that are suited for different tasks and can be seen as complex functions mapping the input to the output. Initially, before data is introduced, the model can be seen as simply knowing the type of the function without knowing the values of the parameters.

There are two main types of machine learning, supervised and unsupervised learning (discussed in more detail in sections 2.1.1 and 2.1.2, respectively). The main distinction is the availability of a label (quantity of interest) for each data point example, i.e. having examples of correct (x, y) pairs. Supervised learning is carried out in cases when a label is available for the training examples, whereas unsupervised learning is applied to cases with no available label.

An example of a supervised task is classifying inputs into two groups, where the model used could be the perceptron. This uses a function of the form:

$$y(x) = \begin{cases} 1 & w.x + b > 0 \\ 0 & otherwise \end{cases}$$

As can be seen, the model determines the general relationship between the input x and the output y, without giving the specific values of the parameters w and b. These are found through a process called training, which consists of the following steps:

- passing the input through the model and obtaining a prediction
- comparing the prediction to the input's true label

- based on the difference between the prediction and the true label, modify the parameters

The specific modification the parameters go through is determined by the model's architecture. In the case of the perceptron if the prediction and the true label match, the weights are not changed. However when the prediction and the true label do not match, for a positive true label the weights are made more positive and for a negative true label the weights are made more negative. This is done to encourage the prediction towards the correct label.

In order to obtain more stable values for the parameters, this process needs to be repeated for all available data points several times. The amount of times the data is passed through the model is a hyper-parameter that is pre-set, i.e. not learned from the data. However, setting that number to a very high value on a small data set can lead to learning just the specific examples given. For this reason, a big enough data set with variability is needed in order to obtain a stable model.

Several best practices have a central role in machine learning in order to determine the quality of the model. One of the main ideas is the held-out set. A typical machine learning project would split the available data into a training data set and a testing data set. Training data is used for calculating model weights whereas testing data is used to calculate the performance. The reason performance is not calculated on training data is that that would typically overestimate the performance. Another practice applied during this process is repeated experimentation. Some models rely on random initial states, which means the resulting model can have some variability in their performance. In order to assess that, repeated training and testing of the model is performed to obtain a measurement of the stability of the result.

2.1.1 Unsupervised Learning

As previously mentioned, unsupervised machine learning is used when there is no known label in the data and, therefore, only patterns based on similarities between the inputs can be found. The main approach used for this type of learning is clustering. This consists of finding groups of data based on some predefined distance measurement.

The simplest approach to clustering is k-means. It is widely used, because its implementations are the most computationally efficient. At its core, k-means attempts to find k clusters, each one defined by a central point (called a centroid). A solution is found by changing the position of the centroid and trying to minimize the distance between the points and the centroids in each cluster.

2.1.2 Supervised Learning

The other type of machine learning is supervised learning, which is applied to data with available labels. The goal of supervised learning is to learn a mapping between the input and its label. There are two main types of supervised learning, regression (when the labels have continuous values) and classification (when the labels are categorical). For some types of data, these labels can be obtained during the data extraction process or calculated from the data itself. However, more complex tasks require other means for obtaining the labels, like data annotation.

2.1.2.1 Data Annotation

Tasks like image recognition and language understanding typically utilize annotation — i.e. a human (annotator) giving each data point a label. An example of a task needing annotation is sentiment analysis (i.e. identifying if a text is positive or negative). In a set of tweets for example, a typical way of detecting whether the tweet is positive or negative is to ask a human to look at the text and decide.

Tweet	**Label**
blagh class at 8 tomorrow	neg
Gonna catch sum rays on this glorious day!!!	pos
I had such a nice day. Too bad the rain comes in tomorrow at 5am	pos

Table 2.1: Example of data that requires annotation. The tweet's text is the only thing that is automatically acquired. The label is obtained through manual annotation (neg = negative; pos = positive).

As with any process relying on human judgement, data annotation is also prone to inaccuracies and biases. Depending on the task at hand, some inaccuracies could be introduced to the label — e.g. if the goal is to put a border around an object, everyone could draw that border over slightly different pixels. Additionally, bias can be introduced. If tasked with detecting emotion in an online comment as in Table 2.1, the decision is based on each individual annotator's experience and capability of detecting emotion. These differences make the labels produced by a single annotation unreliable.

Tweet	**A1**	**A2**	**A3**	**Label**
blagh class at 8 tomorrow	neg	neg	neg	neg
Gonna catch sum rays on this glorious day!!!	pos	pos	pos	pos
I had such a nice day. Too bad the rain comes in tomorrow at 5am	neg	pos	neg	neg

Table 2.2: Example of multiple annotations. The label requires the decisions from each individual annotator – A1, A2 and A3 (neg = negative; pos = positive) and is based on the majority vote.

To mitigate these issues, crowdsourcing with multiple annotators is required. This process involves each data point being labeled by several people and the final decision is based on the majority vote as can be seen from the example in Table 2.2. This approach makes annotation very expensive, as repeated work is needed for obtaining a single data label.

2.2 Feature Representations

As previously discussed a data point is described by the values of its features. Due to the mathematical nature of the models used, these values need to be numeric. Thus, for NLP tasks, there is a need to translate text into numeric values.

In classification tasks, a useful representation of the data is one which allows for a separation of the points into the required groups. This means that the features need to capture the useful information that correlates closely with the true label. For example, data with features x and y and the four data points A, B, C, D as seen in Figure 2.1 is not linearly separable in its respective classes (blue vs. red) when using the original features. However representing them as features with value x^2 and y^2 results in linearly separable data, making these engineered feature representations more useful.

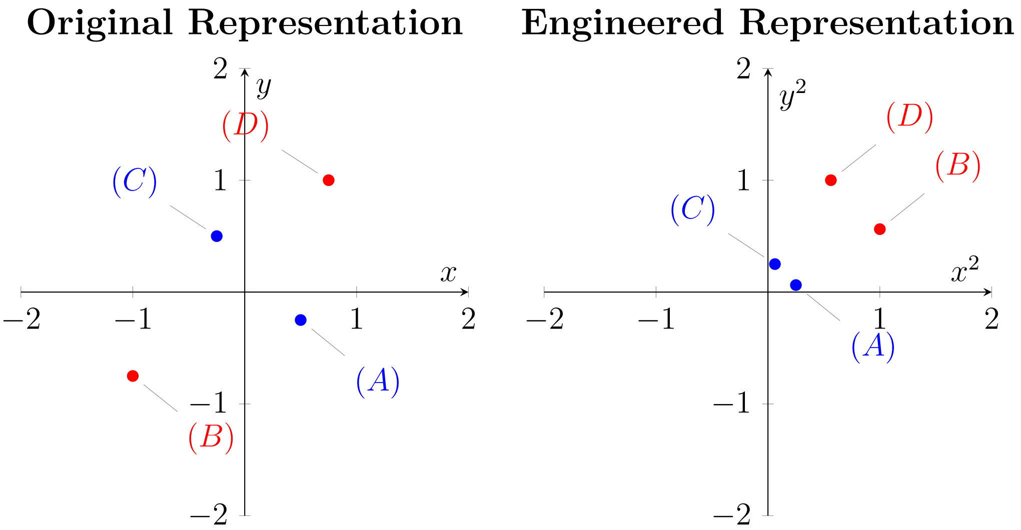

Figure 2.1: Examples of less useful (left) and more useful (right) feature representations.

There are several methods that are generally applied for transforming text into numeric values. One possible representation is bag-of-words with count matrices. In the bag-of-words approach each text is represented by the individual words (tokens) present in it, ignoring the sequence of the text. In a count matrix each row is a single document and each document is represented by a vector containing its feature values. In those vectors each dimension is a specific token and the value is the number of times that token occurs in the specified document. Other representations are also used, such as n-grams, which are similar to bag-of-words, however instead of counting the presence of single words, utilize sequences of n items (which could be words, letters, phonemes, etc.).

This type of transformation has several hindrances. One of them being the amount of time needed, since these representations are manually engineered.

2.3 Neural Networks for Text Representations

A way of circumventing the need for manual feature engineering is to learn that feature representation automatically. A good model architecture for achieving this is a neural network. The universal approximation theorem [5] states that a neural network can approximate any continuous function. This means that any feature representation could be approximated.

2.3.1 Neural Network Basics

A neural network is a graph-like machine learning model consisting of nodes connected by edges as seen in Figure 2.2. The neural network consists of layers of nodes with edges connecting them. There are three types of layers: input, hidden, and output. The first one is where the input data is passed through, while the last layer is the output. These two can be connected by either one or several layers, called hidden layers. The process of producing an output from an input is called a forward pass and consists of calculating the values of each node.

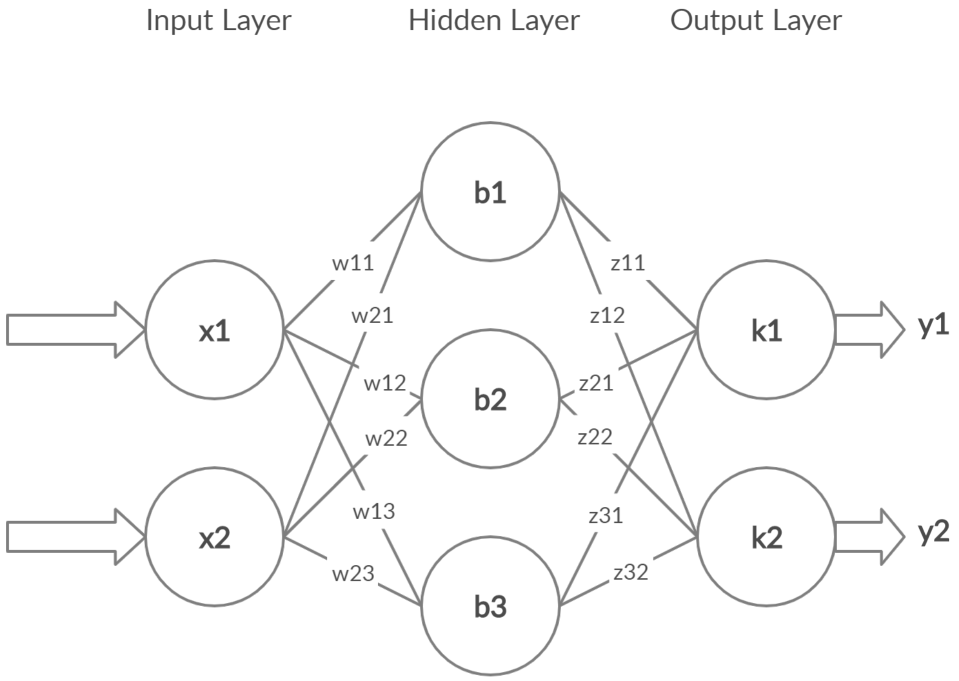

Figure 2.2: An example of a neural network.

For the example given in Figure 2.2, for a data point x_1, x_2, the values for the three hidden nodes are respectively:

$$\text{top node value} = b1 + w11 \cdot x1 + w21 \cdot x2 \tag{2.1}$$

$$\text{middle node value} = b2 + w12 \cdot x1 + w22 \cdot x2 \tag{2.2}$$

$$\text{bottom node value} = b3 + w13 \cdot x1 + w23 \cdot x2 \tag{2.3}$$

These values are then passed through a function, typically a sigmoid or tanh, obtaining the final node values $h1$, $h2$, $h3$. This is done to allow for non-linear data relationships and is called an activation function. The values for the output layer are calculated in a similar fashion. That is:

$$\text{top node value} = k1 + z11 \cdot h1 + z21 \cdot h2 + z31 \cdot h3 \tag{2.4}$$

$$\text{bottom node value} = k2 + z12 \cdot h1 + z22 \cdot h2 + z32 \cdot h3 \tag{2.5}$$

These are again passed through an activation function to produce the final prediction $y1$ and $y2$.

When training a network the values of the parameters $w\{ij\}$, $b\{i\}$, $k\{i\}$ are calculated by a process called backwards propagation. They are usually initialized to random numbers and small changes are applied with each new training data point passed through the network. When a point is passed through the model, the difference between the predicted y values and the true ones is called the loss. The gradient of this loss with respect to each parameter is calculated. Afterwards, a step is taken into the direction of negative gradient in order to minimize the function.

2.3.2 Transformers

Typical neural network architectures for text tasks are based on recurrent neural networks (RNN) [4]. These take into account the sequential nature of the data. The example in Figure 2.3 shows an RNN used for translating a sentence from one language to another.

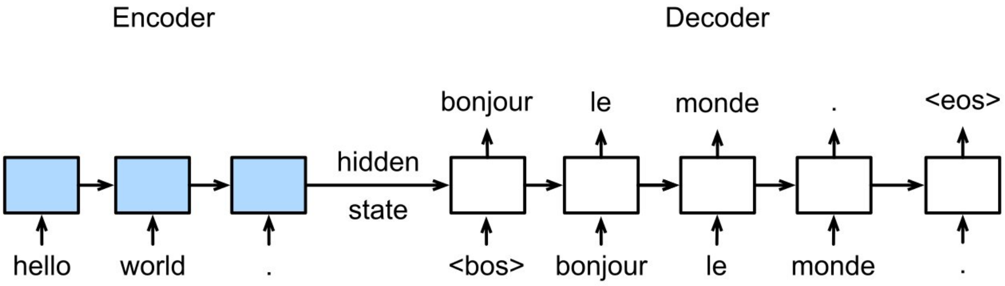

Figure 2.3: An example of a recurrent neural network [29].

This architecture is split into two parts – an encoder and a decoder – that are simply neural networks. The encoder is fed one word at a time and outputs a hidden state value. This hidden state value, together with the next word are fed through the encoder again. At the end of the input sequence, the hidden state value is fed to a decoder neural network, which produces an output word and a further hidden state value. Those are then fed to the decoder again in order to predict the next word. An issue with this architecture is that when the word "monde" needs to be output, the word that is being translated is "world", which is passed through the model several steps before. This means that the information needs be retained within the hidden state values for several passes through the model. A way to solve this is to introduce a mechanism that allows the decoder to access the relevant parts from the input sequence. This type of mechanism is called attention [2].

A further development in the field is to simply apply the attention mechanism without recurrence. This new architecture is called a transformer [25].

