Multilingual Hate Speech Detection

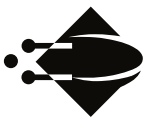
Various Solutions for Multilingual Offensive Speech and Hate Speech Detection in Social Media

Erida Nurçe – ernu@itu.dk,

Jorgel Këci – joke@itu.dk

Advisor: Leon Derczynski

Submitted: June 2020



IT University
of Copenhagen

Acknowledgments

We would like to first express our gratitude towards our supervisor, Leon Derczynski, for his constant support and contribution to this work during the fruitful discussions we have shared in the past semesters. This work would not have been the same if it wasn't for his expertise and enthusiasm, that motivated us to work even harder. Thank you Leon, it has been a pleasure working with you.

Furthermore, we would like to thank our friends and families that have always supported us and pushed us to be the most hard-working versions of ourselves. Your daily contribution does not go unnoticed.

Signatures

Erida Nurçe: _____

IT University of Copenhagen, June 2020

Jorgel Këci: _____

IT University of Copenhagen, June 2020

Abstract

This thesis aims to investigate abuse in online media, specializing in social media platforms. We set a special focus on experimenting with various multilingual hate speech detection settings and conduct five categories of experiments to tackle hate speech detection, namely: monolingual, bilingual, multilingual, knowledge transfer and zero shot or cross-lingual experiments. We also contribute with the creation of the largest annotated Albanian dataset for offensive and hate speech, which was annotated following the OffensEval schema.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The usage of internet and social media continues to grow in recent years, with statistical data showing that 4.8 billion people are using the internet at the start of 2020, accounting to 60% of the world's population now being online (**kemp_2020**). Furthermore, the Digital 2020: Global Digital Overview reports that 89% of internet users aged between 13 and 64 use social media apps each month (ibid.), testifying to the importance and significant impact social media now has in our daily life.

As the time people spend online follows an ever-growing trend, so does the information produced on the web, bringing with it both positive and negative consequences. Social media in itself largely contributes to the generation and exchange of information, with billions of people posting about their experiences, opinions and beliefs and interacting with each other on a regular basis. However, the freedom of online speech combined with the large amount of data being generated in the process has brought about new challenges to the digital space. These challenges are directly related to the misuse of these online platforms regarding offensive and/or hate speech in the forms of bullying, racism,

sexism, etc. The lack of control during content creation that results from the fundamental right of freedom of speech has to be countered with effective mechanisms for detecting and possibly removing such abusive content. As an end result, we aspire to create environments that are safe and welcoming for users of all genders, races, ethnicities, sexual orientations, etc.

1.1 Hate Speech Detection Challenges

While some forms of offensive speech are relatively easy to distinguish, hate speech per se has proven to be a more difficult topic. Its definition varies in different publications, and its perception is at times subjective and often affected by the reader's point of view as well as their upbringing, race, gender, etc. The article "Interrupted Social Peace: Hate Speech in Turkish Media", by Esra Arcan includes a thorough discussion of hate speech definitions from several academic papers and regulations (**arcan_2013**). In this thesis, we follow the definition of hate speech and offensive language described in "Automated Hate Speech Detection and the Problem of Offensive Language"(**Davidson_2017**):

"Hate speech is defined as language that is used to express hatred towards a targeted group or is intended to be derogatory, to humiliate, or to insult the members of the group."

Social media giants such as Facebook, Instagram, YouTube etc, have taken noticeable measures to prevent abusive speech by manually reviewing their content and deciding whether it should be censored or not (**twitter**), (**facebook**). Such initiatives involve data that is mainly in English, while less widespread languages are unable to profit from the latest advancements in the field.

1.2 Our Contribution

Our first contribution consists in the creation and publication of what is to our knowledge the largest Albanian dataset for offensive and hate speech. We also make use of this dataset to carry out various experiments related to offensive and hate speech detection, which is relatively unexplored for this language.

In addition, we aim to contribute with the creation of a multilingual mechanism towards the detection of offensive and hate speech. We introduce and carry out 5 different types of experiments where we want to investigate how already existing Machine Learning and Deep Learning algorithms behave when dealing with hate speech expressed in multiple languages. These experiments are categorized as Monolingual, Bilingual, Multilingual, Knowledge Transfer and Zero-Shot. The languages included in this work are the Albanian, English, Danish and Turkish language. We commence our work with the use of a Naive Bayes classifier, which will serve as a baseline for our further experiments and later compare it with a Bidirectional Long Short-Term Memory (BiLSTM) classifier. Lastly we explore the Bidirectional Encoder Representations from Transformers (BERT) and differentiate between results achieved by each model.

The experiments in this thesis are divided into 5 categories in order to answer the following questions:

1. How do these models behave in the monolingual environment?
2. How do these models behave when a mixture of languages is used to train and test them? Furthermore, is there a combination of languages which proves to be more effective?

3. Can these models learn from additional languages added during training to provide a better detection on specific languages?

4. How effective are these models in detecting hate speech in languages they have not been trained on?

5. How robust are these classifiers when training and testing with many languages at the same time?

6. How can word embeddings contribute to improving detection across different languages?

In the work carried out for this thesis, we achieve a state of the art F1 score of 0,80 when detecting offensive speech in the Albanian language. ** Add other contribution to the usage of such models with multilingual settings etc **Include the best scoring results here**

1.3 Document Structure

Having introduced the problem itself, the actions taken so far and what we want to achieve with this work, we structure the rest of the paper as follows. In chapter 2 , we firstly provide a literature review on the definitions and challenges when dealing with hate speech multilingually. Then we discuss the classifiers used for the task in hand along with which datasets that have been used. Lastly, we elaborate on the OffensEval Schema which we will be using when conducting our experiments. Chapter 3 describes the motivation behind the experiment in relation with the questions asked in Introduction and arguments about the choice of datasets and languages. In chapter 4, we firstly include information regarding the datasets formation and structure. Secondly,

we discuss details about the classifiers employed in this work, such as their structure, and lastly the necessary steps to prepare our data for the classifiers. Chapter 5 handles the types of experiments that we have constructed, the evaluation metrics we have based our performance on and the parameters used for each model to run the experiments. In chapter 6, we present our results and spur discussion related to the interesting findings of our classifiers. Lastly we conclude our work with a summary of our initial expectations for each experiment along with an analysis of our results, as well as possible improvements that can be achieved in future work in chapter 7. In appendix A we include an extensive collection of results obtained during the experiments conducted for this thesis.

Chapter 2

Related Work

2.1 Definitions

Offensive and hate speech in online media and especially in social media platforms has gained a lot of attention in recent years, with focus being set on creating efficient tools for detecting such content. Despite the progress being made in the field, a common definition of offensive and hate speech has yet to be established. Various publications provide their own interpretation that they follow in their work, while several legislations and regulations also elaborate on various definitions. A consensus on the definition of hate speech would positively contribute to research on hate speech, by making the task of annotation more reliable and easier, as mentioned in Measuring the Reliability of Hate Speech Annotations: The Case of the European Refugee Crisis (**DBLP:journals/corr/RossRCCKW17**).

2.1.1 Definitions In Regulations & Legislations

The Appendix to the Council of Europe's Committee of Ministers' Recommendation no. R (97) 20, cited in (**arcan_2013**), defines hate speech in the following manner:

"The term 'hate speech' shall be understood as covering all forms of expression which spread, incite, promote or justify racial hatred, xenophobia, anti-Semitism or other forms of hatred based on intolerance, including: intolerance expressed by aggressive nationalism and ethnocentrism, discrimination and hostility against minorities, migrants and people of immigrant origin."

A few years later, the European Commission against Racism and Intolerance (ECRI) provides a more inclusive definition updated with other aspects that hate speech encompasses in the modern world. In its General Policy Recommendation No. 15: Combating Hate Speech, published in March 2016, the European Commission against Racism and Intolerance (ECRI), cited in Media Regulatory and Hate Speech (**council**), provides the subsequent definition of hate speech:

"[hate speech] entails the use of one or more particular forms of expression – namely, the advocacy, promotion or incitement of the denigration, hatred or vilification of a person or group of persons, as well any harassment, insult, negative stereotyping, stigmatization or threat of such person or persons and any justification of all these forms of expression - that is based on a non-exhaustive list of personal characteristics or status that includes "race", colour, language, religion or belief, nationality or national or ethnic origin, as well as descent, age, disability, sex, gender, gender identity and sexual orientation".

2.1.2 Definitions By Social Media Platforms

Among others, the largest social media companies have also published their own definitions of hate speech.

In its Community Standards, Facebook states the following:

"We define hate speech as a direct attack on people based on what we call protected characteristics – race, ethnicity, national origin, religious affiliation, sexual orientation, caste, sex, gender, gender identity, and serious disease or disability. We protect against attacks on the basis of age when age is paired with another protected characteristic, and also provide certain protections for immigration status. We define attack as violent or dehumanizing speech, statements of inferiority, or calls for exclusion or segregation." (**facebook**).

In its Rules and Policies, Twitter condemns hateful conduct in its platform by providing the following statement (**twitter**): *"Hateful conduct: You may not promote violence against or directly attack or threaten other people on the basis of race, ethnicity, national origin, caste, sexual orientation, gender, gender identity, religious affiliation, age, disability, or serious disease. We also do not allow accounts whose primary purpose is inciting harm towards others on the basis of these categories."*

The Instagram Community Guidelines mention hate speech among other unwanted content in their platform, without explicitly providing a definition for it. However they break down behaviours that fall under this concept in the following paragraphs: (**community**):

"We remove content that contains credible threats or hate speech, content that targets private individuals to degrade or shame them, personal information meant to blackmail or harass someone, and repeated unwanted messages. ... It's never OK to encourage violence or attack anyone based on their race, ethnicity, national origin, sex, gender, gender identity, sexual orientation, religious

affiliation, disabilities, or diseases. ... Serious threats of harm to public and personal safety aren't allowed. This includes specific threats of physical harm as well as threats of theft, vandalism, and other financial harm. ..."
It should be pointed out that Instagram will allow hate speech if it is being shared to challenge it or raise awareness, however that intent should be clearly stated in such cases. (ibid.)

2.1.3 Definitions In Scientific Papers

In their paper (**waseem_hovy_2016**), Waseem and Hovy classify a tweet as offensive if it:

1. *uses a sexist or racial slur.*
2. *attacks a minority.*
3. *seeks to silence a minority.*
4. *criticizes a minority (without a well-founded argument).*
5. *promotes, but does not directly use, hate speech or violent crime.*
6. *criticizes a minority and uses a straw man argument.*
7. *blatantly misrepresents the truth or seeks to distort views on a minority with unfounded claims.*
8. *shows support of problematic hashtags. E.g. "#BanIslam", "#whoriental", "#whitegenocide"*
9. *negatively stereotypes a minority.*
10. *defends xenophobia or sexism.*

11. *contains a screen name that is offensive, as per the previous criteria, the tweet is ambiguous (at best), and the tweet is on a topic that satisfies any of the above criteria.*

Fortuna et al (**fortuna_nunes_2018**), provide the following definition based on their analysis of other definitions:
"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 orientation, gender identity or other, and it can occur with different linguistic styles, even in subtle forms or when humour is used."
In our thesis we concur with Davidson et. al, definition of hate speech which is:
"language that is used to express hatred towards a targeted group or is intended to be derogatory, to humiliate, or to insult the members of the group." (**Davidson_2017**)
It is worthy of noting that their definition does not include all instances of offensive language, given that people often use terms that are highly offensive to certain groups in a qualitatively different manner (ibid).