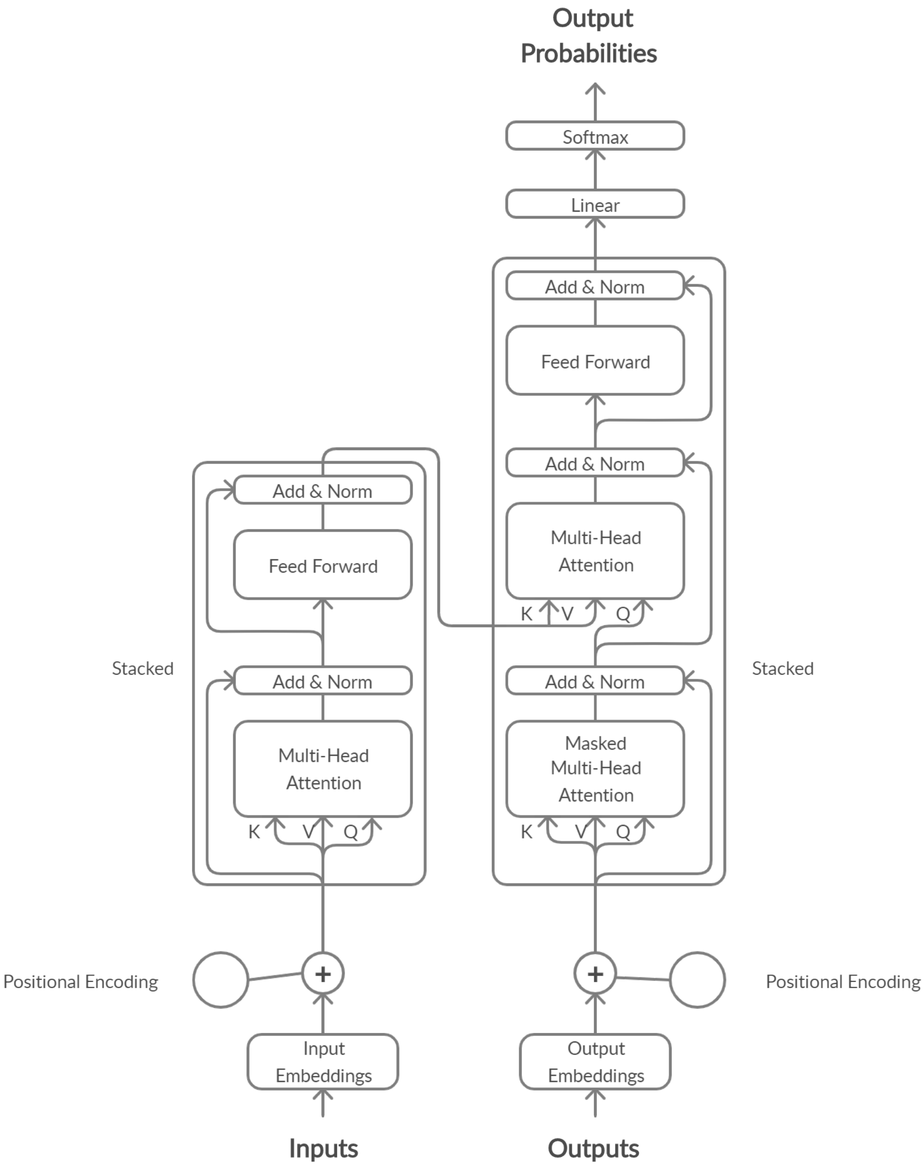

Figure 2.4: An example of the transformer architecture where both on the left and on the right side, several of the components are stacked on top of each other.

The attention mechanism in a transformer is based on a query (Q), key (K), value (V) triplet. In the case of the attention layer, which combines the input sequence and the current output, the key and value come from the input sequence and the query comes from the output. These are combined using the following formula:

$$attention(Q, V, K) = softmax\Big(\frac{QK^T}{\sqrt{d_k}}\Big)V \tag{2.6}$$

where d_k is the number of dimensions of K.

In this formula, the dot product QK^T has larger values for keys that are similar to the requested query, the softmax function essentially "picking" the relevant keys. The corresponding values to those keys are then selected. The key-value pairs could be seen as interesting facts found in the input sequence and the query is how they are accessed.

2.4 Transfer Learning

Transfer learning is a machine learning approach which is based on the idea that knowledge gained from learning how to solve problem A could be useful for solving a different problem B in the case they share some similarities. Some similarities in the data type are required – e.g. both problems focusing on images or text data. However, the tasks could be quite different – for example, problem A could be to classify images into 'cat' and 'dog' categories, whereas problem B could be to identify the exact boundaries of the cat or dog within the image. The knowledge gained most typically consists of the parameter values of the model trained to solve problem A. These can be used as initialisation of the model used for solving problem B instead of starting from an untrained model, making fine-tuning, i.e. the training for solving problem B, much faster.

For example in Figure 2.5 a model meant to identify the breed of dogs in an image can be used to obtain insights that are helpful to identifying the breed of a cat. These insights could consist of the presence of similar animal features within the images such as fur, nose and eyes. However, the same model for identifying the breed of a dog would not be helpful for classifying MRI images and whether they show a healthy brain or not. This is because the extracted features from the first model share no similarities with the second one.

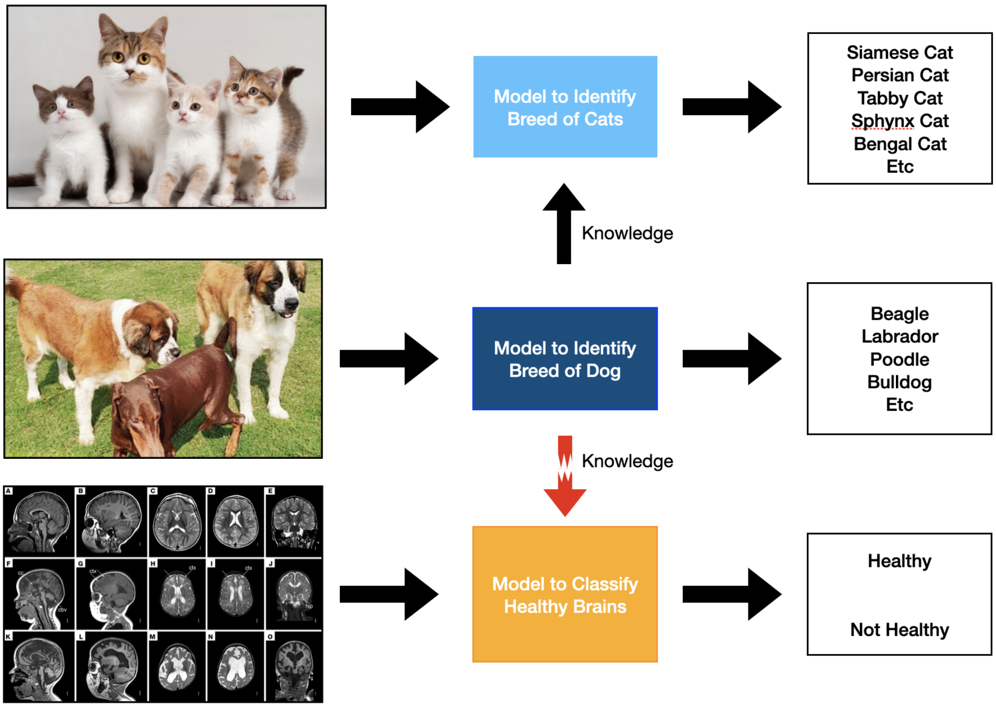

Figure 2.5: An example of transfer learning. The main model is used to identify the breed of dogs. The knowledge gained could be useful for identifying breeds of cats, however, it is probably not useful for identifying healthy brains in MRI images.

There are two main benefits of using transfer learning. The first benefit is that it makes solving the second task much less time consuming, since instead of training the entire model, one would just need to fine tune the initial model. The second benefit is that due to the vast amount of data used to train the initial model, it can significantly increase the performance even without much data at hand for the second task.

2.4.1 BERT

Several high-performing transfer learning models currently available are based on the transformer architecture, such as GPT [20] and BERT [8]. The GPT model uses transformers, which use attention in both directions. However, the task the model is trained on is next word prediction in a left-to-right direction only. Alternatively, BERT has deep bidirectionally due to both the use of the encoding part of the transformer architecture (left part of Figure 2.4) and the language model used for pre-training. The training tasks for BERT are (1) trying to reconstruct a sentence where some words have been masked, and (2) trying to predict whether in sentence pair A-B, sentence B follows sentence A. This allows it to outperform previous models [8] on the General Language Understanding Evaluation (GLUE) leader board [1].

The way BERT encodes an input sequence consists of three levels that are combined as can be seen in Figure 2.6. The first one is the positional encoding of the word, which refers to its place in the sequence. The second part refers to which document the word appears in, as BERT deals with two documents. The third level is the embedding of each word, where two special tokens are additionally used: one for the beginning of the input (CLS), another for indicating the end of a document (SEP). The model outputs a vector for each of these tokens. Typically, for classification tasks, a linear layer is added on top of the output vector for the CLS token.

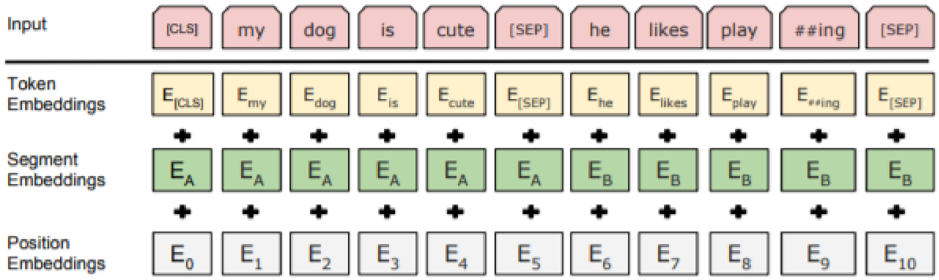

Figure 2.6: BERT input representation, where the input is a sum of three levels of embedding [8].

BERT has also shown good results identifying hate speech when applied to English data, as can be seen in [15] and [17].

2.5 Cross-Lingual Representations

Cross-lingual zero-shot learning is a type of transfer learning, where the training language is different to the testing one. This allows for the use of a resource rich language for training. The method relies on the ability of neural network models to learn mappings between different distributions, in this case between the training and the testing language. One approach is to pre-train a model on a multilingual corpus of data. In this set-up, identical subwords in a shared vocabulary can act as anchor points for learning an alignment between languages. Additionally, training on multiple languages at the same time can amplify this effect. Furthermore, due to the ability of deep networks to learn complex patterns, ones that extend beyond simple vocabulary mappings can also be found [22]. Ideally, this would lead to similar representations for texts with similar meanings, independent of the languages. That is, given two text inputs in different languages that have the same meaning (one could be a translation of the other), their representation should be similar.

2.5.1 Multilingual BERT

An extension to BERT is its multilingual version that utilizes the concept discussed above. This extends the base version by training the model on a Wikipedia data set containing 104 different languages. Results using multilingual BERT suggest that there is some alignment between languages that emerges automatically in its representations. By training on a specific language and testing on a different one, the model has shown some cross-lingual knowledge transfer is occurring for named-entity recognition and part of speech (POS) tagging [19]. Named-entity recognition locates and classifies parts of text that represent one of a number of pre-defined categories. For example, in a corpus of text one might be interested in all names of organizations, all locations, etc. Meanwhile, part of speech tagging consists of attempting to mark each word within a text with its corresponding grammatical class, e.g. finding all verbs in a corpus of text. Good results are achieved even for languages in different scripts – e.g. a model trained on Urdu produces 91% accuracy when evaluated on Hindi for POS tagging.

2.6 Data Augmentation

The data augmentation approach is used for creating new data points from existing ones. This is done by slightly changing the feature values of one data point to create a new one. Data augmentation techniques have been applied successfully to image data. One such example is to blur an image available in the data set, as can be seen in Figures 2.8 and 2.9.

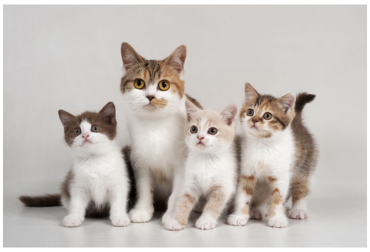

Figure 2.7: The original image.

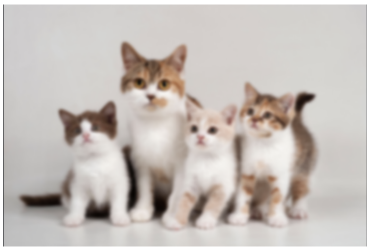

Figure 2.8: Original image blurred.

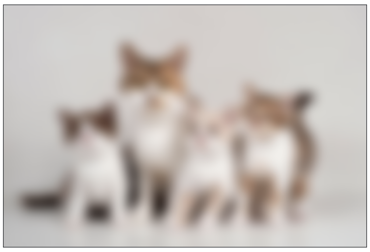

Figure 2.9: Original image blurred too much.

One of the most important aspects in the process is the trade-off between diversity and validity – i.e. the issue of choosing the correct range for the size of the changes. The validity of the label could deteriorate when the changes are too big. As can be seen in Figure 2.9, blurring the image too much makes the object in it unrecognizable. Thus, invalidating the cat label.

Making changes too small, however, diminishes the diversity of the data and therefore its usefulness as a means to create more examples.

With careful selection of the augmentation technique and thresholds for the size of the change, this is a powerful tool for increasing the data size and thus improving the robustness of the model, as by introducing more diverse data, the model will likely perform better on an unseen set.

Creating data augmentation techniques for text is an active area of research. It is not immediately obvious how well-known image augmentation techniques, like stretching and blurring, can be applied to text. Additionally, a particular issue with text augmentation is that even small changes to the data could lead to big changes in the meaning, whereas, images are not as susceptible to change, i.e. changing a pixel value slightly will not change what the image represents. The relevance of this issue could vary with the specific task at hand, e.g. for a topic classifier recognizing financial documents, changing one word might not affect the topic too much.

Three methods that are explored in this section are: TF-IDF synonym replacement, word dropout and back-translation.

2.6.1 TF-IDF Synonym Replacement

The first method that is utilized is TF-IDF (term frequency inverse document frequency) synonym replacement. This method introduces variability by replacing some words with a synonym. In order to not lose the core meaning only the words that do not carry a lot of information should be replaced. Therefore, these are selected based on a TF-IDF score, which is correlated with the importance to the label. It is calculated by multiplying both the term frequency (TF) and the inverse document frequency (IDF), as can been seen:

$$tf\text{-}idf(t, D) = tf \cdot idf = tf \cdot log\frac{N}{df} \tag{2.7}$$

In the equation tf refers to the number of occurrences of term t in all document having label D; N refers to the amount of documents in the corpus; df refers to the number of documents where the term t appears. Finally, the TF-IDF scores for each label are calculated by multiplying the IDF values with that label's TF values. That is, for each label, there is a bag of words where each word has a specific TF-IDF score. For example:

Label: 'Positive'
Document One: 'This is a great car.'

Label: 'Negative'
Document One: 'Just great. My car just broke down.'

Word	**TF pos**	**TF neg**	**IDF**	**TF-IDF pos**	**TF-IDF neg**
this	1/5	0	log(2/1) = 0.3	0.06	0
is	1/5	0	log(2/1) = 0.3	0.06	0
a	1/5	0	log(2/1) = 0.3	0.06	0
great	1/5	1/7	log(2/2) = 0	0	0
car	1/5	1/7	log(2/2) = 0	0	0
just	0	2/7	log(2/1) = 0.3	0	0.09
my	0	1/7	log(2/1) = 0.3	0	0.04
broke	0	1/7	log(2/1) = 0.3	0	0.04
down	0	1/7	log(2/1) = 0.3	0	0.04

Table 2.3: Values of TF-IDF scores for example given above.

As can be seen in Table 2.3, terms that appear in both labels ('great' and 'car') have scores of 0, whereas terms that appear more frequently in a specific label ('just' in 'negative') have a higher score. This exemplifies the correlation between the TF-IDF score and the importance of a word for a given label.

When a document is selected for augmentation, the TF-IDF scores used are based on the label of the original sentence. A uniform random number is chosen and if the word has a lower TF-IDF score (i.e. indicating the word has low importance), it is changed for a synonym in the augmented document. One advantage of this method is that the document should not lose its meaning since the replacement word has a similar definition to the original one. However, the main concern is for words with more than one definition, as the wrong one could be selected for obtaining the synonym. Examples of good and bad augmentations using this technique can be seen below.

Original Document: My brother is a cool guy
Good Augmentation: My brother is a popular guy
Bad Augmentation: My brother is a cold guy

Original Document: Cinderella had to go to the ball.
Good Augmentation: Cinderella had to go to the dance.
Bad Augmentation: Cinderella had to go to the sphere.

For all examples shown above, the method correctly changes a word for a synonym. However, specifically, in the bad augmentation instances, not using the context leads to a change in meaning for the entire document and making it an implausible data point.

2.6.2 Word Dropout

Word dropout is a data augmentation technique where a new document is created by giving every word in the original document the same probability of being removed all together. The new document would then be added to the original data set with the label belonging to the original data point.

An example of word dropout would be:

Original Document:
Sentence: I don't like Syrian refugees
Label: Aggressive
Augmented Document:
Sentence: I don't like refugees
Label Given: Aggressive
True Label: Aggressive

In this case, both carry a similar negative connotation. This means that the label would not change, therefore the new data point would be useful for training a model.

However, the main problem that could arise using this method would be to drop a word that carries a much stronger meaning within the sentence. For example:

Original Document:

Sentence: I don't like Syrian refugees

Label: Aggressive

Augmented Document:

Sentence: I like Syrian refugees

Label Given: Aggressive

True Label: Non-Aggressive

In this case, the augmented sentence does not retain the aggressive meaning. Since this method gives each augmented sentence the same label as the original version, situations like the example above would not be the most optimal.

2.6.3 Back-Translation

Back-translation consists of translating a sequence into a different language and then translating it back to the original language. The main advantage of back-translation over the previous methods is the fact that it translates the entire sentence, therefore the meaning should not change as much. For example,

Original Document:

Sentence: Yesterday, my dad told me the story of the first time he met my mom.

Intermediate Language: Spanish

Sentence: Yesterday, my dad told me the story of the first time he met my mother.

In this specific case, both sentences mean the same thing. The only change being the word “mom” to “mother”. This change could be seen as too small and providing little variability to the augmented data. In order to increase said variability a more diverse set of intermediate languages could be used. However, back-translation could produce examples that change the original meaning or make less grammatical sense. For example,

Original Document:

Sentence: Yesterday, my dad told me the story of the first time he met my mom.

Intermediate Language: Swedish

Sentence: Yesterday, my dad told me the first time I met my mother.

3
Methods

In this chapter an overview is given of the experimental set-up for the project. The data sets that are used are discussed, along with the annotation guidelines used for each one. The chapter also outlines the training process for obtaining the baselines. Additionally, the specific implementation for the cross-lingual zero-shot and data augmentation approaches are presented. Some experimentation is carried out to observe the effects of training set size and the impact of translating of English training data. The evaluation approach is discussed. Finally, some analyses of the data sets are carried out in order to investigate the cohesive groups present within them.

3.1 Hate Speech Data Sets

Several data sets are used to explore the different aspects of the task at hand. All data sets are tweets that are manually annotated for hate speech. These data sets are balanced between the two classes. The main data set is the HatEval [3] data set, containing tweets in both English (EN) and Spanish (ES). Spanish is used as a proxy for a low resource language and the effect of using English during the training process is explored. These data sets are taken as a basis for the project. Since they belong to the same competition, the data distributions between English and Spanish should be most similar in the gathering process, annotation guidelines and time period. However, additional languages are also used. Those are Arabic (AR) [18], Portuguese (PT) [10] and Indonesian (ID) [13]. For the purpose of a comparison two more English data sets are utilized – Waseem & Hovy (WH) [26] and Founta (F) [11], where tweets are scraped through the Twitter API leading to 4440 balanced training examples in set WH and 5151 balanced training examples in set F.

Minimal modification is done to the data – mentions and URLs are removed, the hashtag sign is dropped leaving each hashtag to be treated as a word. If a letter is repeated 3 or more times, the repetitions are removed.

3.1.1 Annotation Guidelines

As previously mentioned, hate speech has slight differences in its definition. These are reflected in the differences in annotation guidelines for each of the used data sets.

- **HatEval**[1]: Hate speech against immigrants is defined as a "message that spreads, incites, promotes or justifies HATRED OR VIOLENCE TOWARDS THE TARGET, or a message that aims at dehumanizing, hurting or intimidating the target" and must have "IMMIGRANTS/REFUGEES as main TARGET, or even a single individual, but considered for his/her membership in that category (and NOT for the individual characteristics)". Additionally, hate speech against women is defined as "a text that expresses hating towards women in particular (in the form of insulting, sexual harassment, threats of violence, stereotype, objectification and negation of male responsibility)". An important note is made that this data set does not label hate speech against any target other than immigrants/refugees or women as hate speech. In other words. it is only concerned with those two target groups.
- **Founta**: "Language used to express hatred towards a targeted individual or group, or is intended to be derogatory, to humiliate, or to insult the members of the group, on the basis of attributes such as race, religion, ethnic origin, sexual orientation, disability, or gender". [11] As can be seen a wider range of targets is considered. In this data set, the annotators are asked to label each tweet with one of the following categories: normal, spam, abusive, and hateful, making an explicit distinction between abusive language and hate speech.
- **Waseem & Hovy** [26]: This data set gives an eleven-point guideline for what is considered hate speech, which is mainly centered around using sexist/racial slurs, attacking/seeking to silence/criticizing a minority or defending such behaviour. Two further points take into account the specific Twitter nature of the data. Tweets supporting problematic hashtags are labeled as hate speech. Additionally, ambiguous tweets from users with offensive screen names are also considered hate speech. This data set focuses on sexism and racism as the only categories of interest.
- **Indonesian** [13] : In order to arrive at a definition of hate speech a focus group discussion is conducted. It is concluded that hate speech has a particular target, category and level. For targets, as in the previously mentioned definitions, both individuals and groups are considered. However, the categories have a wider range with religion, race/ethnicity, physical disability, gender, sexual orientation all being considered. The label is further split into levels, however all levels are considered as hate speech examples.
- **Arabic** : "Hate speech tweets are those instances that: (a) contain an abusive language, (b) dedicate the offensive, insulting, aggressive speech towards a specific person or a group of people and (c) demean or dehumanize that person or that group of people based on their descriptive identity (race, gender, religion, disability, skin color, belief)." [18] Similarly to Founta, the label is split into normal, abusive and hate.
- **Portuguese** : “Hate speech is language that attacks or diminishes, that incites violence or hate against groups, based on specific characteristics such as physical appearance, religion, descent, national or ethnic origin, sexual ori-

[1] https://github.com/msang/hateval/blob/master/annotation_guidelines.md

entation, gender identity or other, and it can occur with different linguistic styles, even in subtle forms or when humour is used." [10]

3.2 Baseline

Two models are used as a baseline for determining multilingual BERT's ability to detect hate speech – a model trained on all available Spanish tweets (Spanish base) and a model trained on the same size of English tweets (English base). These tweets are taken from the HatEval data set that contains both English and Spanish data annotated with similar guidelines. The data is balanced where 50% of the tweets are labeled hate speech.

The data is split into three data sets – training, validation and testing. The training data is passed through the model for finding the weights. The validation set is used to calculate the performance of the trained model for each hyper-parameter set, in order to determine the best one. The testing set is used to calculate the performance of the final model. Separate validation and testing data sets are needed to avoid over-estimating the performance. Since the hyper-parameters are selected based on the performance on the validation set those could be over-fit to the validation data. Therefore, to predict real-world performance a completely held out (testing) set is needed – i.e. one that has not been used in any stage of the hyper-parameter tuning process.

Spanish base utilizes all available data in HatEval, which consists of 3600 tweets. This is the number used for training both the Spanish base and the English base models. As English base is evaluated for cross-lingual zero-shot learning, the performance of the model on its own test set is used as a baseline for the comparison.

The base models are also used for performing a hyper-parameter search, where the hyper-parameters that are found to have the best performance are used for all following models. Several hyper-parameters are important when using BERT.

The first hyper-parameter to be explored is the learning rate which is central to any neural network. This determines the size of the change in the model's parameters at each step. A small learning rate makes the changes too small, making the converging of the model slower. A large learning rate could lead to the impossibility of converging as the big changes in the parameters can 'overshoot' the optimum. For BERT there is a recommended range of learning rates that are explored: 5×10^{-5}, 3×10^{-5}, 2×10^{-5}.

In order to reduce the chance of over-fitting (i.e. learning the training data instead of the general patterns) dropout in both the hidden layer and the attention layer is used [24]. This is controlled by the dropout rate parameter. It determines a percentage of random connections between nodes that are ignored during training. The default dropout rate for the BERT model is 10 percent. Several values are explored to find any improvement in the model's performance. Initially, one parameter at a time is tested to provide a relative range that is used to perform a grid search of both parameters at the same time.

The final hyper-parameter that is explored is the number of epochs – i.e the number of passes through the training data that are needed to find the optimal weights. Since neural networks change their parameters based on a subset of the data at each step of the training process, this means that further data can cause the network to "unlearn" previous patterns – i.e. change the parameters in the opposite direction. Validation accuracy after each epoch is used to determine the correct number. Once the validation accuracy stops increasing with additional epochs the model is stable and further passes through the data do not lead to improvements.

Both baselines are trained and tested 10 times in order to obtain a range for the performance and a sense for the stability of the models. This is done for all models used in this work.

3.3 Cross-Lingual Zero-Shot Learning

The English base model's performance is evaluated on a Spanish test set to assess the possibility for cross-lingual zero-shot learning.

Additionally, several other Twitter data sets annotated for hate speech are used for testing both the English base and the Spanish base model to assess the performance on other languages.

Two additional English Twitter data sources are used for training models. Those models are evaluated on a test set from each of the three available English sources. This offers a baseline to put all the previous results in context, as these models should show minimal drop in performance, due to no cross-lingual learning being needed. There are several possible sources of differences in these data sets: the time period they were collected in – which could lead to difference in the topics discussed, annotation – which could introduce differences in bias, and the collection process – i.e. how is that subset of tweets selected (while some data sets are randomly extracted, others are targeted by searching for specific terms). For a model to be useful for the general detection of hate speech, these differences should be negligible.

3.4 Effects of Training Set Size on Performance

To explore the effect of small training sets on performance, the accuracy of models trained on a range of sizes of the proxy language (Spanish) is obtained. This provides a sense of how performance depends on size and when the scarcity of data becomes a significant issue. A series of sizes are explored – 250, 500, 1000, 1500, 2000, 2500, and 3000 tweets. All sets are balanced and the same hyper-parameters are used as the optimal ones found for the base model that is trained on 3600 Spanish tweets. The expected trend is reduced performance with reduced size.

3.5 Data Augmentation

In this project, data augmentation is applied to increase the size of a data set by adding additional tweets to the already existing ones. As already mentioned, those tweets are generated by three different methods. Three different data sizes (250, 1000, and 2000) are used as examples of very small, medium and relatively large data sets. This shows whether the data augmentation methods lead to improvements for a specific size of data. Each of the three reduced data sets are increased in size in steps to follow the series 250, 1000, 2000 and 3600.

3.5.1 TF-IDF Synonym Replacement

The first data augmentation technique that is implemented is TF-IDF synonym replacement. The implementation of the method is based on previous work [28]. As previously mentioned this augmentation relies on substituting a word for its synonym. In order to obtain a synonym the Open Multilingual Wordnet repository [12] is used. A list of synonyms is extracted by getting all possible translations in English and for each of those translations getting all possible translations back to Spanish. A random word from that list is then selected.

In order to decide which words should be replaced a random number between 0 and 1 is assigned to each word contained within it. If this number is less than the threshold calculated by equation 3.1 the word is replaced by a synonym. This threshold is normalized for each tweet using the mean and maximum TF-IDF score of the entire tweet.

$$threshold = min(1, constant \cdot \frac{maxScore - wordScore}{meanScore}) \tag{3.1}$$

The constant is the parameter that has to be tuned in order to find which value works the best.

3.5.2 Word Dropout

The second data augmentation method that is used is word dropout. This approach is based on previous works as seen in [28] and [27]. It is implemented by selecting a random tweet to be augmented and assigning a random number between 0 and 1 to each of its words. If this number is less than a certain threshold, said word is dropped entirely from the augmented tweet. The main parameter that has to be tuned in this specific method is the threshold.

3.5.3 Back-Translation

The final data augmentation method is back-translation. This is based on previous work described in [28]. It translates each tweet into a random language and then translates it back. The process is done through the Google API and the intermediate language is chosen at random from a list of 106 different ones. As all languages available through the API have the same probability of being chosen as a middle language, the dependency on which one is used is reduced. This also allows for multiple translations, i.e. augmentations, from a single tweet. The linguistic proximity between the original and middle languages could influence the quality of the data augmentation and can be seen as a hyper-parameter for the technique. However, this effect is not been explored in the current project.

3.6 Translation of English Training Data

A further method for obtaining a large training data set is to translate a data set in a resource rich language to the required language. In this particular case the HatEval English training data is translated to Spanish, Arabic, Portuguese and Indonesian. The Google API is used to obtain a single translation. This introduces some further variability which is governed by how well the Google API performs in translating between a particular pair of languages.

3.7 Evaluation

The evaluation follows the standard protocol for this type of task. As can be seen in [3] the metrics that are used are accuracy (ACC), precision (PRC), recall (RCL) and F1 score (F1).