2.2 Challenges

Varying definitions provide challenges for evaluation of hate speech detection systems. Existing datasets differ in their definition of hate speech, leading to datasets that are not only from different sources, but also capture different information. (**macavaney_yao_yang_russell_goharian_frieder_2019**)
One of the first challenges rises from the section described above. Many of the papers introduce a variety of definitions which makes the task

of hate speech detection difficult to begin with, even for human annotators. Such variety introduces a lot of discussion towards what actual is considered hate speech with people having different opinions on the say.

This difficulty is also reflected in the annotation process of data where the responsible people might mislabel the corpus given their own point of view in the matters. This issue is main focus of the paper Are You a Racist or Am I Seeing Things? Annotator Influence on Hate Speech Detection on Twitter (**waseem_2016**), which carries out experiments on data annotated by "amateur" and "expert" annotators to observe how that affects performance of hate speech detection tools. An interesting approach seen in (**ousidhoum_lin_zhang_song_yeung_2019**) includes the sentiment of the annotator when going through the data corpus. The claim that such the emotions of the annotator are key for the model given the subjective nature and social ground of the annotator's sentiment analysis. This definitely proves that the annotators play a substantial role toward hate speech detection.

Another challenge that limits the advancement of hate speech detection tools in multilingual systems is that the majority of the effort made in the field is related to English datasets. As it will become obvious from the following section, almost every paper discussing the task most certainly includes English as their primary language. Even though this is, to a certain degree realistic because the largest amount of data generated online is in English, other languages are lacking representation and thus falling behind in the fight against offensive language.

2.3 Models

After reviewing the definitions and challenges of multilingual hate speech detection, we move over to the classifiers commonly used for such tasks. Classifiers ranging from simple Machine Learning models such as Naive Bayes, Logistic Regression, Support Vector Machines (SVM) etc. to Deep Learning models such as Convolutional Neural Networks and Long short-term memory models are being used to handle the task of identifying hate speech. Moreover, a trend of using higher sophisticated architectures such as ELMo and BERT has immersed.
(**Davidson_2017**)start their work by trying out different Machine Learning models which include Logistic Regression, Naive Bayes and Linear Support Vector Machines (SVM). They train the models on English data collected from Twitter and divided into three categories: hate speech, offensive language and neither. Their results are pretty impressive reaching a F1 score of 0.90 with Logistic Regression as their classifier. Such classifiers are also used in the work of (**waseem_hovy_2016**). During their analysis on (**Davidson_2017**) they found out the model they used has a bias toward classifying the tweets as less offensive as the person who labelled them.
The work in (**macavaney_yao_yang_russell_goharian_frieder_2019**), makes use of the same simple Machine Learning classifiers as (**Davidson_2017**) and utilizes them as baselines for their proposed solution, a Multi-view SVM. The datasets used here are again in the English language. They also analyse patterns emerging from their classifiers and how they differ from the classifications made by BERT. They point out the struggle of interpretability when using BERT even though it is the model performing best. The main achievement of the paper deals with

a 7% improvement of the mSVM F1 score compared to previous similar work of combining classifiers.

The usage of English dataset is more prominent throughout papers and the last review we are including that deals with only English data refers to the work introduced in (**zampieri_malmasi_nakov_rosenthal_farra_kumar_2019**). In this paper, they make use of the OLID dataset, which is divided into three main categories of speech by the work of (ibid) , and following an hierarchical annotation. The annotation schema produced from this paper, is also followed in our experiments. The teams participating in this Evaluation task make use of various different models from Linear Regression to more state of the art classifiers such as BERT and ELMo. The results show that BERT was one of the highest ranked classifiers for each of the subtasks it used. However, (ibid) concluded that also Machine Learning models were competing quite well.

For the purposes of comparison, we include related papers that deal with Albanian datasets. The work in (**zenuni_ajdari_ismaili_raufi_2017**) establishes a dataset by extracting comments through Facebook pages that are popular in Albania. One of the pages, Jeta Osh Qef (JOQ) is an informal news platform dealing with real time news and advocating problems from people in need. The other source page, TV Klan, is one of the main news channels in Albania. Their best scoring classifier was the SVM classifier that produced a F1 score of 0.58. Other work done for the Albanian dataset are described in (**raufi_xhaferri_2018**). The dataset they construct comes from similar sources as the dataset we create in our work, specifically from JOQ and Xing me Ermalin. As a classifier, they train an Artificial

Neural Network and base their results on accuracy metric. The highest quality metric reported from (ibid) is an accuracy of 0.94.
Expanding and investigating the models encountered in hate speech detection, we now focus on the models used in a multilingual environment with the hope of better detecting hate speech online.
In (**sigurbergsson**), we see an attempt of using the same classifier to deal with both English and Danish language. The paper describes a variety of models used, starting with Logistic Regression. This classifier's architecture is based on the works of (**Davidson_2017**). Sigurbergsson et.al, also explores deep learning models referred to as Learned-BiLSTM, Fast-BiLSTM and Aux-BiLSTM. Using the OLID dataset, and their own created danish dataset, (ibid) their work includes all of the subtasks from the OffensEval schema and analyses the results based on these subtasks. For subtask A, dealing with offensive language detection, their best performing classifier for English is the Fast-BiLSTM with an F1 score of 0.735. For the Danish dataset, their best performing classifier is the Logistic Regression with an F1 score of 0.699. Here, the paper also makes an observation that the classifier performing best for English, performs the worst when the Danish dataset is given as input. In the second subtask that deals with the categorization of the data detected as offensive, the classifier that performs best for English is the Learned-BiLSTM, whereas for Danish, the Aux-Fast-BiLSTM achieves the highest F1 score. For the third subtask dealing with identification of such targeted offense, both datasets have the same best classifier, the Learned-BiLSTM. The scores between the subtasks follow a decreasing pattern which is justified by the amount of data annotated for the subtasks becomes lower as they transverse from subtask A to C.

Such work is also described in(**basile_bosco_fersini_nozza_patti_pardo_rosso_sanguinetti_**
where instead of Danish, they compare results with English and Spanish data extracted from Twitter. The task described in this paper, divides into subtasks that follow the same idea in mind with the OffensEval Schema with some slight alterations. They focus more of the hate speech towards the immigrants and women. The first subtask concerns the identification of hate speech and the second subtask deals with the identification of targeted groups. For the first subtask, the best performing team produced a macro F1- score of 0.651 where they use a SVM model with RBF kernel only on the provided data, exploiting sentence embeddings from Google's Universal Sentence Encoder (Cer et al.,2018 cited in (**basile_bosco_fersini_nozza_patti_pardo_rosso_sanguinetti_2019**) as features. Other teams that performed in the top five, used Neural Network models, such as Convolutional Neural Networks(CNNs) and Long short-term Memory models (LSTMs). As for the same subtask under the Spanish dataset, the best scoring team obtains an F1 score of 0.73 by using the SVM model. Other top-performing teams for this subtask have combined Logistic Regression, Multinomial Naïve Bayes, Classifiers Chain and Majority Voting. Higher sophisticated models such as BERT are seen to be used and performed well for this part. Finally in their error analysis they mention that the common errors are highly skewed towards the false positives. However, the unbalance is stronger for English (89.1% false positives) than for Spanish (76% false positives).

In the work of (**steimel_dakota_chen_kÃijbler_2019**), they also experiment with models ranging from Random Forest to Neural Networks. One new addition to the models used is the XGBoost. They investigate the multilingual environment with English and German dataset.

It is not clear from the paper if they are conducting any cross-lingual experiments, but their findings show that XGBoost performs better for English, whereas SVMs better for German.

Zero shot experiments were introduced in the work of (**punyajoy**), where their best performance was achieved when testing on the German dataset. The training dataset includes English and Hindi with around 10000 comments. The paper describes that BERT embeddings for the multilingual settings are used and they make use of models such as SVM and Gradient boosted trees.

More recent work on the multilingual setting can be found in (**chen_awadallah_hassan_wang_cardie_2019**). The model they describe is called MAN-MOE, a multilingual model transfer approach that exploits both language-invariant (shared) features and language-specific (private) features, which departs from most previous models that can only make use of shared features. This mixture comes from the work of (Chen and Cardie (2018a) cited in ibid) where they propose a generalized shared private model for the multi-source setting, where a multinomial adversarial network (MAN) is adopted to extract common features shared by all source languages as well as the target. The mixture-of-expert MOE was related to (Shazeer et al., 2017; Gu et al., 2018 cited in ibid) in order to learn the private features. They train the model over different NLP areas, but we focus on the results for text classification of Amazon reviews. The dataset that the paper uses consists of English, German, French and Japanese languages. These results recorded on accuracy show that the model introduced in the paper is highly effective in relation with other models they compare with.

In the work of (**ousidhoum_lin_zhang_song_yeung_2019**), they produce and annotate a dataset based on tweets coming from English,

French and Arabic in the hopes of detecting hate speech multilingually. They tackle multitask and single task classifications. They also elaborate on the annotation process which is very thorough and consists of directness, hostility type, target attribute, target group and the sentiment of the annotator. The classifier used in (ibid) is a BiLSTM with one hidden layer. During the experiments based on directness, (ibid), finds that the BiLSTM outperforms Logistic Regression scoring a F1 score 0.86 average on the languages. For the multitask classification, the results drop considerably due to the complexity of having a large amount of labels.

2.4 OffensEval Schema

All of the datasets described in the preceding paragraph follow the same annotation schema, also known as the OffensEval schema. This distribution of labels and tasks for the dataset has been mentioned in various papers such as in the works of (**zampieri_malmasi_nakov_rosenthal_farra_kumar_2019_1**) and (**zampieri_malmasi_nakov_rosenthal_farra_kumar_2019**) and (**sigurbergsson**). This hierarchical annotation schema consists of the following subtasks:

1. **Offensive Language Detection**
   The first subtask of this schema deals with the categorization of the language used throughout the dataset as offensive (OFF), or not offensive (NOT).

   - Not Offensive (NOT): This label of the schema refers to comments that do not contain any usage of profane or offensive language.

- Offensive (OFF): This label refers to comments that contain any form of inappropriate language such as the use of profanity, threats, insults, discrimination etc.

2. **Categorization of Offensive Language**

   The second subtask considers the subset of the comments in the dataset that were annotated as Offensive (OFF) during the first subtask of the schema. This categorization consists of two classes.

   - Targeted Insults (TIN): Comments that identify insults towards an individual, a group, or other categories.
   - Untargeted Insults (UNT): Comments that include profane, abusive and/or offensive language, but do not specify a target are categorized in this label.

3. **Offensive Language Target Identification**

   The third subtask is carried out on the comments of the dataset that were annotated as Targeted Insults (TIN) during the second subtask. Each comment is assigned one of the following three labels.

   - Individual (IND): Any insult targeted towards an individual. The individual targeted can be a celebrity, a named or an unnamed participant of the comment.
   - Group (GRP): Any targeted insult towards a certain group regarding their ethnicity, gender or sexual orientation, political affiliation, religious belief, or other common characteristic are given this label.
   - Other (OTH): Any insult that targets an event, organization or issue falls under this label.