In a classification task with two classes, where one is positive and one is negative, there are four types of data points: true positives, true negatives, false positives and false negatives, as can be seen in Figure 3.1. For this specific project the positive class can be seen as the examples containing hate speech.

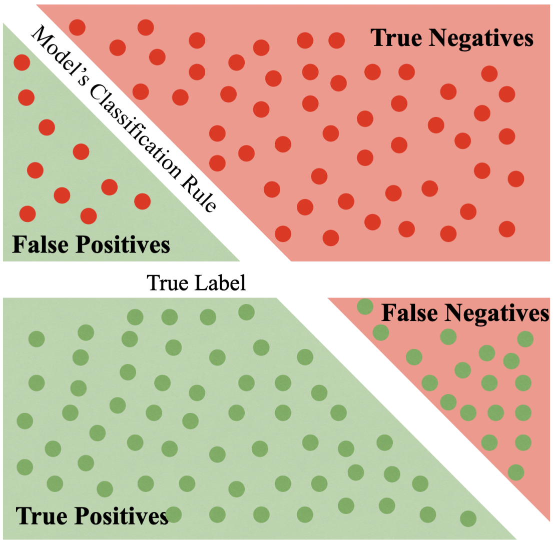

Figure 3.1: Example of true positive and negative elements versus selected positives and negatives. The green and red points refer to positive and negative true labels, respectively. While the green and red backgrounds refer to what the model considers positive and negative, respectively.

Based on said counts, the metrics are calculated as follows:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.2}$$

$$PRC = \frac{TP}{TP + FP} \tag{3.3}$$

$$RCL = \frac{TP}{TP + FN} \tag{3.4}$$

$$F1 = 2 \cdot \frac{PRC \cdot RCL}{PRC + RCL} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \tag{3.5}$$

As the data sets are balanced, accuracy is a good primary measurement. To assess any instability due to the random initialisation of the models each one is trained and tested 10 times, and the average of each metric with its standard deviation is calculated.

For any hyper-parameter or data augmentation threshold search a validation set is used for evaluation, whereas the final results are calculated on a further held-out test set. The validation and test sets for all experiments are of size 400 tweets with 200 hate speech and 200 non-hate speech examples.

3.8 Data Sets Analyses

Some further investigation into the data is performed in order to give a broader picture of all aspects of the task. Clustering and classification analyses are used for detecting any possible groups within the data that could hinder the model. This is done in order to examine whether a different label produces similar results or hate speech is intrinsically hard to model.

3.8.1 Clustering

As previously mentioned in section 2.1.1, clustering is used for discovering groups of similar data. In this project the clustering model applied is k-means. As k-means finds local optima it is dependent on initialization. To get a stable performance ten random initializations are used.

Two data representation are used – TF-IDF and BERT vectorization. As discussed in section 2.2, count matrices are used. For TF-IDF a token is considered to be each word present and the count matrix is normalized by the TF-IDF scores. Whereas for the BERT approach, the BERT tokenizer (based on WordPiece) is used to generate the tokens. This tokenizer can split unknown or longer words into multiple sub-words. A double hashtag in the token represents that this is not the beginning of a word, as can be seen:

Original Document:
Here is the sentence I want embeddings for.
Tokens:
'here', 'is', 'the', 'sentence', 'i', 'want', 'em', '##bed', '##ding', '##s', 'for', '.'

For evaluation of how well the clusters found align with the classes of interest (as defined by the labels) two metrics are used – the purity and inverse purity. These are calculated using:

$$purity = \frac{1}{N} \sum_{k} \max_{j} |w_k \cap c_j| \tag{3.6}$$

In the formula above w_k refers to the data points having label k and c_j refers to the data points in cluster j. In order to calculate the inverse purity the w_k refers to the cluster and c_j refers to the data point label. An example of these calculations is given in Figure 3.2.

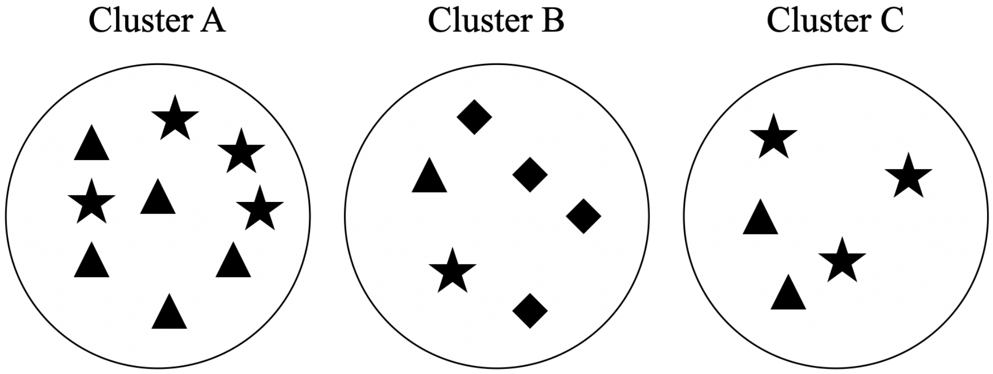

Figure 3.2: Purity and inverse purity calculation for the three clusters above. Majority class for each cluster: cluster A: ▲, 5; cluster B: ♦, 4; cluster C: ★, 3. Purity = $\frac{1}{20}(5+4+3) \approx 0.6$. Majority label for each class: ▲: cluster A, 5 ♦: cluster B, 4 ★: cluster A, 4. Inverse purity = $\frac{1}{20}(5+4+4) \approx 0.65$.

Furthermore, in order to investigate the contents of each cluster the top 10 most descriptive tokens are obtained. Each cluster is represented by a centroid, which is a vector with a dimensionality equal to the number of available tokens. Each dimension/token has a corresponding weight. The tokens corresponding to the highest weights are selected as the most descriptive ones.

In the current project the groups of interest are hate speech vs non-hate speech. Additionally, clustering is applied in order to determine whether the three different English training data sets are distinct enough to form individual groups.

3.8.1.1 Clustering for Hate Speech

A clustering algorithm is applied on both the English HatEval and Spanish HatEval data sets. This is done in order investigate whether there is a strong signal in the data that can be used to split into cohesive groups. Ideally the resulting clusters would show a correlation between the clusters found and the hate speech vs. non-hate speech label. This would mean that the vocabulary found in each class is distinct enough.

3.8.1.2 Clustering English Training Data Sets

A clustering algorithm is applied on an amalgamation of all three English training data sets, i.e. all three data sets are joined into a single one. The label used for this investigation is the training data set source. A correlation between the cluster and the label would mean that a data source uses vocabulary distinct from the other sources. This could happen if one of the data sources focuses on completely different events, e.g. discussing the American elections versus the Syrian refugee crisis in Europe.

A more detailed view of the correlation between clusters and labels of interest can be obtained by producing data distributions by clusters. This is done by plotting the percentage of data falling within each cluster for each label value. For example, in the case when data sources are explored, the percentage of data coming from Source 1 and falling in cluster 1, 2 and 3 respectively, is calculated. Similarly, this is calculated for Source 2 and 3. Both data source and hate speech labels are explored, to evaluate their correlation with 3 and 2 clusters respectively. If clusters perfectly correspond to labels, the plots are expected to approximate the ones presented in Figure 3.3, i.e. each of the groups should have 100 percent of its data in its own distinct cluster.

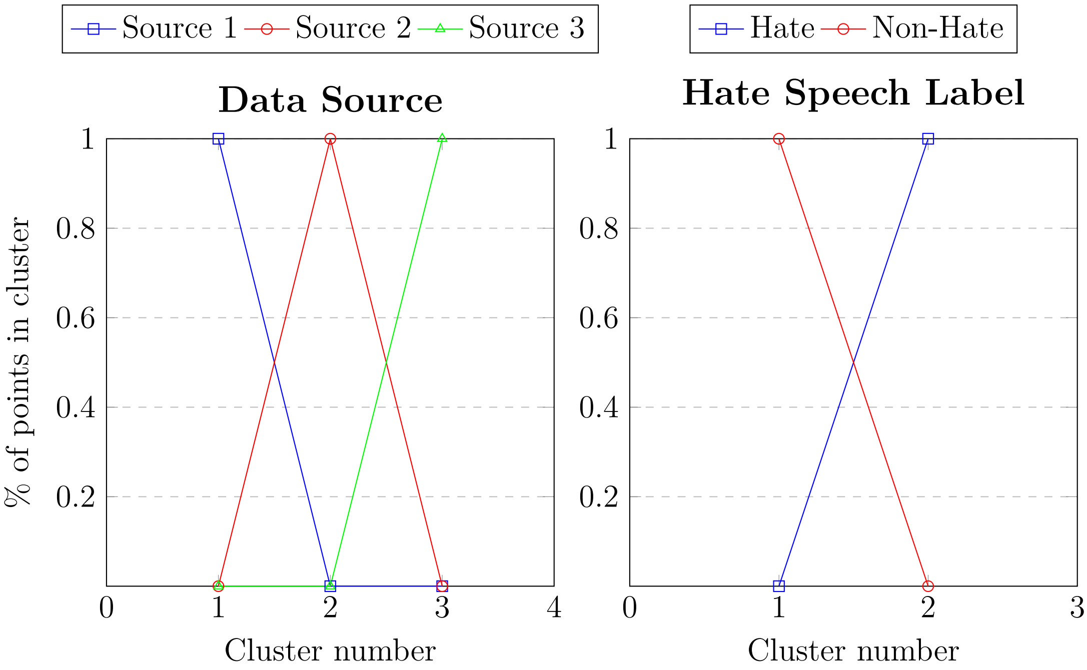

Figure 3.3: Distribution of data compared to labels when perfect correlation is observed.

3.8.2 Classifying English Training Data Sets

A further investigation into a possible dissimilarity between the different English data sources is done. A pre-trained Multilingual BERT classifier is fine tuned on the same data set as Section 3.8.1.2, where all the English training data sets have been joined together. If the classifier shows good performance it could point to the model being able to learn particularities of the data, rather than the general patterns of hate speech.

For evaluation the accuracy and a confusion matrix are used. The latter shows the number of data points that are labeled by the model as a specific class vs the true label of those data points. An example of this for a case with two classes where one is positive and the other is negative can be seen in Table 3.1

		True Label	
		Positive	Negative
Predicted Label	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

Table 3.1: A confusion matrix with two classes

The training and testing is repeated ten times in order to obtain mean and standard deviation for each of the evaluation matrices in question.

4
Results

The current section presents the results for both explored methods. Cross-lingual zero-shot learning shows better results than a random classifier, while still exhibiting a substantial drop from the baseline. Meanwhile, two of the data augmentation techniques show improvements when used on the smaller data set sizes.

For these models, the main metric used for evaluation – the accuracy – is relatively stable with standard deviations in the range 1% - 4%. However, the other three metrics show more variability with standard deviations going all the way up to 11.57%.

The additional data set exploration experiments show no substantial topic differences in either the hate vs non-hate speech groups or between the three different English data sets. However, the classification experiment suggest the BERT model might be able to learn data set particularities rather than general hate speech patterns.

4.1 Baseline

As previously mentioned, two baselines are trained using 3600 English and 3600 Spanish tweets respectively. A hyper-parameter search is performed as a first step in order to determine optimal values.

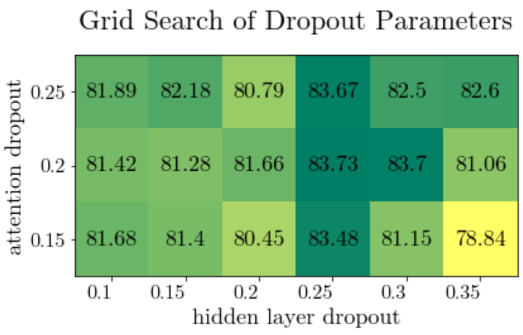

Figure 4.1: Spanish base grid search. Darkest green indicates highest accuracy value – 83.73.

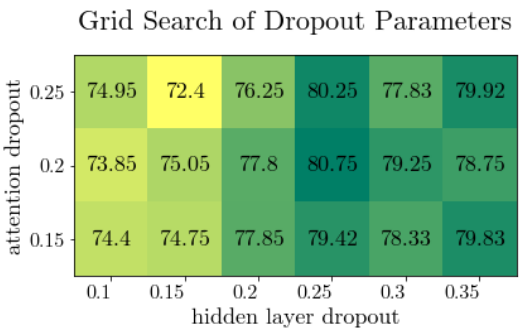

Figure 4.2: English base grid search. Darkest green indicates highest accuracy value – 80.75.

From the explored hyper-parameters the learning rate that shows best performance for both the English and the Spanish base models is 2×10^{-5}. Performance on the validation set seems to reach its peak after 10-15 epochs while not dropping for more epochs. As multiple models are trained and tested, in order to accommodate any models that need more epochs to reach their peak some margin is added. Models are evaluated based on performance after they are trained for 30 epochs. For dropout rate a grid search is performed, testing each set of values. The results are shown in Figures 4.1 and 4.2 where it can be seen that the best set for both models is hidden layer dropout rate of 0.25 and attention layer dropout rate of 0.20.

The results on the testing sets are shown in Table 4.1. The English base results are comparable to previous work on the same data set as seen in [15], where the accuracy obtained is 74.8% using the base English BERT model.

Train	Test	ACC	RCL	PRC	F1
EN	EN	73.9	66.6	78.1	71.65
ES	ES	80.67	80.55	80.83	80.62

Table 4.1: Performance of the English Base (EN) and Spanish Base (ES).

4.2 Cross-Lingual Zero-Shot Learning

As mentioned in Section 3.3, both baselines are tested for performance on all other available languages. A completely random classifier with two classes and balanced data would result in an accuracy of 50%. As can be seen from Table 4.2, most of the cross-lingual zero-shot experiments have an accuracy above a random classifier. However, there is a significant drop from the baselines (i.e. trained and tested on English and trained and tested on Spanish).

Train	Test	ACC	RCL	PRC	F1
EN	EN	73.9	66.6	78.1	71.65
EN	ES	56.03 ↓	26.05	66.15	36.52
EN	AR	57.65 ↓	23.0	76.06	34.64
EN	ID	51.58 ↓	7.55	62.24	13.22
EN	PT	55.55 ↓	17.35	74.04	27.73
ES	ES	80.67	80.55	80.83	80.62
ES	EN	54.18 ↓	19.55	64.16	29.44
ES	AR	50.72 ↓	8.25	55.82	14.12
ES	ID	51.52 ↓	9.3	61.42	15.84
ES	PT	60.35 ↓	36.45	70.04	47.69

Table 4.2: Performance of English and Spanish base on other languages. Baselines with no cross-lingual learning are shown in gray.

The base English model, when evaluated on the Spanish test data set shows a significant drop in performance of 16%. This trend is also observed for the other test languages – summarized in Table 4.2. The worst performance is shown by the base Spanish model evaluated on Arabic and best performance is shown by the same model evaluated on Portuguese. None of the cross-lingual zero-shot evaluations show an accuracy above 60% independent of the original model's performance on its own test set – i.e. even though the base Spanish model has almost 10% higher accuracy score on its own data set, the drop in performance is comparable for both the English base and Spanish base models. Some correlation between performance and language group is observed (e.g. Spanish on Portuguese performs better than Spanish on Arabic), however, these differences seem to be dominated by the general drop for all languages. It can also be observed that a drop in recall is much more evident than a drop in precision, likely driving the drop in accuracy. This points to much more lenient decision-making on different languages, marking less examples as hate speech.

The results from all three available English data sets tested on each other can be seen in Table 4.3. The two additional data sets show a higher accuracy on their own testing data, however a similar drop into the 60s is observed when testing on other English test sets.

Train	Test	ACC	RCL	PRC	F1
HatEval	HatEval	73.9	66.6	78.1	71.65
HatEval	WH	57.75 ↓	28.18	68.81	39.58
HatEval	F	57.87 ↓	22.83	76.46	34.87
WH	WH	82.55	79.97	84.38	82.05
WH	HatEval	60.08 ↓	57.85	61.09	58.92
WH	F	62.11 ↓	37.98	74.00	49.35
F	F	81.64	78.30	84.07	80.99
F	HatEval	59.25 ↓	75.05	57.13	64.75
F	WH	66.13 ↓	68.81	65.68	66.84

Table 4.3: Performance of different English models. Baselines on own test sets are shown in gray. WH refers to [26], F refers to [11] and HatEval refers to [3].

Since both Waseem & Hovy and Founta perform better on their respective held-out test sets, both models are tested on all other available languages to test their cross-lingual zero-shot performance. The results can be seen in Tables 4.4 and 4.5 for Waseem & Hovy and Founta, respectively. Waseem & Hovy shows a similar accuracy to HatEval, however the F1 score is higher mainly due to more balanced precision and recall scores. Founta shows accuracy improvement for all languages other than Spanish, as well as a substantial improvement in F1 scores.

Train	Test	ACC	RCL	PRC	F1
WH	WH	82.55	79.97	84.38	82.05
WH	ES	53.52	43.15	54.58	47.6
WH	AR	55.62	34.6	60.73	41.66
WH	ID	53.25	17.3	62.09	26.48
WH	PT	56.75	34.8	62.3	44.23

Table 4.4: Performance of Waseen & Hovy data set on other languages. Baseline with no cross-lingual learning are shown in gray.

Train	Test	ACC	RCL	PRC	F1
F	F	81.64	78.30	84.07	80.99
F	ES	56.5 ↓	51.7	57.09	54.03
F	AR	60.25 ↓	51.85	62.41	56.02
F	ID	59.38 ↓	58.35	59.7	58.87
F	PT	63.5 ↓	69.1	62.22	65.36

Table 4.5: Performance of Founta data set on other languages. Baseline with no cross-lingual learning are shown in gray.

4.3 Effects of Training Set Size on Performance

As stated in Section 3.4, the effects of training set size on performance are evaluated. This is done in order to evaluate when resources become too scarce and start substantially affecting the accuracy. This relationship can be seen in Figure 4.3. When data set size is in the thousands accuracy does not seem to depend as much on size. In that region accuracy increases 5.65% when size is increased from 1000 to 3600 (3.6 times). However, when sizes are in the hundreds a much stronger dependency is observed with accuracy dropping 11.55% when size is decreased from 1000 to 250 (4 times).

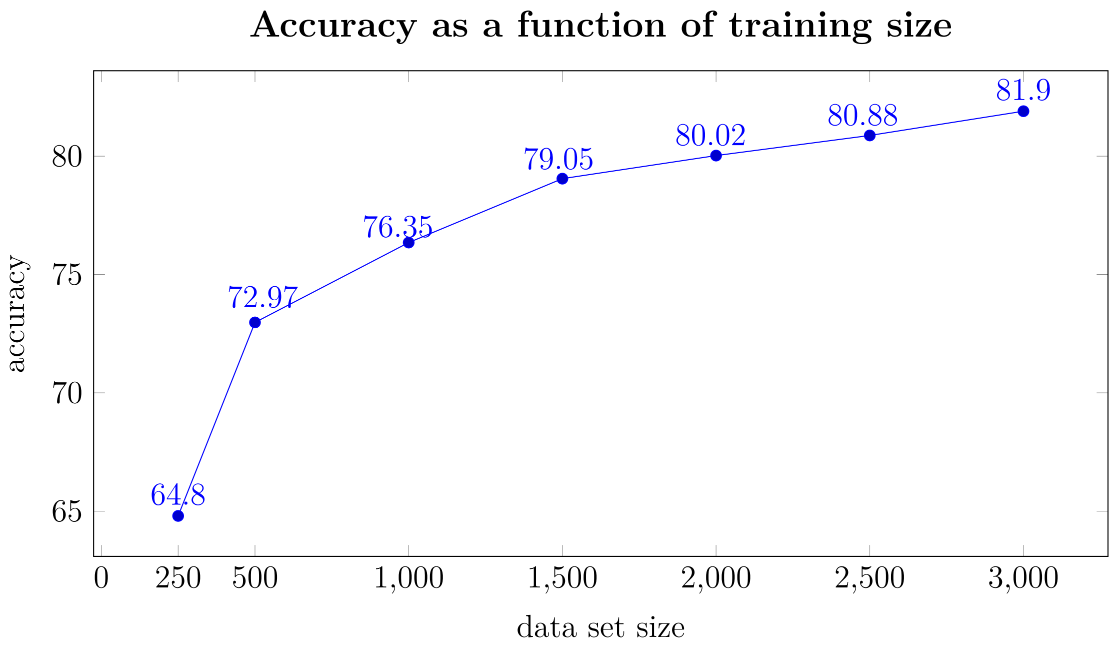

Figure 4.3: Size dependence of performance for Spanish data.

4.4 Data Augmentation

Data augmentation shows most improvement for the small data set size. For larger data sizes there is marginal or no improvement at all. This is consistent with previous work on data augmentation [27], where best results are obtained for the smallest data sizes.

4.4.1 TF-IDF Synonym Replacement

The TF-IDF synonym replacement method substitutes a word for its synonym based on its relevance to the meaning of the sentence. This meaning is quantified by using the TF-IDF score, since its correlated to it. The synonym is chosen at random from a list created by obtaining synonyms for all possible meanings.

This approach shows no statistically significant improvement during the parameter search phase. Preliminary results on the validation set can be seen in Table 4.6. Some increase in accuracy can be observed, however, this is outweighed by the variance of the results. Based on the poor performance on the validation set no further experimentation and evaluation on the test set is executed.

Initial Size	Final Size	Threshold	ACC	StDev
250	250	–	66.51	1.75
250	1000	0.3	62.42 ↓	3.43
250	1000	0.5	62.67 ↓	2.41
250	1000	0.9	61.66 ↓	3.48
1000	1000	–	76.37	1.63
1000	2000	0.3	76.65 ↑	1.71
1000	2000	0.5	75.74 ↓	1.61
1000	2000	0.9	74.80 ↓	5.45
2000	2000	–	82.45	1.91
2000	3600	0.3	82.94 ↑	1.42
2000	3600	0.5	82.21 ↓	1.56
2000	3600	0.9	81.83 ↓	1.70

Table 4.6: Validation set TF-IDF results. Baselines where no DA is applied are shown in gray.

4.4.2 Word Dropout

In the next data augmentation technique – word dropout – in order to create a new data point a tweet from the training data is duplicated and some words are removed at random from the duplication.

After running the parameter search for word dropout, it is found that the threshold that works best for sizes 250 and 1000 is 0.3 and for size 2000 it is 0.1. Using said thresholds the size dependency is tested. As can be seen in Table 4.7, the accuracy does increase when augmenting the smaller data set. However, when augmenting the 1000 data set to 2000 the accuracy is not significantly improved, while it decreases when the same data set is augmented to 3600. The final data set that contains 2000 tweets and is augmented to 3600, shows a slight drop in accuracy compared to the original 2000 data set.

Initial Size	Final Size	ACC	RCL	PRC	F1
250	250	64.8	84.6	61.24	70.48
250	1000	68.97 ↑	83.65	65.1	72.98
250	2000	69.47 ↑	70.45	69.4	69.51
250	3600	70.15 ↑	77.35	68.09	71.75
1000	1000	76.35	85.55	72.72	78.36
1000	2000	76.97 ↑	82.15	74.82	78.13
1000	3600	73.43 ↓	75.45	72.87	73.81
2000	2000	80.02	80.8	79.81	80.11
2000	3600	79.4 ↓	81.7	78.42	79.7

Table 4.7: Results obtained using word dropout. Baselines where no DA is applied are shown in gray.

4.4.3 Back-Translation

When using the final data augmentation, back-translation, a tweet from the original training data set is translated to an intermediate language and then back again in order to obtain a new data point.

Since back-translation has no parameters that are tuned in the current work, the final test results are summarized in Table 4.8. It can be seen that there is a small improvement when using this augmentation technique on the smallest sized data set. However, for both the medium and largest data set it shows a drop in performance.

Initial Size	Final Size	ACC	RCL	PRC	F1
250	250	64.8	84.6	61.24	70.48
250	1000	66.5 ↑	56.6	70.69	62.2
250	2000	68.72 ↑	70.85	68.34	69.3
250	3600	68.58 ↑	63.3	70.87	66.67
1000	1000	76.35	85.55	72.72	78.36
1000	2000	75.37 ↓	81.3	73.04	76.69
1000	3600	73.95 ↓	77.85	72.46	74.87
2000	2000	80.02	80.8	79.81	80.11
2000	3600	76.12 ↓	73.35	77.87	75.3

Table 4.8: Results obtained using back-translation. Baselines where no DA is applied are shown in gray.

4.5 Translation of English Training Data

In order to create a further baseline for comparing the cross-lingual zero-shot experiments, a translation of the HatEval training data set into each respective language is used for training a model. The comparison between cross-lingual zero-shot models and translation of the English training data set is presented in Table 4.9. The only statistically significant change is observed for the Spanish data, where translation shows an improved performance. The other three languages tested show small increase or decrease with translation, but, as mentioned, those changes are not statistically significant. However, the F1 score improves in all cases where the biggest improvement is 24% for Spanish.

Train	Test	ACC	RCL	PRC	F1
EN	ES	56.03	26.05	66.15	36.52
ENtoES	ES	61.72↑	58.45	62.73	60.14
EN	AR	57.65	23.0	76.06	34.64
ENtoAR	AR	57.45 ↓	30.85	66.32	41.31
EN	PT	55.55	17.35	74.04	27.73
ENtoPT	PT	56.22 ↑	19.3	73.96	30.5
EN	ID	51.58	7.55	62.24	13.22
ENtoID	ID	52.4↑	17.15	59.89	25.23

Table 4.9: Translated vs cross-lingual zero-shot results.

4.6 Data Set Analyses

The results from the clustering do not suggest a strong correlation between clusters and neither, the data source label nor the hate speech label. However, the classification approach produces quite good performance for the task of distinguishing data sets.

4.6.1 Clustering

The clustering experiments are mainly evaluated by purity scores and inverse purity score. As mentioned in Section 3.8.1, the purity score is higher when each cluster predominately contains data points from one unique label, whereas the inverse purity is higher when all data points from a label are contained within the same cluster, rather than being scattered amongst several. Additionally, as described in the same section, two different data representations are used – TF-IDF and BERT. The former uses whole words and normalizes them by the TF-IDF score, whereas BERT uses the WordPiece tokenizer that splits words into sub words.

All clustering experiments consistently show low purity scores. However, high inverse purity scores are observed in most cases. In general, the clusters do not seem to be strongly correlated with the explored labels.

4.6.1.1 Clustering of HatEval Data Sets

Two different experiments are presented, using two and four clusters. The former is used to explore cluster correlation to the hate and non-hate speech groups. Meanwhile, the latter is used in order to also account for the two types of hate speech within the data set, towards women and towards immigrants. The clustering results can be seen in Table 4.10 for English data and Table 4.11 for Spanish data.

The baselines are calculated for a balanced test data set, i.e. what purity and inverse purity is expected when the cluster assignment is random. The purity is generally low. In the English data set, best results are seen with TF-IDF and four clusters. In the Spanish data set, TF-IDF also shows highest purity score regardless of the amount of clusters. Additionally, inverse purity shows a more substantial increase from the baseline.

Data Representation	**Clusters**	**Purity**	**Inverse Purity**
baseline	2	0.50	0.50
TF-IDF	2	0.52	0.83
BERT	2	0.55	0.90
baseline	4	0.50	0.25
TF-IDF	4	0.68	0.61
BERT	4	0.55	0.68

Table 4.10: Clustering results for hate speech in English training HatEval data.

Data Representation	**Clusters**	**Purity**	**Inverse Purity**
baseline	2	0.50	0.50
TF-IDF	2	0.58	0.87
BERT	2	0.50	0.81
baseline	4	0.50	0.25
TF-IDF	4	0.58	0.34
BERT	4	0.51	0.71

Table 4.11: Clustering results for hate speech in Spanish training HatEval data.