Chapter 3

Task Framing

3.1 Experiment Description

As mentioned previously, in this thesis we aim to explore a wide variety of settings related to multilingual hate speech detection. The experiments conducted in this work will provide answers to the questions stated in the Introduction chapter and provide insights into how models act in multilingual settings in comparison to monolingual settings. We also investigate whether models created for languages that are less widespread and therefore less included in Natural Language Processing studies can easily learn and benefit from additional datasets available for widespread languages that are commonly used in the field. Lastly, we inquire whether it is possible to use the state of the art models trained on these widespread languages alone as out of the box solutions for the lesser studied languages that often lack research as well as large sets of annotated data.

In our first set of experiments we would like to get an insight on how well our chosen models perform under the monolingual setting. Here,

we undergo both the training and testing phase with the same language in order to form a baseline for our questions and see how the addition of other languages affect the performance of the models.
In the second set of experiments we want to explore whether the combination of two languages can improve the performance of a model. Additionally, we want to see if combinations of languages that have some common characteristic yields even better results than a random pairing. For this reason we have chosen the datasets mentioned in chapter 4. The common characteristics regarding our chosen datasets are that Danish and English originate from the same family of languages, whereas Albanian and Turkish share many words and expressions due to the influence of the Ottoman Empire during a 500 year occupation of Albania. We perform the experiments on all possible combinations for the bilingual setting, and observe whether our hypothesis holds.
Our third set of experiments involves training and testing models on very large datasets comprising a variety of languages. In our multilingual experiments we investigate whether providing the model with abundant knowledge on language-specific offensive speech characteristics leads to better results, or whether the system is saturated with information that clouds its judgement.
In our fourth set of experiments we investigate whether it is possible to obtain a better performance for models catering to specific languages by including data from additional languages in the training phase. This is especially beneficial and promising for languages in which large corpora of annotated data are not available, and allows for efficient hate speech detection systems to be created in a short period of time as the need for data extraction, cleaning and annotation is eliminated. It is however

advantageous in other settings as it facilitates swift knowledge transfer from additional sources, hence the name of the experiment.
Our fifth and final set of experiments includes testing a modelâĂŹs performance on languages it is unfamiliar with or, in other words, languages that it has not encountered during its training phase. Similarly to the previous experiment, a positive outcome in this situation would prove to be extremely favourable for languages whose annotated data corpora are of sizes that do not allow for efficient training and testing splits, as the model cannot learn sufficient information. A more profitable outcome of this experiment is the opportunity of creating well-trained hate speech detection models on one or more well-studied languages, and then applying these models to a variety of different languages without the need for language dependent tweaks on the model. An "out of the box" offensive and hate speech detection system would be very useful to social media platforms as it allows immediate language independent detection without the need for content translation or similar steps.

3.2 Choice of languages

One of our aspirations for this piece of work was to provide a contribution in the field of offensive and hate speech detection for languages that have not been commonly included in related studies. As a result, three of the four languages that we have chosen to experiment on - namely Albanian, Danish and Turkish - are, to our opinion, languages that are not quite widespread. This also means that the regulating policies for hate speech detection in social media for the languages in hand might not be as advanced as they are for more common languages such as En-

glish. We believe that having good detection systems for such languages will serve as a stepping stone towards more positive communication in social media platforms.

Furthermore, these languages were chosen because the data corpora available for them conformed to the guidelines specified in the OffensEval schema up to a certain degree. Given that the Turkish and Danish datasets were only annotated for the first subtask defined in the OffensEval schema, all of the experiments described in this thesis are conducted solely on this subtask in order to assure inclusivity and a common ground for comparing the results obtained in each setting.

The next chapter describes the technical aspect of the hypotheses, starting with the introduction of the models and their specifics along with other preparation needed for their training.

Chapter 4

Methods

4.1 Datasets

In order to create solutions that cater to the multilingual nature of the problem in hand, we have chosen to conduct our experiments on datasets of 4 different languages, namely Albanian, English, Danish, and Turkish. In the following subsections we provide information about each of the aforementioned datasets, including their creation, data sources and data distribution.

4.1.1 Albanian

The Albanian dataset is created by the authors of this thesis and is an extension of the dataset described in "Hate Speech and Offensive Speech Detection for Multiple Languages"(**nurce_keci**). The data is extracted from several Albanian accounts on Instagram and Youtube, and the final version of the dataset which we make use of in the following experiments includes a total of 11874 comments. For the data extraction from Instagram, we have used the tool Instaloader(**instaloader**),

whereas for the comments retrieved from YouTube we have made use of the Google Developer's API(**google**). The comments have been labelled by four annotators following the OffensEval Schema (**zampieri_malmasi_nakov_rosenthal_farra_kumar_2019_1**), with the authors of this thesis revising annotations made by the two other annotators. Table 4.1 includes the distribution of different categories of comments present in this dataset, for each of the accounts they were extracted from. Figure 4.1 provides the percentages of each of the labels used to annotate where we see that 87% is considered as Not Offensive and the rest is distributed in between the categorization of the offensive and their target. A more detailed description of the dataset, its creation and content is given in (**nurce_keci**)

Label/Source	JOQ	LagjiaJone	Youtube	Total
NOT	9204	902	201	10307
OFF, UNT	384	55	9	448
OFF, TIN, IND	644	30	64	738
OFF, TIN, GRP	219	7	7	233
OFF, TIN, OTH	134	7	7	148
TOTAL	10585	1001	288	11874

Table 4.1: Statistics for each datasource of the Albanian dataset

4.1.2 English

For the English language, we make use of the OLID dataset that is described in the work of (**zampieri_malmasi_nakov_rosenthal_farra_kumar_2019_1**). The dataset is formed by using APIs to extract the data from Twitter and annotating them using the hierarchical OffensEval Schema which is described in chapter 2 and consisting of 13241 comments for training and 860 for testing. Since the other dataset chosen did not include the

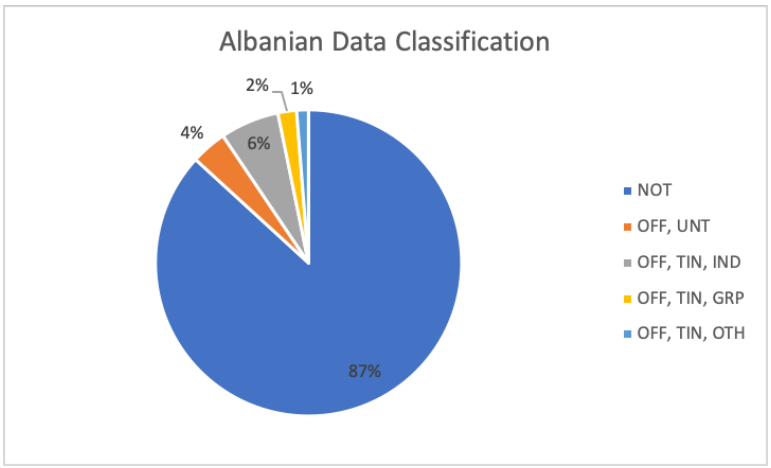


Figure 4.1: Data distribution for each label in the Albanian dataset

predefined testing data, we decided to include only the training data for English as well. The details of the dataset regarding its distribution between Offensive and Not Offensive data are reflected in Table 4.2. Figure 4.2 gives us a view on the percentage distribution between these data labels distributed by category and target.

Label/Source	**Twitter**
NOT	8841
OFF, UNT	524
OFF, TIN, IND	2407
OFF, TIN, GRP	1074
OFF, TIN, OTH	395
TOTAL	13241

Table 4.2: Statistics for the English dataset

4.1.3 Danish

The Danish language dataset included in our experiments is created in the work of (**sigurbergsson**). This dataset was made available through hatespeechdata.com. The dataset described in (ibid.), is constructed by

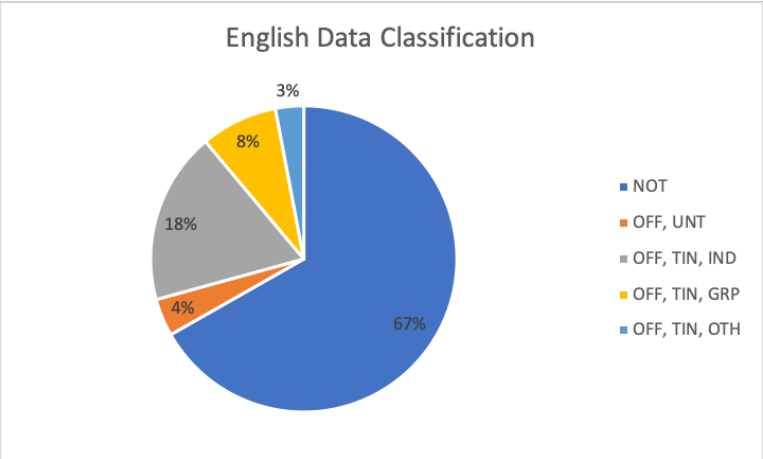


Figure 4.2: Data distribution for each label in the English dataset

extracting information from various platforms such as Twitter, Reddit and Newspapers and it follows the OffensEval schema hierarchical annotation. The details of the distribution of labels is shown in table 4.3 and figure 4.3 where we see a 13% distribution of the data is annotated as Offensive (OFF) and the rest as Not Offensive (NOT).

Label/Source	**Twitter**
NOT	2577
OFF	384
TOTAL	2961

Table 4.3: Statistics for the Danish dataset

4.1.4 Turkish

For the Turkish language we include a dataset described in (**ÃĞÃűltekin**) and is again available from hatespeechdata.com website. The corpus constructed in the paper (ibid) includes 36232 sentences extracted from micro-blog posts on Twitter. In this work, the dataset is annotated again using an hierarchical structure starting with dividing

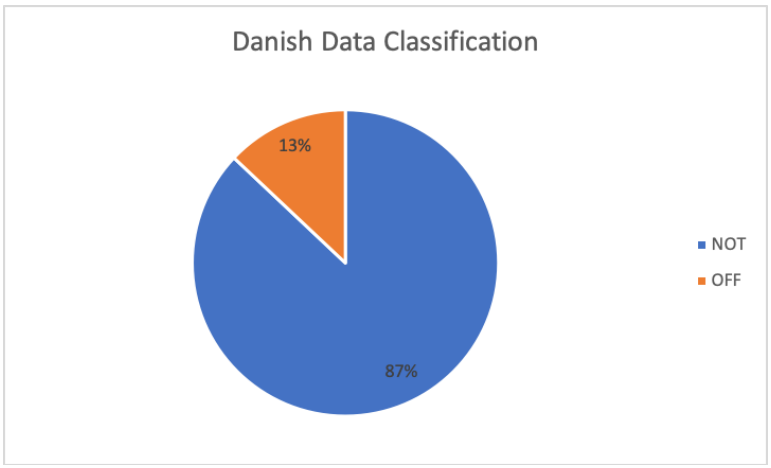


Figure 4.3: Data distribution for each label in the Danish dataset

the data into what is considered as Not Offensive (NOT) and Offensive(OFF) which is then divided into Targeted and Not Targeted data and finally divide the Targeted into Individual, Group and Other labels. Out of this corpus of data, we could only access the training set labelled solely for the first subtask in OffensEval. The details concerning the division between Offensive (OFF) and Not Offensive (NOT) are displayed in table 4.4. Figure 4.4, then displays this distribution between the labels percent-wise.