As described in section 3.8.1, the most descriptive tokens for each cluster are obtained by selecting the most influential features of its centroid. When using BERT as a tokenizer these are not very informative, as BERT deals with parts of words which do not necessarily carry a lot of meaning. However, the TF-IDF representation can show some topic separation. The most important tokens when using two and four clusters respectively are:

English

Cluster 1: https, refugees, women, immigration, illegal, immigrant, migrants, woman, men, buildthatwall

Cluster 2: bitch, fuck, ass, fucking, hoe, cunt, like, whore, stupid, women

Spanish

Cluster 1: puta (whore), co, https, zorra (female fox/slut), si (yes/if), callate (shut up), cállate (shut up), hijo (son), madre (mother), mujer (woman)

Cluster 2: perra (bitch), cállate (shut up), callate (shut up), mereces (deserve), si (yes/if), vos (you), maldita (damned), amiga (friend), voy (going), re

English

Cluster 1: buildthatwall, maga, buildthewall, illegal, realdonaldtrump, trump, nodaca, illegals, noamnesty, wall

Cluster 2: https, refugees, immigrant, migrants, immigration, people, immigrants, woman, time, amp

Cluster 3: bitch, whore, cunt, fuck, ass, fucking, hoe, stupid, like, slut
Cluster 4: women, men, rape, hysterical, woman, like, https, just, don, know

Spanish

Cluster 1: zorra (female fox/slut), si (yes/if), acoso (harrassment), polla (cock), mujer (woman), arabe (arab), árabe (arab), escoria (human waste), mujeres (women), guarra (slut)
Cluster 2: puta (whore), callate (shut up), hijo (son), cállate (shut up), madre (mother), mereces (deserve), mierda (shit), boca, (mouth), si (yes/if), novia (girlfriend)
Cluster 3: perra (bitch), cállate (shut up), callate (shut up), mereces (deserve), si (yes/if), vos (you), maldita (damned), amiga (friend), voy (going), re
Cluster 4: co, https, inmigrantes (immigrants), refugiados (refugees), puta (whore), si (yes/if), árabes (arabs), indocumentados (undocumented), vía (via), acoso (harassment)

4.6.1.2 Clustering English Training Data Sets

When clustering is applied to the combined data from all three English sources, the results are evaluated both based on the source and on the hate vs non-hate speech labels. The resulting purity and inverse purity scores are summarized in Table 4.12 for data source and Table 4.13 for hate speech. Baselines shown in grey are calculated based on a random cluster assignment. Additionally, as outlined in Section 3.8.1.2, the distributions by cluster for each of the groups of interest are calculated.

Data Representation	Clusters	Purity	Inverse Purity
Baseline	3	0.39	0.34
TF-IDF	3	0.53	0.72
BERT	3	0.41	0.68

Table 4.12: Clustering results for data source in English training data.

Data Representation	Clusters	Purity	Inverse Purity
Baseline	2	0.54	0.51
TF-IDF	2	0.53	0.87
BERT	2	0.50	0.72

Table 4.13: Clustering results for hate speech in English training data.

For data source purity and inverse purity scores are higher than the baseline with TF-IDF showing the better results. Additionally, the TF-IDF distribution by cluster for data source label seen in Figure 4.4 is the only one showing substantial deviation for one of the labels – namely Waseem & Hovy.

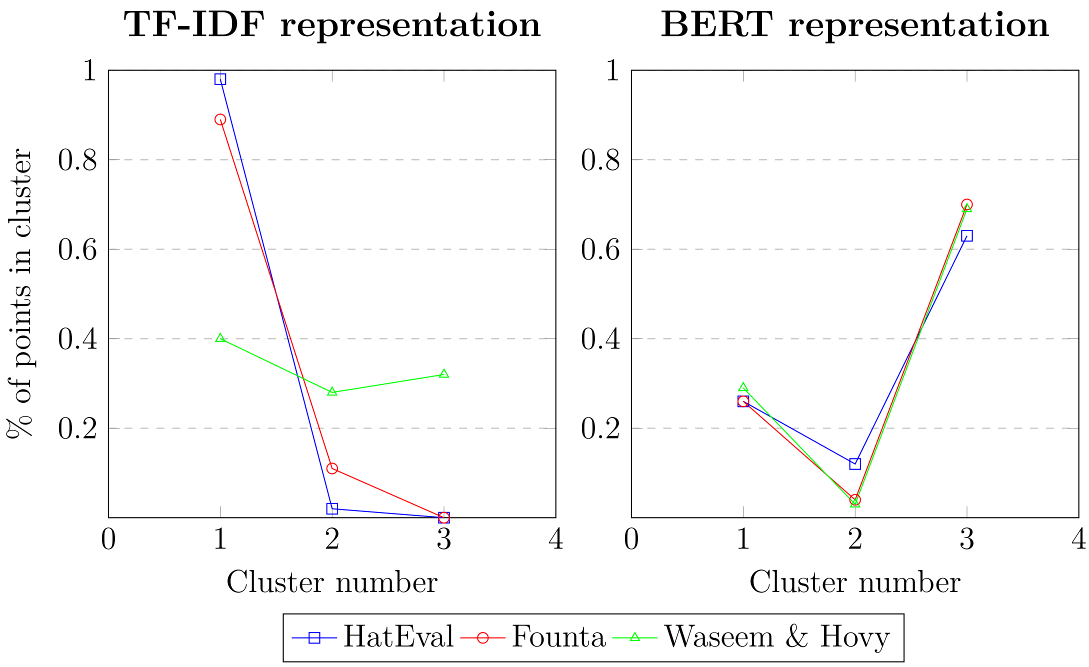

Figure 4.4: Distribution of tweets within all available English data sets in each cluster.

For hate speech clustering experiments, the purity score remains close to the baseline, while the inverse purity shows a significant increase. The distributions by cluster seen in Figure 4.5 show a balanced distribution between hate speech and non-hate speech examples in each cluster.

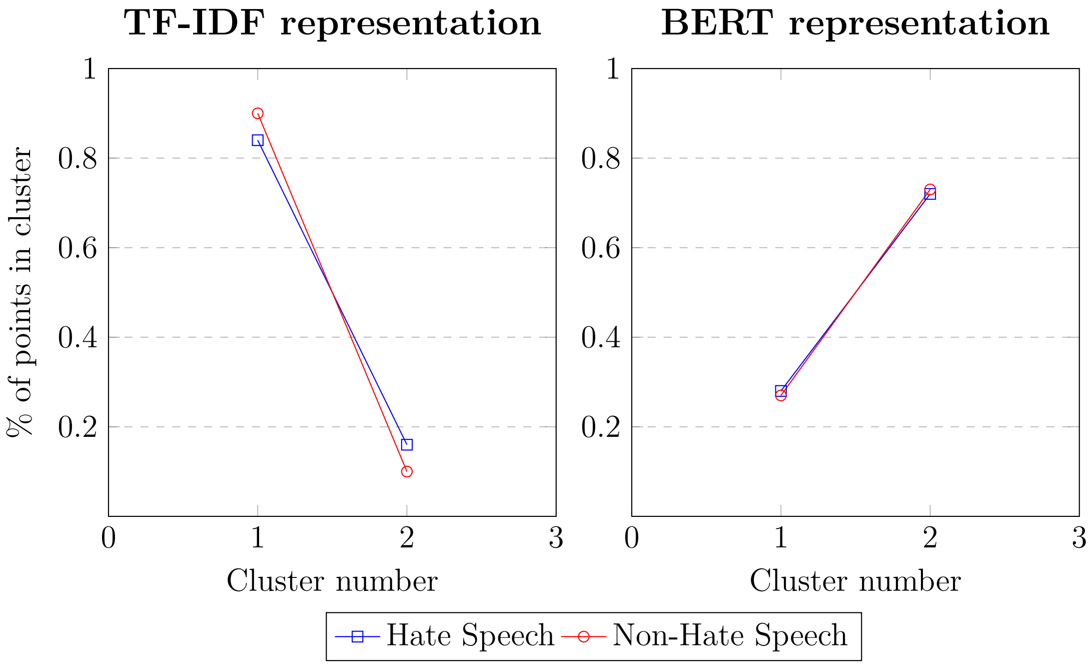

Figure 4.5: Distribution of hate and non-hate tweets within all available English data set in each cluster.

4.6.2 Classifying English Data Sets

In order to further investigate the possible differences between the English data sets, a classifier based on multilingual BERT is trained in order to predict the data source. This model has an accuracy of 89.78 percent with a standard deviation of 0.73 based on 10 runs. The confusion matrix shown in Table 4.14 also has high diagonal values.

		True Label		
		HatEval	Waseem & Hovy	Founta
Predicted Label	HatEval	104 (1)	1 (1)	4 (1)
	Waseem & Hovy	3 (2)	144 (2)	9 (2)
	Founta	8 (4)	16 (4)	111 (4)

Table 4.14: Data source classification results. Showing number of data points having a particular label vs the model prediction. Standard deviation on those counts are also shown in parentheses.

5

Discussion

In this section the significance of the results previously shown are explored in much further detail, analyzing the possible underlying reasons for the observed performance and focusing on some examples of the data augmentation techniques.

5.1 Cross-Lingual Zero-Shot Learning

The observed drop in recall that drives the drop in accuracy points to a more lenient decision-making in the test data – i.e. a lot of the second language's hate speech tweets are labeled as non-hate speech. One reason for this could be that the hate speech class is seen as the "specific class", i.e. what the model is looking for, and whenever an input does not match any of the already seen patterns it is labelled non-hate speech.

Additionally, the cross-language results show that the significant drop in performance is independent of the original language used for training. Some correlation between performance and language group is observed (e.g. Spanish on Portuguese performs better than Spanish on Arabic), however, these differences seem to be dominated by the general drop for all languages. The results of the HatEval data sets show that even data that is gathered in a similar manner carries enough differences in its distributions to hinder the cross-lingual zero-shot approach. It is not clear whether the issues are caused by BERT's ineffectiveness to generalize between languages or inconsistency of the data. There is research in the area suggesting low annotator agreement could be a contributing factor for the bad performance of the models [21].

Furthermore, the English on English results point to a general lack of similarity between different data sets, independent of language. Even without the need for learning a mapping between languages, the differences are substantial enough. This can be seen as further evidence that performance issues could be attributed to the inconsistency of the data rather than inability to learn cross-language mappings.

This is also supported by Founta's and Waseem & Hovy's cross-lingual zero-shot results. All three English data sets display a considerable difference in performance. Therefore, performance seems to be highly influenced by the specific data set used for training. Additionally, it can be seen that Founta does not suffer from the same recall drop as the HatEval model. This data set focuses on a broader spectrum of hate speech targets. This could support the argument that the more varied examples that are introduced to the model result in more varied patterns that are being learned.

5.2 Data Augmentation

DA seems to enhance performance of small training data sizes. However, as the original training data size is increased, this effect is reduced until lost completely. A common issue found within all data augmentation techniques used is the fact that the augmented sentence can lose its grammatical structure. Therefore, the meaning of the sentence can be obfuscated. This is a severe problem in this project, since hate speech requires a very strict and specific structure. In other words, hate speech requires a target and focused intention. Losing any of that lessens the severity of the statement.

5.2.1 TF-IDF Synonym Replacement

A sample of TF-IDF synonym replacement examples shows that there are a few issues with this method. In some cases, especially with shorter tweets, it can add an unchanged version of the sentence due to not finding a synonym for any of the words. Another issue that is observed is that there are cases where a word can have two meanings. This could be mitigated using a word sense disambiguator. However, when working with a low resource language it is highly likely that there is no readily available word sense disambiguator. Additionally, this is a difficult research problem in itself and falls outside the scope of this project.

An example of TF-IDF synonym replacement can be seen below:

Label: *Non-Hate Speech*

Original Sentence: *tu eres el menos indicado para hablar de españa como patria porque la odiaslargate de una puta vez con tu jefe maduro gilipollas*
Original Translation: *You are the one least indicated to speak about Spain as a nation since you hate it get the fuck out at once with your asshole chief Maduro*
New Sentence with TF-IDF WR: *tu eres el menos indicado para pronunciar de españa como país de origen porque la odiaslargate de una pelandusca vez con tu patrón maduro gilipollas*
New Translation with TF-IDF WR: *You are the least indicated to pronounce of Spain as a country of origin because you hate it get out at whore once with your asshole boss maduro*

As can be seen from this example, what initially is an aggressive tweet to anybody now feels slightly more pointed towards immigrants. This is mostly due to how the word "patria" which means nation is replaced by "país de origen" which means country of origin. Even though they are synonyms, the context around it gives it an anti-immigrant bias that is not present in the original tweet. Introducing phrases with racist or sexist connotations, could render the 'non-hate speech' label invalid. It is also a case where the curse word "puta" which originally is used as "fucking", is replaced by a word that can only mean "whore", making the new tweet more aggressive.

Another example found is:

Label: *Non-Hate Speech*

Original Sentence: *callate infeliz hijo de puta sos una mierda tenembaun sos un hijo de puta y ahora te haces el solidario andate a la puta que te pario mierda sorete de periodista*
Original Translation: *shut up unhappy son of a bitch you are a piece of shit Tenembaun you're a son of a bitch and now you're pretending to be supportive go to the whore who bore you shit piece of shit of a journalist*
New Sentence with TF-IDF WR: *callate infeliz hija de buscona sos una mierda tenembaun sos un ninzzo de buscona y ahora te haces el solidario andate a la buscona que te pario mierda sorete de diarista*
New Translation with TF-IDF WR: *shut up unhappy daughter of a whore you are a piece of shit Tenembaun you're a child of a whore and now you pretend to be supportive go to the whore who bore you shit piece of shit of a journalist*

Even though in this specific case the translations are quite similar, the replacement of "hijo" (son) to "hija" (daughter) makes the example sound more sexist than originally intended. Another issue is related to the commonly used idiom "hijo de puta", which would most closely translate to "son of a bitch" and is typically used in unofficial speech without being understood literally. Once "puta" is replaced with "buscona", the phrase reads much more literal. Even though both words are synonyms, the first one is part of a phrase that is colloquially used, while the second one makes it sound much more aggressive than what is intended. However, the label of the augmented example does not seem to change.

5.2.2 Word Dropout

While observing the tweets created by using word dropout it is seen that there are cases where a sentence's meaning could vary depending on which words are taken out. For example,

Label: *Hate Speech*

Original Sentence: *quiero vivir en suecia lastima que no soy un arabe de mierda*
Original Translation: *I want to live in Sweden shame I am not a shitty arab*
New Sentence with Dropout: *vivir suecia lastima que no soy de mierda*
New Translation with Dropout: *to live Sweden shame I am not a shitty*

In this specific case it can be seen that the anti-immigrant sentiment that is found in the original sentence is gone and now does not make much sense at all. Due to the general loss of grammatical structure the 'hate speech' label could be invalidated.

However, even though some sentences are shown to stop making grammatical sense, there are quite a few where the meaning is not lost and the label is intact.

An example is,

Label: *Hate Speech*

Original Sentence: *mis tios diciendo que ines arrimadas es una zorra a ver callate la boca*
Original Translation: *my uncles saying that ines arrimadas is a slut let's see shut your mouth*
New Sentence with Dropout: *mis tios diciendo que ines arrimadas es una zorra a callate boca*
New Translation with Dropout: *my uncles saying that ines arrimadas is a slut shut your mouth*

5.2.3 Back-Translation

While working with back-translation there are several issues that are seen when augmenting Twitter data. The first problem is the lack of correct punctuation making it very difficult to correctly translate what is written. This can be seen in the following example,

Label: *Hate Speech*

Original Sentence: *quiero vivir en suecia lastima que no soy un arabe de mierda*
Original Translation: *I want to live in Sweden, shame I'm not a shitty arab*
New Sentence with Back-Translation: *Yo vivo en Suecia, pero mi dolor no es el maldito Arabe*
New Translation with Back-Translation: *I live in Sweden, but my hurt is not the damn Arab*

As can be seen in the example, gone is the anti-immigrant sentiment that is present in the original sentence. In its place is a sentence that carries a completely opposite meaning, as it is now defending the same group that is originally targeted. Thus, making the original label incorrect for the augmented document.