Label/Source	**Twitter**
NOT	25625
OFF	6131
TOTAL	31756

Table 4.4: Statistics for Turkish dataset

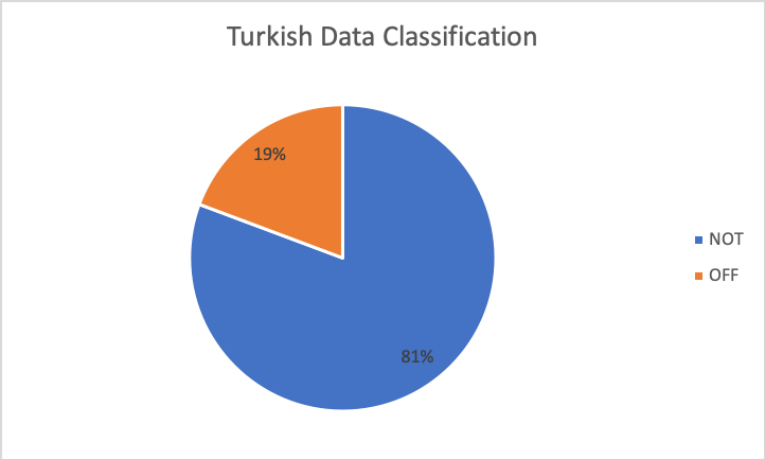


Figure 4.4: Data distribution for each label in the Turkish dataset

4.2 Models

4.2.1 Naïve Bayes

The Naïve Bayes classifier is a simple supervised Machine Learning algorithm. It applies the formula from Bayes' Theorem to perform Machine Learning tasks. Differently from many other classifiers, it uses probability to determine the class on a certain sentence. Figure 4.5, displays the formula in which Naïve Bayes is based on. It calculates the posterior probability, meaning given a feature x what will class c be. Such information can be conducted only after knowing the likelihood, which is the probability of a feature given the class it falls under, the class prior probability, meaning the probability of a certain class and lastly the predictor prior probability meaning the probability of the feature x. The Naïve part of the classifiers name comes from the assumption it makes of the independence of features towards a class. Naïve Bayes is used as a baseline classifier for our work as it has shown to be very effective in real time scenarios when dealing with many appli-

cations We use the sklearn library's implementation when constructing the experiments with Naïve Bayes classifier. The library has many versions of the Naïve Bayes, but we make use of only the Gaussian Naïve Bayes which assumes the likelihood of the features to be Gaussian.

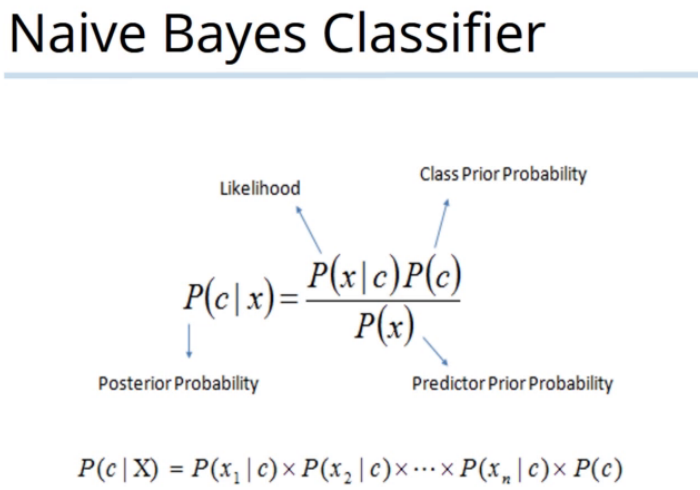


Figure 4.5: Bayes' Rule.

4.2.2 BiLSTM

The synthesis of the Bidirectional LSTM begins with the Recurrent Neural Networks. The structure of RNN starts by reading the input tokens or vectors and using them to compute an output vector that represents the whole inputed sentence. The main issue as described in (**sigurbergsson**), concerns the vanishing gradient problem. Therefore, models such as LSTM are more commonly used when performing tasks regarding text classifications. A more extended model would be the BiLSTM that we are using as one of the classifiers in our thesis. BiLSTM as the name might suggest includes two layers of LSTM, one going from

the first input vector to the last and the other LSTM vice-versa. Each of the LSTM layers produce a vector from the input layer, and with BiLSTM we concatenate these vectors into one. The end product of the classifier would be a prediction of one of the labels available, in our case either 0 referring to Not Offensive (NOT) content or 1 referring to Offensive (OFF) content. The last layer makes use of the softmax function to determine which label would come out of the classifier. The implementation of such architecture is done by using the Keras API where each layer described is constructed in a sequential order. For the word embedding layer of BiLSTM, we have chosen to use the word to vector representations that are publicly available from FastText . Figure 4.6 displays a common architecture of BiLSTM.

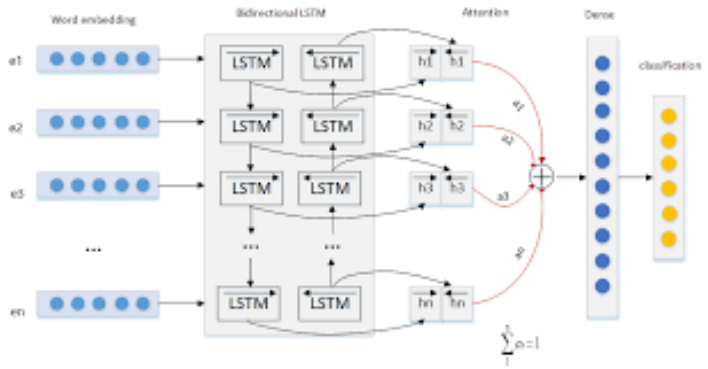


Figure 4.6: The architecture of BiLSTM.

4.2.3 BERT

BERT stands for Bidirectional Encoder Representations Transformers. Its first known architecture is described in the work of BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding . The idea behind BERT sparks the lack of techniques that improve

fune-tuning approaches of language model.The techniques so far have only used a unidirectional standard language model. BERT introduced a masked language layer to remove this barrier from previous techniques. Figure 4.7, taken from the work in (ibid), describes the procedures of BERT. The base model, which we also make use of in our work, has a total of 12 layers, 768 hidden size and 12 self-attention heads with a total of 110 million parameters. In our thesis, we used the implementation of such a model from the huggingface library. This library includes a variety of implementations that include also more advanced versions of BERT such as AlBERT and roBERTa. We, however, use only the base version of BERT in a monolingual environment and also in the multilingual environment.

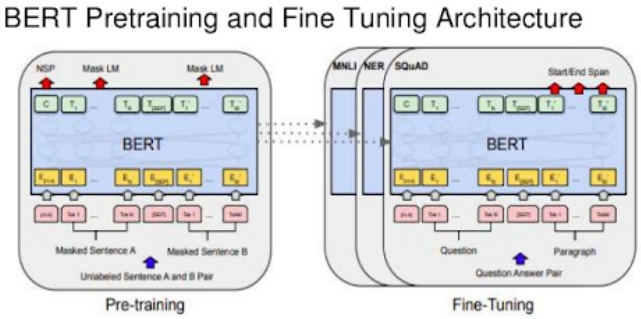


Figure 4.7: The architecture of BERT.

4.3 Data Cleaning

The first step towards having a reliable data fed to the model has to do with data cleaning. In this step, we firstly replace any links such as "https://..." with a tag "URL". We do a similar replacement every time

a valid user is being referenced and that means for every word starting with a @ sign, we replace the actual username with the tag "@USER" to keep the data clean and mask the user from becoming public to the reader. We chose to keep the emoticons in our dataset as we noticed that a lot of the comments used them and by their usage the context of the comment would change, or the emoticons expressed a lot meaning to the comment. The inclusion of the emoticons is also seen in the work of "A Lexicon-based Approach for Hate Speech Detection". Lastly all of the words were lower-cased. This procedure was conducted on all the different datasets.

4.4 Tokenization

4.4.1 Naïve Bayes

For the tokenization process in the Naive Bayes experiment, we used "Term Frequency times Inverse Document Frequency", referred herein as *tf-idf*. The tokenization process is deconstructed two steps, the Term Frequency step and Inverse Document Frequency. The first step is achieved by dividing the occurrence count of each word in our data corpus with the total number of tokens in it. The second step deals with extracting how much information a certain word provides from the data corpus. To do this, we compute the logarithm of the fraction between the number of samples in our data corpus and the number of samples from the corpus that actually contain the word (REF= https://en.wikipedia.org/wiki/Tf%E2%80%93idf). There is an implementation of the tf-idf tokenizer within the sklearn library

(REF=https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.Tf
which we use in the construction of the classifier.

4.4.2 BiLSTM

For this model, we firstly create a map to store all the words, symbols or emoticons uniquely and attach a counter incrementing by one for each new word. Thus, we have created a dictionary of word-to-index that will help us with converting the data into integers for the model to train. The size of this dictionary dictates also the vocabulary size. Furthermore, to complete the process of tokenization, we make use of a method from the Keras library called *pad_sequences* which helps us construct sentences of the same length. We decided that this length would be the length of the longest sentence in our dataset *max_length* so we do not lose any information when feeding the data to the model. Sentences with length smaller than the *max_length* variable would be padded in the end by the special integer value that does not coincide with any integer from the word-to-index. Therefore, our word-to-index mapping starts from the integer 1 and the padding integer is 0. The following structure describes a simple transformation from the actual sentence to what is fed to the model. Figure 4.8 shows three simple sentences that are used for this demonstration. Figure 4.9 shows the transformation of the words to their index representations. And lastly figure 4.10 show the final form of the sentences.

4.4.3 BERT

For this model, we make use of a library known as huggingface (REF= huggingface), so the tokenization is done in a way which fits BERT better

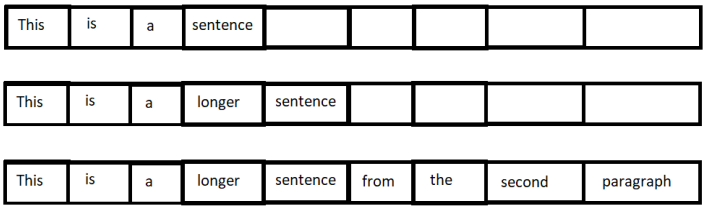


Figure 4.8: Three sample sentences.

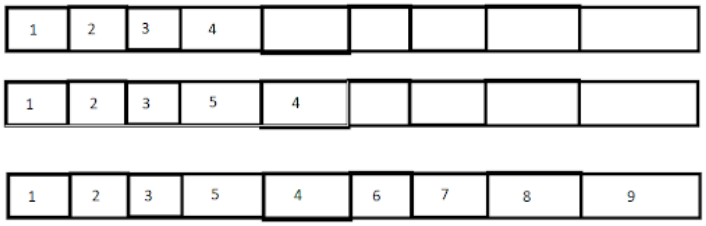


Figure 4.9: Word to index transformation.