Meanwhile, the next example has a grammatical error due to the lack of a space, while also having the issue of a proper noun being a common noun as well.

Label: *Non-Hate Speech*

Original Sentence: *tu eres el menos indicado para hablar de españa como patria porque la odiaslargate de una puta vez con tu jefe maduro gilipollas*
Original Translation: *You are the one least indicated to speak about Spain as a nation since you hate it get the fuck out at once with your asshole chief Maduro*
New Sentence with Back-Translation: *son los menos propensos a hablar de España como pais, porque cogida odiaslargate con su culo adultos jefe*
New Translation with Back-Translation: *They are the least likely to talk about Spain as a country, because "taken" hate it get out with your asshole adults chief*

As can be seen, the second half of the sentence makes no sense due to the lack of space between "odias" and "largate". Since they are treated as one word it is impossible to translate the sentence correctly resulting in a barely understandable sequence of words. While the original tweet is referencing Venezuelan president Nicolas Maduro, the translator mistakes that last name since it is also the word in Spanish for "mature". Finally, the reason "taken" is in quotation marks is because the translator changes "una puta vez" which means "at fucking once" into "cogida", which is slang for having sexual intercourse in some regions, but in most regions is seen as meaning "grabbed". However, since the sentence makes no sense with either definition, one can not understand which definition is used from the context. For this specific case the label would not change, however if a hate speech augmented example presented a similar loss in meaning its label would change. Even without the hate speech label issue it is unclear whether this example of back-translation could be considered good augmentation, since it no longer seems plausible to be a tweet written by a person.

Another issue with Twitter data is the informality that most users tend to use. This leads to several acronyms that the translator cannot translate, thus making the output sentences somewhat patchy.

Label: *Hate Speech*

Original Sentence: *como no te mato por hija de puta y por puta provocadora perra mereces la muerte sucia hdp*
Original Translation: *How do I not kill you daughter of a bitch and for provocative whore bitch you deserve a dirty death hdp*
New Sentence with Back-Translation: *como no se mata a una perra puta zorra provocadora y digno de HDP sucia muerte*
New Translation with Back-Translation: *how can it not be killed a bitch whore slut provocative and worthy of HDP death dirty*

In this case, "hdp" is an acronym that stands for "hijo de puta" or "son of a bitch". As can be seen in the final sentence the translator is not sure where to put the original acronym, placing it at a random position within the sentence. Even though the sentence has lost the connection to that particular acronym, the rest of the sentence retains enough structure for the label to be unchanged.

5.3 Translation of English Training Data

A further experiment is done to determine the effect of translating a resource rich language into the target language and use that as training examples. As can be seen from the comparison in Table 4.9 the language which benefits the most from this translation is Spanish, where accuracy increases from 56.03% to 61.72%, with all other languages showing no statistically significant change in this metric. However, all languages show an improved F1 score, with Spanish having a substantially higher increase. These could be explained by the difference in the translation model performance, i.e. the English to Spanish translations could work better than the other translations due to its widespread use.

The translated examples suffer from similar issues as the ones outlined in the back-translation section. These include loss of grammatical structure, inability to translate misspelled words, etc.

5.4 Data Set Analyses

The clustering results show that the data is not separated into neither the hate speech vs. non-hate speech labels nor the three English data source, as other topics prove to be more prevalent in the data. However, the good classification performance suggests that there are some patterns that are unique for each data source, allowing the BERT model to learn that mapping.

5.4.1 Clustering of HatEval Data Sets

Both the Spanish and English HatEval data sets are clustered in order to explore any cohesive groups present in the data. These groups' alignment with the hate speech label is also explored.

As can be seen from Tables 4.10 there is no strong correlation when only two clusters are used. This is supported by the most descriptive words. As can be seen Section 4.6.1.1. the clusters that are found are: one discussing immigration, and one discussing women. Therefore, it seems those topics have a more distinct vocabulary than the hate and non-hate speech groups.

When four clusters are used, the TF-IDF representation, shows an increase of 18% in the purity score from the baseline. From the most descriptive words two immigration clusters are present with one of them aligned with Trump-related hashtags; additionally, two clusters discussing women are present with one of them seeming more aggressive due to its heavy emphasis on curse words. The political alignment of one of the clusters and the high aggression of the other could be contributing to the stronger correlation with hate speech.

On the other hand, the results for Spanish remain consistently low for both representations and for both numbers of clusters. As can be seen from the word representations, it is harder to identify specific topics. However, there could be other patterns that have an effect on the clustering results. For example, when using four clusters, cluster one contains the term "polla" that is common slang used in Spain, whereas cluster four contains the term "indocumentado" that tends to be more aligned with users from Central and North America.

5.4.2 Clustering English Training Data Sets

When clustering the combined English data sets, the purity score for the hate speech label is low for both representations. The inverse purity is substantially higher than the baseline. However, this is due to one cluster containing most of the data from both the hate speech and the non-hate speech groups, as can be seen in Figure 4.5.

When comparing the clustering results to the data source label, the BERT representation shows a purity score close to the baseline with a high inverse purity.

However, the high inverse purity can again be explained by 60-70% of the data from each source going into the same cluster.

When using TF-IDF representation, the purity and inverse purity scores are both higher than the baseline. It can be seen in Figure 4.4 that one of the data sources – Waseem & Hovy – shows a slightly different cluster distribution with one of the clusters containing predominantly Waseem & Hovy data. From the most descriptive words, it can be seen that this cluster contains terms related to the Australian reality TV show "My Kitchen Rules", like its name, related hashtags and contestants' names. This can be attributed to the specific data collection process used. An initial search using common slurs is performed. In the gathered data frequent terms are identified and used to perform the data extraction. The "My Kitchen Rules" topic is identified as prompting sexist tweets and is used as a query in order to collect more data, explaining the presence of this cluster.

5.5 Classifying English Data Sets

The classification results of the combined English data sets, unlike the clustering results discussed above, show that a BERT-based classifier is good at distinguishing the source data sets. The accuracy of the model is 89.78% and the confusion matrix shows no preference for any of the sources – i.e. performs equally well on all data sets. This could explain the poor transferability of the model between different English data sets seen in Table 4.3. In other words, the BERT model is capable of finding specific data set patterns before generic hate speech ones.

6

Conclusion

As social media platforms become more global and hate speech rhetoric keeps rising within them, the need for hate speech detection in languages other than English becomes much more imperative. The methods proposed in this work show an indication that these could be a plausible solution to the task.

Cross-lingual zero-shot results suggest that knowledge transfer is occurring, as most of the models perform better than a random classifier. However, a substantial dependency between performance and training data set used is observed. When using the three English data sets, each one gives substantially different results in the cross-lingual setting. Additionally, the English on English testing suggests these results reflect differences in data sets rather than only incapability of cross-lingual knowledge transfer.

Word dropout and back-translation seem to be effective data augmentation techniques for small data sets. However, these perform poorly for larger data sets. This suggests that these are good for when resources available are very low, but lose their suitability as resources increase. Meanwhile, TF-IDF synonym replacement shows no improvement regardless of the amount of original training data. This could be contributed to this method being more susceptible to losing the original meaning of the sentence.

When exploring the proposed solutions several issues stemming from the data itself are found, these relate both to the specific topic of hate speech and its Twitter nature.

Since hate speech has no official definition, each data set is labeled using different annotation guidelines. Additionally, the varying gathering approaches used by each data set lead to each one sub-sampling a very different segment of the hate speech examples in Twitter. One of the consequences of this variety, is the absence of a single well explored data set that can be used as a benchmark for hate speech detection. As mentioned before, hate speech is very dependent on a strict structure, making data augmentation techniques less effective. One of the obstacles being the corrupted grammatical structure. This could be offset by using more grammatically centered solutions, such as [23]. These issues are further exacerbated by the use of Twitter data and its particular use of acronyms and punctuation.

In general, when working with hate speech data in a low resource language, the best approach is to gather and annotate a data set internally. However, the annotation guideline and the bias it produces must be played close attention to. If the gathered data set is too small, data augmentation techniques can be considered. Presuming that no data set can be gathered, a cross-lingual zero-shot approach could be applied. In that case, the training data selected will have a great influence on the performance of the model, so the gathering process and annotation guideline need to be closely examined. Additionally, if regardless of the lack of resources for the language of interest, there exists a good translation model this could be used to further aid performance.

Glossary

Annotation The process of manually assigning labels to data. Usually performed by crowd-sourcing.[1]

Classification The process of training a model to predict the class of a given data point. This is achieved by introducing a training data set with labeled examples to the model.[2]

Clustering The process of finding cohesive groups whose members are similar in some way.[3]

Cross-Lingual Learning A type of learning that attempts to learn patterns across languages. This could be used for learning language patterns from a rich corpus of data in one language and utilizing the learned patterns for another.[4]

Data Augmentation A method for producing additional inputs by transforming existing ones.[5]

Machine Learning A study of statistical models used for inferring patterns from data. This allows computer systems to perform specific tasks without using a set list of instructions.[6]

Modelling The process of training a specific statistical model to learn patterns from available data.[7]

Natural Language Processing A sub-field of computer science that deals with how computers interact with human languages. It is concerned with the task of training computers to process and analyse human language data, e.g. machine translation, speech recognition, etc.[8]

[1] https://lionbridge.ai/articles/data-annotation-machine-learning/

[2] https://www.edureka.co/blog/classification-in-machine-learning/

[3] https://home.deib.polimi.it/matteucc/Clustering/tutorial$_{h}tml/$

[4] https://ruder.io/unsupervised-cross-lingual-learning/

[5] https://nanonets.com/blog/data-augmentation-how-to-use-deep-learning-when-you-have-limited-data-part-2/

[6] https://towardsdatascience.com/introduction-to-machine-learning-f41aabc55264

[7] https://elitedatascience.com/model-training

[8] https://towardsdatascience.com/introduction-to-natural-language-processing-for-text-df845750fb63

Neural Networks A machine learning approach for training computers to perform some task by analysing labeled examples of data. These networks consist of several simple processing nodes that are densely interconnected, which are typically organized in layers.[9]

Transfer Learning An approach which stores knowledge gained from solving a problem A and then uses it to solve a different problem B that has some similarities with the original one.[10]

Zero-Shot Learning A type of learning where the classes found within the training data and the test data share no overlap.[11]

[9] https://victorzhou.com/blog/intro-to-neural-networks/

[10] https://towardsdatascience.com/a-comprehensive-hands-on-guide-to-transfer-learning-with-real-world-applications-in-deep-learning-212bf3b2f27a

[11] https://towardsdatascience.com/applications-of-zero-shot-learning-f65bb232963f

Bibliography

[1] GLUE Benchmark leaderboard. https://gluebenchmark.com/leaderboard/. Accessed: 2020-02-09.

The General Language Understanding Evaluation (GLUE) Benchmark is a collection of resources for evaluating natural language understanding systems. The leaderboard contains the best-performing models at a specific time.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2014.

The paper proposes an improvement on the previously available translation models. Proposing to extend these models by allowing them to automatically soft search parts of the source sentence that are relevant to predicting a target word.

[3] Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, and Manuela Sanguinetti. SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 54–63, Minneapolis, Minnesota, USA, June 2019. Association for Computational Linguistics.

This paper describes the data set created for the SemEval 2019 Task 5 regarding the detection of hate speech. It is comprised of Spanish and English twitter data that is manually annotated as containing hate speech or not. This is data set used for this project and it is also used in [15]

[4] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078, 2014.

In this paper a novel neural network model called RNN Encoder-Decoder that consists of two recurrent neural networks (RNN) is proposed. One RNN encodes a sequence of symbols into a fixed-length vector representation, and the other decodes the representation into another sequence of symbols.

[5] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems (MCSS)*, 2(4):303–314, December 1989.

In this paper, it is shown that any function can be arbitrarily well approximated by continuous feedforward neural networks with only a single internal, hidden layer and any continuous sigmoidal nonlinearity.

[6] Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. Automated hate speech detection and the problem of offensive language. In *Eleventh international aaai conference on web and social media*, 2017.

This paper uses a data set comprised of Twitter data manually annotated in three categories: hate speech, offensive but not hate speech or neither. This data set can be used for Hate Speech detection tasks, see [15]

[7] Ona de Gibert, Naiara Perez, Aitor García-Pablos, and Montse Cuadros. Hate speech dataset from a white supremacy forum. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 11–20, Brussels, Belgium, October 2018. Association for Computational Linguistics.

This paper describes a hate speech data set extracted from Stormfront (a white supremacist forum). A custom annotation tool has been developed for manually labelling the data. This tool allows the annotator to read the context of a sentence before a label is given. See [15]

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

This is the paper which introduces novel NLP architecture based on transformers and a masked language model called BERT.

[9] Atefeh Farzindar, Diana Inkpen, and Graeme Hirst. *Natural Language Processing for Social Media: Second Edition.* Morgan & Claypool Publishers, 2nd edition, 2017.

This book details known issues and specifics of dealing with social media text data in modelling tasks.

[10] Paula Fortuna, João Rocha da Silva, Juan Soler-Company, Leo Wanner, and Sérgio Nunes. A hierarchically-labeled Portuguese hate speech dataset. In *Proceedings of the Third Workshop on Abusive Language Online*, pages 94–104, Florence, Italy, August 2019. Association for Computational Linguistics.

This paper describes a data set in Portuguese annotated for hate speech. It is comprised of Portuguese Twitter data that is manually annotated as containing hate speech or not.

[11] Antigoni-Maria Founta, Constantinos Djouvas, Despoina Chatzakou, Ilias Leontiadis, Jeremy Blackburn, Gianluca Stringhini, Athena Vakali, Michael Sirivianos, and Nicolas Kourtellis. Large scale crowdsourcing and characterization of Twitter abusive behavior. *CoRR*, abs/1802.00393, 2018.

This paper describes a methodology that utilizes crowd-sourcing to label a large-scale collection of tweet with a set of abuse related labels.

This method produced a large English Twitter data set labelled for hate speech, which is used in the current work.

[12] Aitor Gonzalez-Agirre, Egoitz Laparra, and German Rigau. Multilingual central repository version 3.0: upgrading a very large lexical knowledge base. In *Proceedings of the 6th Global WordNet Conference (GWC 2012)*, Matsue, 2012.

This paper describes the update of the Multilingual Central Repository, in order to include the Spanish language into the system.