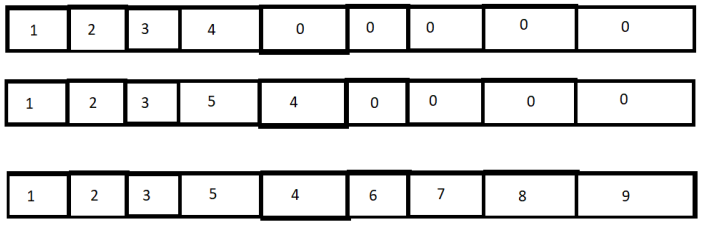


Figure 4.10: The final padded sentences.

than our own implementation. The huggingface (REF = transformers library) library that we have previously mentioned as well, seems to have a great implementation through all the steps required for classification tasks for each of their BERT implementations. We make use of two tokenizers presented in the AutoTokenizer class implementation of the transformers library, which are the *"bert-base-multilingual-cased"* and its simpler version. For each of the sentences we create use the *encode_plus* method to perform these steps:

1. Tokenize the sentence

2. Add the [CLS] and [SEP] tag for each sentence in the beginning for the first and end of the sentence for the latter.

3. It maps the tokens to the ids

4. It padds or truncates according to the specified max length

5. And creates the attention mask for [PAD] tokens.

These steps are standard procedure when tokenizing the data and feeding them to the BERT architecture. Lastly all the lists are turned into tensors using pytorch (REF = pytorch) library. Figure 4.11 shows the transformation from the original content of the dataset into their tokenized form to be fed in the model.

```
Original:  @user and twitter seems to be beholden to cnn
Token IDs: tensor([  101,   137, 29115, 10111,   188, 56082, 10877, 34208, 10114, 10347,
        10347, 78926, 10115, 10114, 88234, 10115,   102,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0])
```

Figure 4.11: Tokenization in BERT.

4.5 Word Embeddings

During the recent years, a trend of using pre-defined and pre-trained word embeddings has emerged. These word embeddings are usually trained using Neural Network models with a very large unannotated data corpus. By doing this, the embeddings turn out to be similar for words with similar meanings in all languages. This representation is done by creating a vector for each word, hence

the idea of word to vectors. There are various types of embeddings used for models that utilize embedding layers. Some examples include Glove, Word2vec or even pre-trained FastText embeddings(**joulin_grave_bojanowski_mikolov_2017**). The main difference between types of representations is that Word2vec(**word2vec**) and Glove(**glove**) are trained usingword n-grams whereas FastText embeddings are trained using character n-gram. The latter is described as more efficient for the task we would like to achieve(**sigurbergsson**). Another difference between the latter is the data corpora they are trained in. By using word embeddings as a feature to our classifiers, we are using their advantage of expressing information about the word in their context of usage. The advantage of this word vector representations are observed in the works of (**basile_bosco_fersini_nozza_patti_pardo_rosso_sanguinetti_2019**). These papers include pre-trained vectors into their classifiers that help them achieve better results than randomly initialized embeddings.

In our thesis we will make use of the pre-defined vector representation available from FastText (**fasttext**). The vector dimension available from FastText is 300, with the option of limiting them to a lower dimensionality space. The vector files are provided as .txt files and consist of around 1 million word to vector representations for the English and Albanian languages whereas for the Danish and Turkish language, they include around 2 million entries.

In order not to load the vector files every time we want to build an experiment, we construct a file to store each word, the index of that word and its vector. This file consists of only words that are actually present in our dataset and thus, relieves some of the complexity of the experiments. During the construction of such files, we also have to consider

the words that are present in the dataset, but not in the vector files. This might be seen as a disadvantage of the vector files where they only have a selection of words represented. In order to have a vector representation of these left-out words from the pre-trained FastText, we randomly initialize vectors following a normal distribution with mean 0 and deviation 1 and of course follow the dimension space of the pre-trained vectors. This step is done once depending on the setting of our experiments. Once the files are finalized, they will be used in the embedding layer of the models.

Chapter 5

Experimental Setup

5.1 Types of experiments

The work presented in this thesis is structured into five experiment categories, each of which aims to explore a different aspect of offensive and hate speech detection with the purpose of working towards a robust multilingual hate speech detection mechanism. The five experiment categories are namely:

1. Monolingual
2. Bilingual
3. Multilingual
4. Knowledge transfer
5. Zero shot

The motivation behind each experiment is previously explained in the Task Framing section (LINK to section). In the following subsections

we elaborate on the practical setup of each of the categories and any additional details on the experiments conducted.

5.1.1 Monolingual

The conducted experiments that fall under this category deal with training and testing the chosen models on the same language. The annotated data of the respective language is split into a training set and a testing set, following a 80:20 or 90:10 ratio. Table 5.1 displays the information about the training and test split samples for every language chosen in this thesis.

Language	**Training set**	**Testing set**
Albanian	10686	1188
Danish	2368	592
English	10592	2648
Turkish	25149	6288

Table 5.1: Monolingual train and test split

5.1.2 Bilingual

The experiments conducted that fall under this category deal with training and testing the models on a combination of two languages, i.e we train a model with a partition of the Albanian and English data and test on the rest of the Albanian and English data again. Table 5.2 displays information about the combination of languages that we have experimented on.

Language	Training set	Testing set
Albanian, Danish	11867	2967
Albanian, English	20091	5023
Albanian, Turkish	34648	8663
Danish, English	12960	3240
Danish, Turkish	27517	6880
English, Turkish	35741	8936

Table 5.2: Bilingual train and test split

5.1.3 Multilingual

Under this category we include experiments that train and test the models with the combination of all the four languages chosen for the paper. Initially, the four datasets are combined and shuffled to mix the different languages with each other. Afterwards, data is split into the training and testing partitions following a 80:20 ratio. Table 5.3 displays information about experiments that fall under the multilingual setting.

Language	Training set	Testing set
ALB, DAN, ENG, TUR	47608	11903

Table 5.3: Multilingual train and test split

5.1.4 Knowledge transfer

Starting from this category of experiments, the languages used during the training and testing phase are no longer the same on both partitions. As the goal of the experiment is to observe whether data from a different language improves the performance for a specific language, the method we follow in this experiment is to add new languages to the training set but not to the testing set. Supposing we want to perform knowledge transfer from English into Albanian, the experiment would follow this procedure:

1. Divide the Albanian dataset into the training and testing set as you would in the monolingual setting.

2. Add the English dataset to the training split.

3. Train and test the model.

Table 5.4 displays information of such experiments.

Language	Training set	Testing set
ALB, ENG –> ALB	22739	2375
ALB, TUR –> ALB	40936	2375
DAN, ENG –> DAN	15608	592
ENG, ALB –> ENG	22466	2648
ENG, DAN –> ENG	13552	2648
TUR, ALB –> TUR	37023	6288

Table 5.4: Knowledge transfer train and test

5.1.5 Zero shot

This experiment involves testing the model on a language that is not part of the training phase. We further divide this experiment into three sub experiments.

1. The first one deals with training the models on the full dataset of a language and testing with the complete dataset of another language, i.e. train on the full Albanian dataset and test on the full Danish dataset.

2. In the second sub experiment, we train the model with the full dataset of two languages and test on the full dataset of a third language, i.e we train on full Albanian and English datasets and test the model on the full Danish dataset.

3. The third part of the experiment revolves around the same idea, however we expand the training dataset with a third language again different from the test language dataset. In this case we would train on Albanian, English and Turkish dataset and test on the Danish dataset.

5.2 Evaluation metrics

In order to assess the efficiency of the models in our chosen experiments, we perform data classification on the test partition of the dataset and record the performance by keeping track of a few quality metrics. The metrics that we have recorded are recall, precision, F1 and macro averaged F1-score. The first three quality metrics have been used in a wide variety of papers such as (**zampieri_malmasi_nakov_rosenthal_farra_kumar_2019**) and would help us compare some of the results depending on the tasks. When dealing with classification tasks, it is very interesting to see how the model classifies and missclassies labels for specific sentences in our dataset. When prediction is done, the classifier displays a mixture of potential outcomes that fall under one of the four classes:

- True Positives (T_p): The data is originally labelled as positive and its predicted label is also positive.
- True Negatives (T_n): The data is originally labelled as negative and classified as such.
- False Positives (F_p): The data is originally labelled as negative, but is classified as a positive one.

- False Negatives (F_n): The data is originally labelled as positive, but is classified as negative instead.

Based on the number of predictions that fall under each of the aforementioned classes, recall and precision are recorded as follows. Recall will be defined as the ratio of true positives over the sum of true positives and false negatives(equation 5.1).

$$Recall = r = \frac{T_p}{T_p + F_n} \tag{5.1}$$

Precision will be defined as the ratio of true positives over the sum of true positives and true negatives (equation 5.2).

$$Precision = p = \frac{T_p}{T_p + F_p} \tag{5.2}$$

However, basing the model score only on these metrics would not be enough as the model might suffer from over-learning and classifying all test samples as one class, leaving us with a high recall and low precision. Therefore, we use F1 score to have one balanced informed metric. F1 will be defined in equation 5.3

$$F1 = 2 * \frac{p * r}{p + r} \tag{5.3}$$

As we are dealing with multiple labels here, we need a slightly different and more informed quality metric, which is the macro averaged F1 score. The macro F1 score as described in (**sigurbergsson**), will evaluate the scores of each class independently and calculate an unweighted average of these. Its benefits over metrics such as accuracy again as described in (ibid) would be that when having the model classifying a lot more samples of one class would produce a high accuracy, but a low

macro average F1 score. Thus, we use the macro averaged F1 score to see if our models have been over-fitting.
Lastly, we introduce other quality metrics that we will base our comparisons of model scores. However this metric score will depend on the actual application of the hate speech detection task. In various applications of hate speech detection, we might want to favour a high precision over a high recall or vice versa. Therefore we introduce F-beta as a quality metric as well so that we can draw conclusions on different applications that the models can be used in hate speech detection. The general formula for F-beta is is given in equation 5.4.

$$F\beta = (1 + \beta^2) * \frac{p * r}{r + \beta^2 * p} \tag{5.4}$$

The inclusion of the last metrics in addition to the commonly used F1-score has to do with the applications that the system described in our thesis can be used on. These applications might be of different purposes which require a high precision or high recall which can be calculated using different values of beta. A more detailed discussion is followed in the Results & Analysis chapter of the paper.

5.3 Imbalanced-learn Library

During the gathering of datasets, we noticed that their sizes varied significantly from one language to another. The Turkish dataset, for example, had almost 36000 entries while the Danish dataset only 3000. The disbalance is present not only on the overall size of the datasets, but also in the ratio of data labelled as Offensive (OFF) and Not Offensive (NOT) within each dataset as well as between datasets of different languages.

For this reason, we decided to perform various experiments in which we teaked the balance between data of each label in order to observe if such a change would positively affect the performance of the models. Given that offensive data is under-represented in the dataset, we performed the undersampling method on the non-offensive data partition of the dataset. This process essentially removed some of the data labelled as Not Offensive, which allowed for comparable numbers of the OFF and NOT data provided to the model. The balancing process was carried out using the imbalanced-learn library (**imbalanced**). Further details regarding these experiments are given in the respective subsection of the Results & Analysis chapter.

5.4 Experiment parameters and Hardware components

During the experiments, we have alternated the parameters of each classifier to get some insights on the performance of the classifier. These parameters include number of epochs, batch sizes, learning rate, optimizers and different loss and activation functions. To find the most performant form of each, we have experimented by keeping one of the parameters as a variable and others as constants. We have done this for each parameter to get a feeling where the classifier would achieve high scores. After this session we have decided to run the classifiers with these parameters.

5.4.1 BiLSTM

For the BiLSTM classifier we have chosen to run our experiments with the parameters shown in table 5.5.

Epochs	Batch size	Learning Rate	Optimizer	Loss function	Activation function
10	128	0.001	adam	sparse categorical crossentropy	relu, softmax

Table 5.5: BiLSTM parameters

The choice of epochs also was made to reduce the training time, given that the experiments we want to conduct are heavy computational and time wise. Testing of batch size included using batch size of 128 and 266, but our experiments ran better on 128. For the learning rate, tests were done with a range from the chosen value until the 0.008. In little cases, we observed a slight increase of performance with a learning rate of 0.003, however the chosen one was the most stable. Adam optimizer was our go to function from the beginning because we were more familiar with it. Other tests included the usage of Rectified-Adam(**pypi**) as an option for the optimizer, however during testing, it performed slightly worse than Adam. The most interesting parameter we came to contact was the loss function. Usually, when dealing with only two labels, Keras' example implementation suggested the *binary_crossentropy* loss function. However, we observed that by using this loss function in our experiments, the classifier would be overconfident with the detection of labels and marked all the test samples as Not Offensive.

5.4.2 BERT

During the testing phase for our parameters with BERT, we have relied heavily on the suggestions that the huggingface library implementation suggests. The range for the number of epochs, batch_size and learning

rate were already described in their interface. They concluded that the number of epochs should be between 2 and 4, the batch size 16 or 32 and the learning rate 2e-5, 2e-5 or 5e-5. The only optimizer we test on is the AdamW implemented again in the library. A few variations of epochs, batch size and learning rate were tested and we decided to carry out 4 epochs, on batch sizes of 32 and a learning rate of 2e-5.

All the experiments are run under Google's Colaboratory research notebooks (**colaboratory**) and by making use of two GPUs, TESLA K80 and Tesla P100-PCIE-16GB , depending on random allocation given. The operation of these two has only shown difference in time consumption of each experiment and no effect on the actual results we are interested in.

Chapter 6

Results & Analysis

In this chapter, we present the results of our experiments divided into the categories mentioned in chapter Experimental Setup. We elaborate on the best performing classifier under each category and analyse their strengths and weaknesses while dealing with multilingual hate speech detection. Finally we spark some discussion based on the findings of each model and how they performed during the testing phase, on which language the models tend to make more errors in their predictions as well as some common characteristics of mislabelled data.

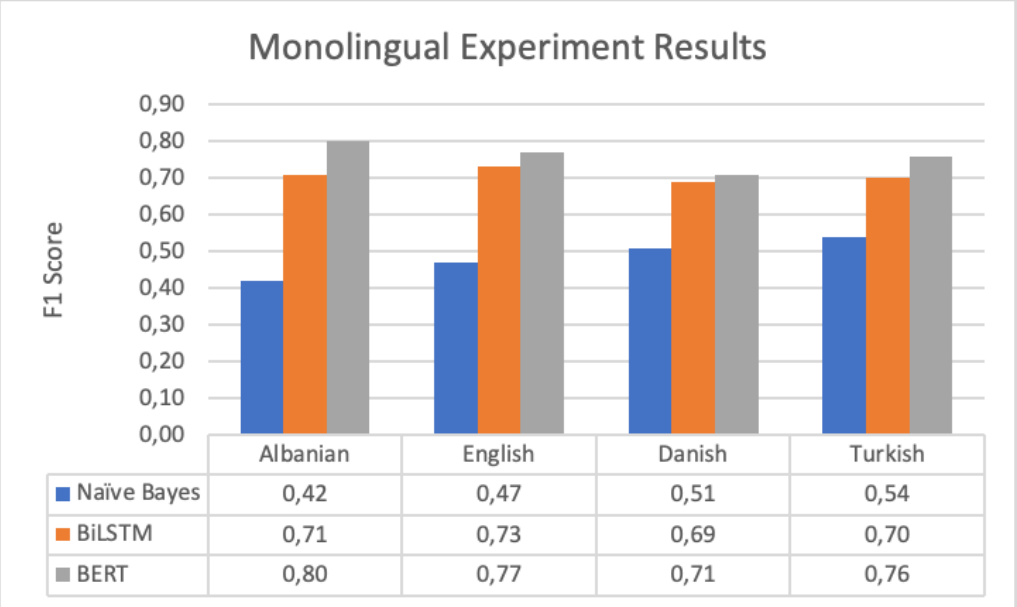
6.1 Results

6.1.1 Monolingual

In Figure 6.1we present the best results of each model for each language under the Monolingual setting of experiments. As expected the baseline classifier is the lowest performing one as we have not adjusted any of the parameters when conducting the experiments. As we had anticipated, BERT turns out to be the classifier with the highest F1 score

independently of the language it was trained in. We also observe that even though the sizes of the datasets vary, BERT still performs almost with the same F1 score leading us to believe that BERT might be a good choice when adding more languages to the experiments. Comparing the datasets constructed, it seems that the Albanian languages follows a good distribution between the OFF and NOT labels on both partitions and therefore resulting as the highest F1 score between languages. We also see that our own constructed dataset performs better with the classifiers than the english dataset. Such performance is also observed in (**basile_bosco_fersini_nozza_patti_pardo_rosso_sanguinetti_2019**), where the classifiers there performed better when testing with Spanish than with an english dataset. The results achieved for the Albanian language - an F1 score of 0.80 - are not only the best within our set of monolingual experiments, but they also outperform other work done for offensive or hate speech detection within the Albanian language, such as Automatic Hate Speech Detection In Online Contents Using Latent Semantic Analysis (**zenuni_ajdari_ismaili_raufi_2017**) which achieve an F1 score of 0.58. Moreover, the F1 score of 0.71 achieved in this experiment for the Danish language outperforms the results given in Multilingual Hate Speech Detection, Detecting the Types and Targets of Offensive Language in English and Danish Social Media Data (0,699) (**sigurbergsson**) , which is to our knowledge the best result achieved for Danish. Lastly, we see a resemblance with the work in (**zampieri_malmasi_nakov_rosenthal_farra_kumar_2019_1**) where the OLID dataset is synthesized. Our implementation does not outperform their best scoring classifier which is CNN, however we observe a high resemblance between the F1 scores achieved using the BiLSTM classifier in this work and (ibid). The same reasoning is followed with the results

for the Turkish language where a similarity between our scores and the scores of (**ÃĞÃűltekin**) is observed.



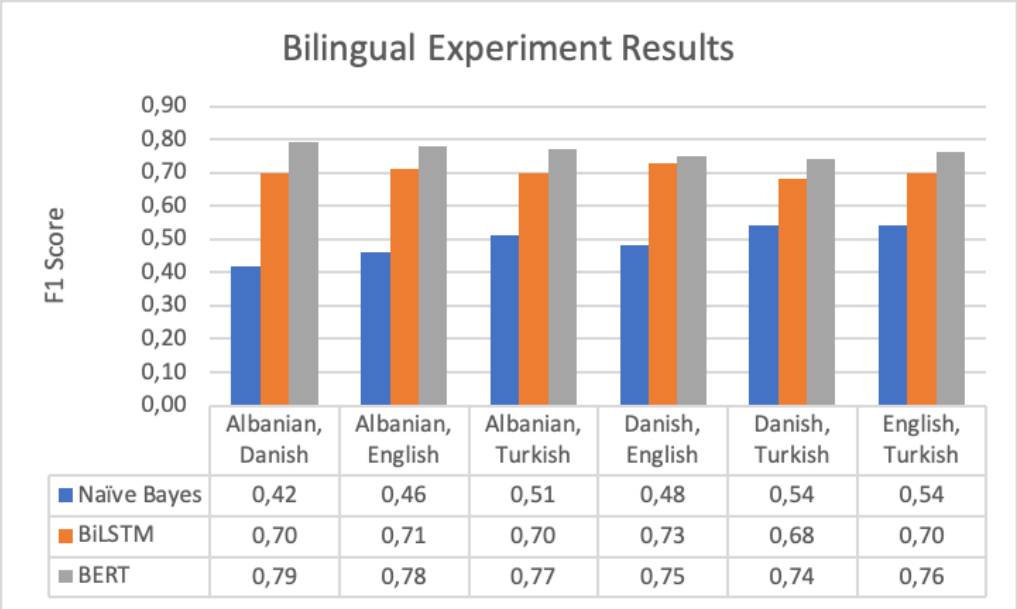
	Albanian	English	Danish	Turkish
Naïve Bayes	0,42	0,47	0,51	0,54
BiLSTM	0,71	0,73	0,69	0,70
BERT	0,80	0,77	0,71	0,76

Figure 6.1: Results from the Monolingual Experiment.

6.1.2 Bilingual

In fingure 6.2, we witness a continuation of the pattern of BERT scoring the highest F1 values which arises in the Monolingual setting. We see that a combination of Albanian and Danish scores the best results. This leads us to believe that classification for languages with small annotated datasets such as the Danish one in this work, could be improved by providing the model with additional knowledge from datasets of other languages. However, taking into account that the testing data is a random mix of the two languages and that we do not know whether the percentage of the mislabelled Danish data has increased or decreased in comparison to the monolingual Danish experiment, we cannot con-

sider this as a deterministic result. On the other hand, we believe that situations in which the distribution between languages in the bilingual dataset is rather equal, such as the case of Albanian and English, could lead to better results while experimenting with BERT.



	Albanian, Danish	Albanian, English	Albanian, Turkish	Danish, English	Danish, Turkish	English, Turkish
Naïve Bayes	0,42	0,46	0,51	0,48	0,54	0,54
BiLSTM	0,70	0,71	0,70	0,73	0,68	0,70
BERT	0,79	0,78	0,77	0,75	0,74	0,76

Figure 6.2: Results from the Bilingual experiment.

6.1.3 Multilingual

In figure 6.3, we have included all the languages in both the training and testing partitions of the dataset. This setting was, by far, the most time and resource consuming due to the total size of training data constructed. Even though, by previous experiments, we believed that BERT would be outscoring BiLSTM, here we observe that the scores are quite similar, with BiLSTM actually surpassing BERT to the highest F1 score. This result would also serve as a good indication that the word embeddings prove to be a helpful common ground in multilingual exper-

iments. In comparison to the other experimental categories, we see a drop in performance for each classifier with any of the additions made. With these experiments, we can see that the baseline performs very poorly compared with the other two approaches. With this in mind, we can answer our hypotheses by saying that both BiLSTM and BERT seem to be an appropriate choice for dealing with multilingual hate speech and offensive speech detection.