[13] Muhammad Okky Ibrohim and Indra Budi. Multi-label hate speech and abusive language detection in Indonesian Twitter. In *Proceedings of the Third Workshop on Abusive Language Online*, pages 46–57, Florence, Italy, August 2019. Association for Computational Linguistics.

This paper introduces an Indonesian Twitter data set annotated for the target, category and level of hate speech.

[14] Ritesh Kumar, Atul Kr. Ojha, Shervin Malmasi, and Marcos Zampieri. Benchmarking aggression identification in social media. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 1–11, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics.

This paper describes a competition that uses data sets that comprised of Facebook and Twitter data in both English and Hindi. These sets are labelled with three levels of aggression (overtly, covertly or not aggressive) and can be used for Hate Speech Detection tasks, see [15].

[15] Sean MacAvaney, Hao-Ren Yao, Eugene Yang, Katina Russell, Nazli Goharian, and Ophir Frieder. Hate speech detection: Challenges and solutions. *PloS one*, 14(8), 2019.

This paper compares several different models, such as Support Vector Machines, Neural Ensmebles and BERT. The evaluation is done on four different data sets - Stormfront[7], TRAC[14], Hatebase Twitter[6], HatEval [3], all of which tackle the task of hate speech detection on media. On the HatEval dataset it was found that BERT outperformed the other models, while also performing consistently well for the other data sets.

[16] Mainack Mondal, Leandro Araújo Silva, and Fabrício Benevenuto. A measurement study of hate speech in social media. In *Proceedings of the 28th ACM Conference on Hypertext and Social Media*, HT '17, pages 85–94, New York, NY, USA, 2017. ACM.

This paper provides a systematic large-scale measurement study of hate speech in online social media. The data sets used are from two social media applications, Whisper and Twitter. This study explores the broad spectrum of groups affected and the contributing factors.

[17] Marzieh Mozafari, Reza Farahbakhsh, and Noel Crespi. A BERT-Based transfer learning approach for hate speech detection in online social media. 12 2019.

This paper introduces a transfer learning approach based on BERT for detecting hate speech. It is evaluated on two publicly available Twitter data sets.

[18] Hala Mulki, Hatem Haddad, Chedi Bechikh Ali, and Halima Alshabani. L-HSAB: A Levantine Twitter dataset for hate speech and abusive language. In *Proceedings of the Third Workshop on Abusive Language Online*, pages 111–118, Florence, Italy, August 2019. Association for Computational Linguistics.

In this paper, the first publicly-available Levantine Hate Speech and Abusive (L-HSAB) Twitter dataset for automatic detection of online Levantine toxic contents.

[19] Telmo Pires, Eva Schlinger, and Dan Garrette. How multilingual is multilingual bert? *CoRR*, abs/1906.01502, 2019.

This paper describes a version of BERT, called M-BERT, which is trained on multilingual data. This corpus consists of 104 languages and evaluates its performance on cross-lingual tasks. Showing a high level of transfer ability from one language to another.

[20] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019.

This paper introduces a transformer based unidirectional transfer learning model, called GPT. This achieved state-of-the-art results before the model used in this project.

[21] Björn Ross, Michael Rist, Guillermo Carbonell, Benjamin Cabrera, Nils Kurowsky, and Michael Wojatzki. Measuring the reliability of hate speech annotations: The case of the European refugee crisis. *CoRR*, abs/1701.08118, 2017.

This paper explores whether hate speech can be reliably annotated. It is shown that annotation reliability is very low overall.

[22] Sebastian Ruder, Anders Søgaard, and Ivan Vulić. Unsupervised cross-lingual representation learning. In *Proceedings of ACL 2019, Tutorial Abstracts*, pages 31–38, 2019.

This tutorial highlights key insights and takeaways regarding unsupervised multilingual deep models.

[23] Gözde Gül Şahin and Mark Steedman. Data augmentation via dependency tree morphing for low-resource languages. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5004–5009, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

This paper proposes two data augmentation techniques that utilize dependency trees, i.e. incorporating the grammatical structure of the text.

[24] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.

This paper introduces dropout – a novel approach for neural network regularization. In order to reduce the risk of overfitting, a number of units along with their connections are randomly dropped from the neural network during training.

[25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

This paper introduces the novel transformer architecture, which is based on attention without the need for recurrence.

[26] Zeerak Waseem and Dirk Hovy. Hateful symbols or hateful people? Predictive features for hate speech detection on Twitter. In *Proceedings of the NAACL Student Research Workshop*, pages 88–93, San Diego, California, June 2016. Association for Computational Linguistics.

This paper produces hate speech annotations for a publicly available corpus of Tweets. These annotations are based on a list of criteria founded in critical race theory.

[27] Jason W. Wei and Kai Zou. EDA: easy data augmentation techniques for boosting performance on text classification tasks. *CoRR*, abs/1901.11196, 2019.

This paper introduces four techniques for text augmentation: syonym replacement, random insertion, random swap, and random deletion. These are evaluated on five text classification tasks.

[28] Qizhe Xie, Zihang Dai, Eduard H. Hovy, Minh-Thang Luong, and Quoc V. Le. Unsupervised data augmentation. *CoRR*, abs/1904.12848, 2019.

This paper explores how different types of data augmentation affect performance. In particular, back-translation and word replacing with TF-IDF are used for text classification tasks. The BERT model was used on multiple data sets showing lower error rates when Data Augmentation was used.

[29] Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola. *Dive into Deep Learning*. 2019. `http://www.d2l.ai`.

This book introduces basic concepts in Deep Learning, including Recurrent Neural Networks and Machine Translation.

A

Appendix 1

Train	Test	L±std	ACC±std	H±std	RCL±std	PRC±std	F1±std
EN	EN	69.52±2.22	73.9±2.12	78.1±1.99	66.6±7.03	78.1±1.34	71.65±3.77
ES	ES	76.78±1.65	80.67±1.49	84.48±1.42	80.55±3.52	80.83±1.95	80.62±1.74

Table A.1: Performance of English Base (EN) and Spanish Base (ESP).

Train	Test	L±std	ACC±std	H±std	RCL±std	PRC±std	F1±std
EN	EN	69.52±2.22	73.9±2.12	78.1±1.99	66.6±7.03	78.1±1.34	71.65±3.77
EN	ES	51.2±1.56	56.03±1.5	60.85±1.46	26.05±7.44	66.15±4.6	36.52±7.66
EN	AR	52.85±2.33	57.65±2.26	62.4±2.26	23.0±6.6	76.06±7.35	34.64±7.83
EN	ID	46.62±1.43	51.58±1.38	56.55±1.41	7.55±3.67	62.24±11.57	13.22±6.0
EN	PT	50.72±1.94	55.55±1.87	60.35±1.82	17.35±4.95	74.04±5.97	27.73±6.59
ES	ES	76.78±1.65	80.67±1.49	84.48±1.42	80.55±3.52	80.83±1.95	80.62±1.74
ES	EN	49.25±1.51	54.18±1.45	59.05±1.36	19.55±5.72	64.16±4.2	29.44±6.67
ES	AR	45.75±0.56	50.72±0.59	55.72±0.59	8.25±3.02	55.82±5.03	14.12±4.48
ES	ID	46.55±0.8	51.52±0.82	56.45±0.88	9.3±3.39	61.42±6.95	15.84±5.08
ES	PT	55.58±1.48	60.35±1.42	65.1±1.42	36.45±4.98	70.04±2.59	47.69±4.28

Table A.2: Performance of English and Spanish base on other languages.

Train	Test	L±std	ACC±std	H±std	RCL±std	PRC±std	F1±std
HatEval	HatEval	69.52±2.22	73.9±2.12	78.1±1.99	66.6±7.03	78.1±1.34	71.65±3.77
HatEval	WH	53.73±1.79	57.75±1.79	61.76±1.79	28.18±6.08	68.81±2.1	39.58±6.27
HatEval	F	53.5±1.67	57.87±1.62	62.17±1.57	22.83±4.74	76.46±2.42	34.87±5.82
WH	HatEval	55.32±1.47	60.08±1.47	64.82±1.47	57.85±7.78	61.09±3.2	58.92±3.01
WH	WH	79.41±0.72	82.55±0.72	85.62±0.64	79.97±2.62	84.38±2.16	82.05±0.85
WH	F	57.83±2.32	62.11±2.29	66.36±2.29	37.98±9.29	74.0±2.71	49.35±7.67
F	HatEval	54.48±1.19	59.25±1.16	64.0±1.16	75.05±5.35	57.13±1.35	64.75±1.45
F	WH	62.23±1.45	66.13±1.38	69.97±1.38	68.81±7.03	65.68±3.41	66.84±2.13
F	F	78.18±1.29	81.64±1.19	84.98±1.11	78.3±3.43	84.07±2.64	80.99±1.36

Table A.3: Results obtained using different English Sets.

Train	Test	L±std	ACC±std	H±std	RCL±std	PRC±std	F1±std
F	F	78.18±1.29	81.64±1.19	84.98±1.11	78.3±3.43	84.07±2.64	80.99±1.36
F	ESP	49.5±2.25	54.46±2.19	59.33±2.15	32.92±7.79	58.01±3.64	41.43±6.52
F	ARA	51.04±1.8	55.83±1.78	60.62±1.74	38.42±10.41	58.89±0.71	45.77±7.13
F	IDN	49.79±1.61	54.67±1.62	59.54±1.66	26.75±6.34	60.58±2.99	36.69±6.1
F	POR	53.92±2.2	58.71±2.14	63.46±2.14	42.67±9.92	62.9±1.79	50.17±6.92

Table A.4: Cross-lingual zero-shot results obtained using Founta.

Train	Test	L±std	ACC±std	H±std	RCL±std	PRC±std	F1±std
WH	WH	79.41±0.72	82.55±0.72	85.62±0.64	79.97±2.62	84.38±2.16	82.05±0.85
WH	ESP	48.6±2.26	53.52±2.2	58.42±2.11	43.15±9.25	54.58±2.84	47.6±6.17
WH	ARA	50.75±2.15	55.62±2.06	60.4±2.02	34.6±16.04	60.73±3.69	41.66±12.19
WH	IDN	48.3±1.14	53.25±1.17	58.15±1.11	17.3±6.06	62.09±3.45	26.48±6.65
WH	POR	51.95±1.14	56.75±1.12	61.58±1.02	34.8±6.42	62.3±2.3	44.23±5.04

Table A.5: Cross-lingual zero-shot results obtained using Waseem & Hovy.

Initial Size	Final Size	L±std	ACC±std	H±std	RCL±std	PRC±std	F1±std
250	250	60.1±3.06	64.8±2.99	69.42±2.89	84.6±9.87	61.24±4.18	70.48±2.4
250	1000	64.38±3.35	68.97±3.2	73.43±3.07	83.65±5.12	65.1±4.07	72.98±1.67
250	2000	64.95±2.2	69.47±2.14	73.98±2.04	70.45±8.75	69.4±2.65	69.51±4.05
250	3600	65.65±2.45	70.15±2.35	74.6±2.28	77.35±11.06	68.09±3.17	71.75±4.98
1000	1000	72.17±2.47	76.35±2.32	80.45±2.21	85.55±5.2	72.72±4.25	78.36±1.38
1000	2000	72.82±2.63	76.97±2.47	81.02±2.27	82.15±4.49	74.82±3.85	78.13±1.74
1000	3600	69.0±2.22	73.43±2.11	77.62±1.97	75.45±7.39	72.87±3.47	73.81±3.1
2000	2000	76.0±1.55	80.02±1.49	83.85±1.4	80.8±5.62	79.81±2.75	80.11±2.13
2000	3600	75.4±2.35	79.4±2.21	83.3±2.05	81.7±8.06	78.42±2.94	79.7±3.22

Table A.6: Results obtained using word dropout.

Initial Size	Final Size	L±std	ACC±std	H±std	RCL±std	PRC±std	F1±std
250	250	60.1±3.06	64.8±2.99	69.42±2.89	84.6±9.87	61.24±4.18	70.48±2.4
250	1000	61.85±3.94	66.5±3.85	71.07±3.7	56.6±11.53	70.69±3.31	62.2±7.4
250	2000	64.18±2.35	68.72±2.26	73.28±2.17	70.85±6.26	68.34±3.71	69.3±2.27
250	3600	63.98±1.71	68.58±1.62	73.07±1.62	63.3±6.09	70.87±1.53	66.67±3.3
1000	1000	72.17±2.47	76.35±2.32	80.45±2.21	85.55±5.2	72.72±4.25	78.36±1.38
1000	2000	71.12±0.94	75.37±0.94	79.53±0.85	81.3±5.95	73.04±3.28	76.69±1.37
1000	3600	69.55±2.14	73.95±2.03	78.15±1.9	77.85±5.55	72.46±3.14	74.87±2.26
2000	2000	76.0±1.55	80.02±1.49	83.85±1.4	80.8±5.62	79.81±2.75	80.11±2.13
2000	3600	71.88±2.35	76.12±2.21	80.2±2.06	73.35±6.92	77.87±2.67	75.3±3.3

Table A.7: Results obtained using back-translation.

Train	Test	L±std	ACC±std	H±std	RCL±std	PRC±std	F1±std
EN	ES	51.12±1.59	56.03±1.5	60.78±1.5	26.05±7.44	66.15±4.6	36.52±7.66
ENtoES	ES	56.98±1.98	61.72±1.98	66.48±1.98	58.45±8.13	62.73±2.11	60.14±4.31
EN	AR	52.78±2.35	57.65±2.26	62.42±2.24	23.0±6.6	76.06±7.35	34.64±7.83
ENtoAR	AR	52.62±2.3	57.45±2.24	62.22±2.23	30.85±8.86	66.32±3.8	41.31±7.77
EN	PT	50.73±1.94	55.55±1.87	60.35±1.82	17.35±4.95	74.04±5.97	27.73±6.59
ENtoPT	PT	51.4±0.92	56.22±0.87	61.02±0.82	19.3±2.69	73.96±2.54	30.5±3.32
EN	ID	46.63±1.35	51.58±1.38	56.48±1.28	7.55±3.67	62.24±11.57	13.22±6.0
ENtoID	ID	47.42±1.55	52.4±1.53	57.37±1.5	17.15±9.2	59.89±10.75	25.23±10.13

Table A.8: Translated vs cross-lingual zero-shot results.

Train	Test	L±std	ACC±std	H±std	RCL±std	PRC±std	F1±std
EN	EN	88.75±0.93	91.58±0.78	94.22±0.67	95.96±1.29	88.77±1.2	92.22±0.71
EN	ES	64.85±2.01	69.42±1.93	73.88±1.85	53.74±5.19	80.53±2.32	64.28±3.47

Table A.9: Performance of English trained model for sentiment analysis.