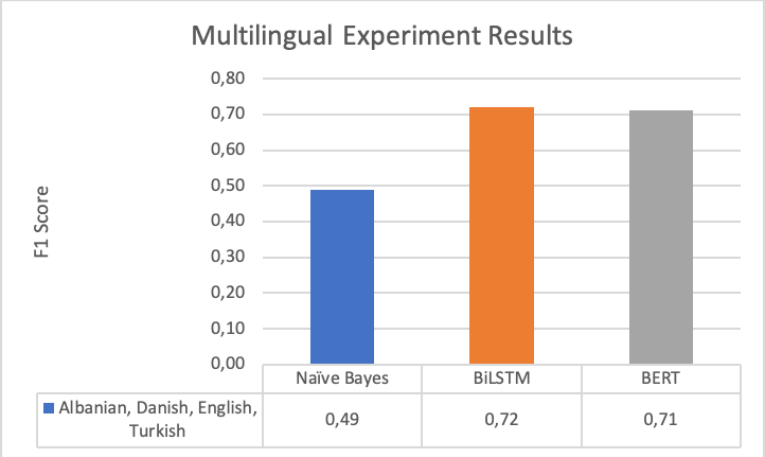
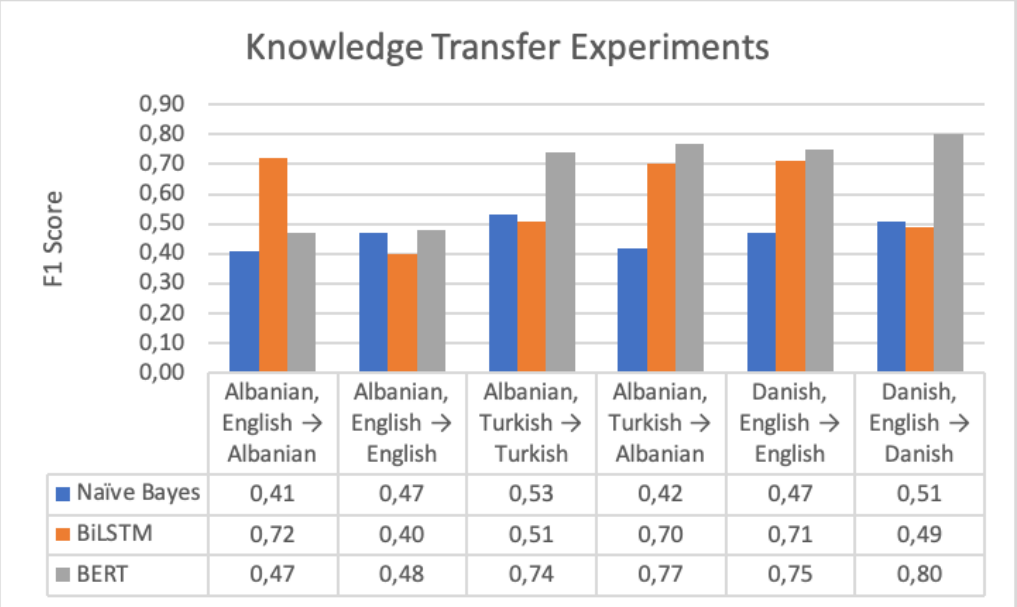


Figure 6.3: Results from the Multilingual experiment

6.1.4 Knowledge transfer

In the knowledge transfer category we still see a pattern where BERT is dominant on the F1 scores. This is portrayed in the graph shown in figure 6.4, depicting the best F1 scores achieved for all possible language combinations in this experiment. Interestingly, the first set of languages experimented, Albanian and English, testing on Albanian produces a large difference in score between BERT and BiLSTM classifiers. The misclassified sentences of this experiment will be part of a discussion included later in the analysis section of the chapter. A very exciting

outcome for this experiment is the F1 score of 0,80 obtained for the Danish language when English is added to the training set. In contrast to the F1 score of 0,71 that we have achieved for Danish in the monolingual experiment, we can see that the additional knowledge the model gains from the English dataset provides a significant boost in performance. This result affirms our expectations for the experiment and provides an affirmative response to the third question stated in the Introduction section(LINK), stating that models are able to learn from additional languages in the training phase, especially for languages with small datasets.



	Albanian, English → Albanian	Albanian, English → English	Albanian, Turkish → Turkish	Albanian, Turkish → Albanian	Danish, English → English	Danish, English → Danish
Naïve Bayes	0,41	0,47	0,53	0,42	0,47	0,51
BiLSTM	0,72	0,40	0,51	0,70	0,71	0,49
BERT	0,47	0,48	0,74	0,77	0,75	0,80

Figure 6.4: Results from the Knowledge Transfer Experiment.

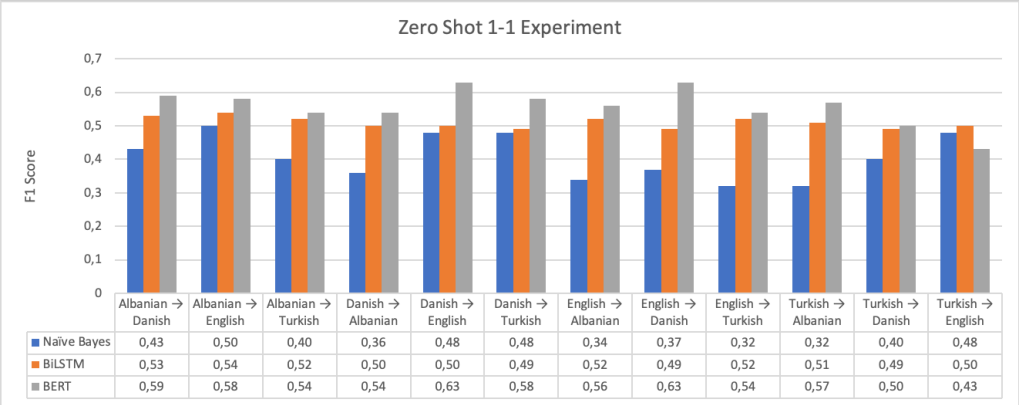
6.1.5 Zero shot

Zero shot 1-1

As shown in figure 6.5 he results obtained from the zero shot experiments, seem to disbalance the performance of BERT throughout all categories. The experiments of subsection 1-1, show that the combination of training in English language and testing on Danish perform the best with an F1 score of 0.63. Contrary to the pattern followed in previous types of experiments, BERT is also the worst performing classifier when training on Turkish and testing on English. We believe that the reason behind this concerns the sentence structure between the languages. In Turkish, one single word can also describe a whole sentence with a subject and an action. However in English we rarely see such occurrences of sentences. A second factor, might be that Turkish includes a lot more letters that are unknown in the English world such as ö and ı. Amongst the worst performing combination is also the training on Danish and testing in Turkish. The explanation above still holds for the difference in languages with the addition that we are training on very little data and testing one the largest dataset on included in our thesis.

Zero shot 2-1

When analysing the results of the Zero Shot 2-1 experiment displayed in figure 6.6, we can see that the model scoring the highest F1 score is BERT trained on the Albanian and Danish dataset and tested on the English dataset, which achieves a score of 0.63. This is an improved performance in comparison to the Zero Shot 1-1 experiment where we train the model in Albanian and test it in English, which scored an F1 score of 0.58. This example shows that even small datasets can provide



	Albanian → Danish	Albanian → English	Albanian → Turkish	Danish → Albanian	Danish → English	Danish → Turkish	English → Albanian	English → Danish	English → Turkish	Turkish → Albanian	Turkish → Danish	Turkish → English
Naïve Bayes	0,43	0,50	0,40	0,36	0,48	0,48	0,34	0,37	0,32	0,32	0,40	0,48
BiLSTM	0,53	0,54	0,52	0,50	0,50	0,49	0,52	0,49	0,52	0,51	0,49	0,50
BERT	0,59	0,58	0,54	0,54	0,63	0,58	0,56	0,63	0,54	0,57	0,50	0,43

Figure 6.5: Results from the Zero Shot 1-1 Experiment.

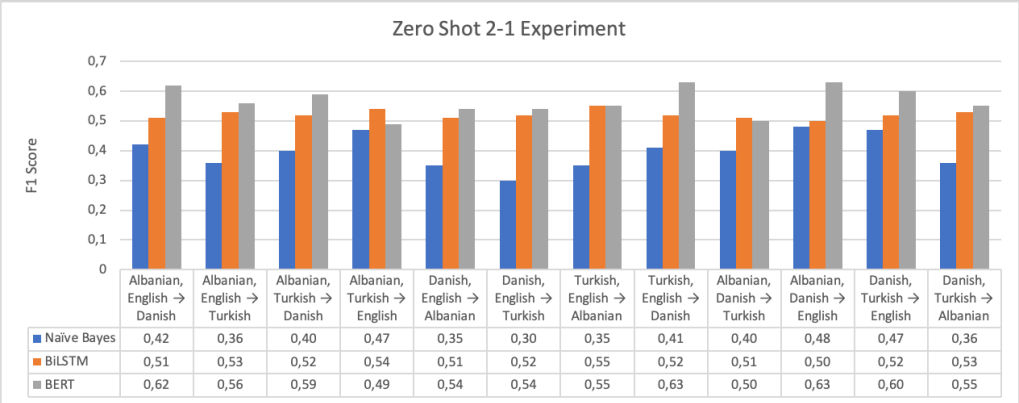
significant contribution in hate speech detection, if the languages chosen have some similarities between them.

The concept of language similarity contributing to a boost in performance is also visible when training the model on Albanian and English and then testing it on Danish, which gives an F1 score of 0.62 in comparison to 0.59 achieved when training in Albanian and testing it in Danish. However, the improvement in this case could also be attributed to the significant size of the added English dataset, which provides the 2-1 model with twice the data (and knowledge) of the 1-1 model.

A third example supporting the language similarity benefits in Knowledge Transfer experiments is the addition of the Danish dataset into the Turkish dataset while training, which is later tested on the English dataset. While training on the Turkish language alone, the model can only reach an F1 score of 0,50, whereas after the addition of the Danish data its performance is boosted by 0,10 and achieves the F1 score of 0,60 despite the small amount of data in this additional dataset.

A last example worth mentioning, is the poor performance of the BERT model when trained on Albanian and Turkish data and tested on the

English data, which receives a low F1 score of 0.49, almost as poor as the Naive Bayes baseline whose F1 score is only 0,47. This shows that even though the union of these two datasets creates a substantial data corpus of 45000 comments, the lack of similarity or common characteristics between English and the two other languages plays a factor in the creation of a model which is hardly useful for the task in hand.



	Albanian, English → Danish	Albanian, English → Turkish	Albanian, Turkish → Danish	Albanian, Turkish → English	Danish, English → Albanian	Danish, English → Turkish	Turkish, English → Albanian	Turkish, English → Danish	Albanian, Danish → Turkish	Albanian, Danish → English	Danish, Turkish → English	Danish, Turkish → Albanian
Naïve Bayes	0,42	0,36	0,40	0,47	0,35	0,30	0,35	0,41	0,40	0,48	0,47	0,36
BiLSTM	0,51	0,53	0,52	0,54	0,51	0,52	0,55	0,52	0,51	0,50	0,52	0,53
BERT	0,62	0,56	0,59	0,49	0,54	0,54	0,55	0,63	0,50	0,63	0,60	0,55

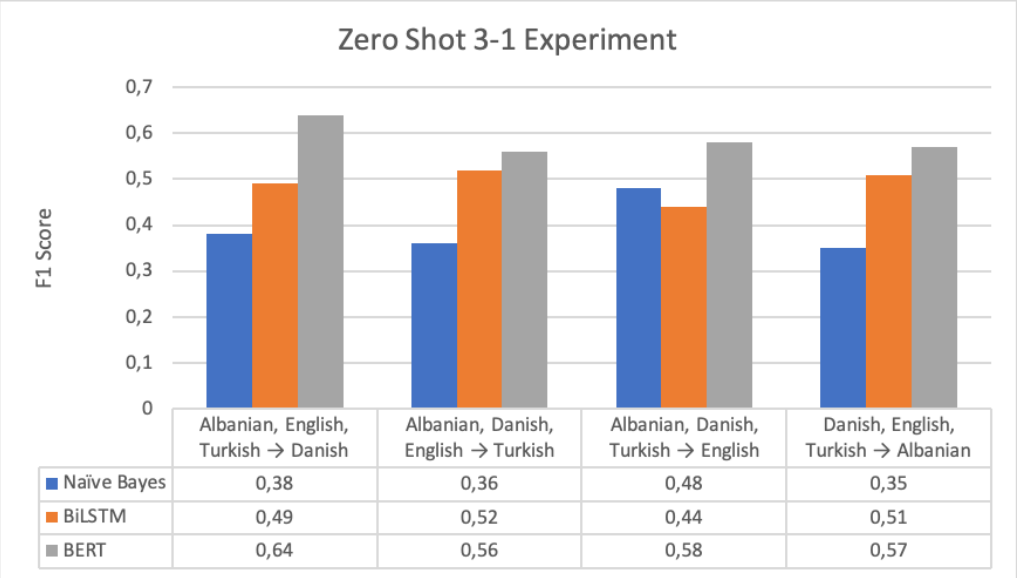
Figure 6.6: Results from the Zero shot 2-1 Experiment.

Zero shot 3-1

For the 3-1 setting of this experiment we achieved the best F1 score of 0.64 when testing on Danish and training on all other languages. We believe that this results from the large size of the training dataset and the knowledge incorporated in it, but also from the small size of the Danish dataset which renders this experiment somehow immaterial, even though the results seem promising. Without having a larger data corpus provided for Danish, we cannot draw any definite conclusion on why this combination performed the best.

The second best F1 score in this experiment is achieved when testing on the English dataset and training on the union of the three other datasets,

which provides an F1 score of 0.58. This result meets our expectations, given that the dataset provided for the task is quite large, but also because offensive content in social media often includes English words and especially English profanity, regardless of the language it is written in. However, it is worth noting that the two other combinations for this experiment obtain a similar performance with F1 scores of 0.57 and 0.56. This goes to show that once a model is trained on a variety of languages, it is able to perceive language-independent features of offensive speech, and offer almost the same efficiency despite the combination of languages chosen to participate in the training. On the downside, we can observe that this focus on language-independent features causes the model to ignore language specific traits of hate speech and have an average performance in the given task. All our best performing results for each classifier are presented in figure 6.7



	Albanian, English, Turkish → Danish	Albanian, Danish, English → Turkish	Albanian, Danish, Turkish → English	Danish, English, Turkish → Albanian
Naïve Bayes	0,38	0,36	0,48	0,35
BiLSTM	0,49	0,52	0,44	0,51
BERT	0,64	0,56	0,58	0,57

Figure 6.7: Results from the Zero Shot 3-1 Experiment.

In this section of the Results and Analysis chapter, we have provided

our best performing classifiers according to F1 score. We see a pattern emerging where BERT seems to be better suited for each of the categories we have experimented on. In the following section of the chapter, we will discuss some interesting results and ideas that we developed from running these experiments.

6.2 Analysis

6.2.1 Computational Limitations

Working with the multilingual environment has presented its own drawbacks during our experiments. Having to deal with multiple datasets simultaneously has notably increased the time necessary for carrying out the experiments. We therefore believe that more powerful machines are necessary to run the classifiers if the number of languages supported by them increases.

The preparation step for the experiments was significantly time consuming for BiLSTM in the multilingual setting given that one has to prepare the necessary files for the embedding layer prior to training the model. Using word vectors for the embedding made this task not only time consuming but also memory inefficient as the RAM was constantly being drained when creating the common vector files for all languages. This is the main reason why we decided to use the first occurrence vector when dealing with multiple languages.

Limitation in memory consumption is also observed with our baseline classifier. Even though this was one of the reasons we chose Naive Bayes as a baseline, the classifier did not manage to fit the whole dataset when training, as it would only support around 18000-20000 entries. Such

an environment restricts us from using the whole dataset for each language to experiment on the multilingual environment and we had to use small partitions of each dataset, which of course determines the performance of this model. The results in the chapter above clearly show the disadvantages Naive Bayes has towards Deep Learning models such as BiLSTM and the transformer approach of BERT.

6.2.2 Word vector Usage

The vector files for each language are trained on specific data sources and may not have word presentations that are suitable or realistic for all settings in which the word may be used. The vectors used in this work for example are trained on Common Crawl and Wikipedia and used on data extracted from social media, which tends to have a different type of communication compared to the vector data sources. Social media texts are often more informal, tend to use colloquialisms, lack punctuation, include region and time specific expressions, jokes, etc and often use words to convey information that is quite different from their original definition.

Another issue with using word vectors is that different ways of (incorrectly) spelling a word will either have different vectors present in the file, or end up as unrepresented in the dataset. Words such as "hahaha", "ahhaha", "haahhaha", etc will be perceived as different words by the model, when they are in fact the same. Unluckily for us, misspelling words is very common in social media, and often done intentionally to avoid automated offensive language detection tools.

Additionally, words with the same spelling may have different meanings in different languages. By this, we tackle the sets of characters

that happen to be words in multiple languages, and are not necessarily translations of each other. A concrete example within the datasets used in this thesis would be the word "kar" which is the Turkish word for snow but in Albanian it translates to dick, which is an offensive term. To handle such situations, we experimented with different approaches of using word vectors in multilingual settings, including the bilingual one.

The first approach was to neutralize the perceived meaning of the word to the model by calculating an average of its representations in different languages. The first downside of this approach was that it was quite memory consuming when dealing with two languages, and virtually impossible to compute with more languages. Another drawback of this approach was that even though a word might have a negative or offensive meaning in one language, having a positive meaning in another language would balance out its vector. An even worse scenario would be if the word is positive in several other languages, in which case the model would tend to continuously associate the word with a positive connotation, regardless of the language it is used in.

The second approach for these words, and the one we chose to follow, was to only consider the first occurrence of the word vector in all of the considered files. A strategic approach in multilingual experiments would be to order the vector files before feeding them into the model, starting with the ones that have been trained the most or in largest datasets. In knowledge transfer experiments however, the more sensible approach is to assign higher importance to the vectors of the "main" language for the experiment, the one that is included in both the training and testing partition of the data. The drawback that was previously stated for the average vector approach still stands when words have con-

trasting sentiments in different languages and a pseudo-random vector is chosen. However, given that this task is unsolvable without language recognition and language dependent vector tailoring, one would prefer choosing a pseudo-random vector in a short amount of time, rather than going through the memory inefficient approach of averaging vectors and still getting a similar result.

The last point of discussion related to the word vector embeddings in our experiments has to do with our choice of allowing the model to train and therefore change these vectors. While this allows for better performance of the model, one can say that it somehow undermines the importance of the using word vector embeddings, as the model freely reassigns new meanings to the words according to the data included in the test set. Another risk associated with this decision is that the model may overfit and become too dependent on the data it is trained on. On the other hand, other approaches where these vectors should not be trained but remain in the state they are found in the vector files, might cause classifiers not to produce the best possible results.

6.2.3 Error Analysis

Misclassification and patterns emerging

During the conducted experiments we have seen a pattern emerging when dealing categories that include Danish in their testing split of data. Compared to the other languages, the classifiers seem to mislabel more Danish sentences than any other language. This pattern is not followed closely to specific sentences, however by studying the confusion matrix we have observed this phenomenon. We believe that this pattern has to do this the size of the Danish dataset. Comparing it with other datasets,

e.g English, the whole size of the Danish dataset is half of the size of offensive labels in the English one. According to statistics from (REF = hatespeechdata.com) for the danish dataset, it contains an abusive percentage of only 0.12 whereas the OLID dataset contains almost 0.33 of the latter (REF = Predicting the Type and Target of Offensive Posts in Social Media). Having the abusive percentage concept in mind, we also calculated it for the Albanian dataset that we have created and obtained a ratio of 0.13 abusive content versus non offensive data. While working with the Albanian dataset, we have observed that it includes several sentences consisting of only one word, or one emoticon or even just a tagged person. In addition it often has comments which are very similar or identical with each other. In order to have a better constructed dataset, we believe that such comments should be omitted or replaced with more meaningful sentences, and that perhaps we should make an effort to include more abusive language in it.

Monolingual BERT:

Another interesting occurrence that we found during these experiments has to do with the labels already given to sentences. For this discussion, we are focusing only on English and Albanian, as we can understand these languages the best. During the prediction phase, we kept a record of sentences that the models misclassified in accordance with the provided dataset. The example sentences are part of the predictions coming from the monolingual setting using BERT as our classifier, but table x in the appendix would show other examples throughout all classifiers used and categories experimented on. Going through these mislabelled predictions, we found out that some of them were given a wrong label to begin with. For example in the Albanian dataset, the sentence:

âĂIJku ka magji qe i mer ato persiper mâĂİ

Which translates to *âĂIJthere is no magic that takes care of theseâĂİ* was identified as an offensive sentence in accordance with the SemiEval schema. The model however, had âĂŸmisclassifiedâĂŹ this sentence as not offensive and sprung a discussion between the authors whether the underlying message of the text was truly offensive or not. One of the annotators found this offensive because the expression is often used to say that "they" are in such a bad state that even magic cannot improve (concerning beauty or some other personal trait). This is basically saying that something/ someone is bad beyond repair which is expressed in an offensive way The opposite mislabel mistake was also observed where the sentence: âĂIJjepja leket ti per te pagujt dritta uj koqe meru me nai pun tjeter fuksâĂİ Which translates to âĂIJyou give them the money to pay electricity and water you dick, go and deal with something else *other offensive term* âĂIJ. In our dataset this sentence was labelled as not offensive, but the usage of the words âĂIJkoqeâĂİ and âĂIJfuksâĂİ should result in the sentence being given an offensive label targeted to an individual.

The same issue is observed in the English dataset where the sentence: âĂIJ@user i never talked about gun control in any formâĂİ This sentence was given an offensive label whereas for the model, it was considered as not offensive. The vice-versa situation can be observed in sentences like: @user @user i donâĂŹt give a damn about a âĂIJ tough âĂİ niggah LAUGHING$_{E}MOJIWaslabelledasnotoffensive, wheretheusageofoffensivewordstowardscertai$

Even though the models seem to have several strengths, throughout the experiments we have noticed that they are suffering from two main and

very common issues when dealing with offensive language. We have encountered sentences that clearly show signs of sarcasm or are derogatory in context without the usage of any profanity or swear words. @user @user f hope ! since heâĂŹs âĂŸ not smarter than a 5th grader âĂŹ maybe she is ! te themi tu befshin ilace e ke pak per ty duhet me shpik mallkim t ri Literal translation of previous: let's say you take medicine and you have a little for you, I have to invent a new curse Adapted translation: wishing you that they (your money) ends up buying you medicine is too little, you deserve the invention of a new curse tamam parkim femre Translation of previous: typical female parking bjonde do kete qene me siguri Translation of previous: she must have been a blonde for sure

This phenomenon comes from the words being used having multiple meanings in daily life. It seems that BERT identifies exactly, women, and parking as not offensive words, leaving behind the general stereotypical behavior which is constantly being targeted as offensive language and thus hurtful to the group they identify.

We also observed that the model was very often unable to detect offensive speech veiled under sarcastic tones, but it was very quick to detect comments containing repetitive explicit offensive words.

Similar issues are discussed in (**nurce_keci**).

6.2.4 Discussions

Imbalanced-learn Library Usage

As we have previously mentioned in chapter Experimental Setup, we make use of the imbalanced-learn library for python to tweak the imbalance between offensive and not offensive data in the datasets. This

is possible by defining the desired ratio we define when training. We experiment with different distributions close to the 50-50 region to get insights in any possible performance increase. Seeing that the imbalance was really high, we thought that the results coming from undersampling our data would produce better results. The only noticeable results were noticed in the Monolingual category for English and in the Multilingual category both running with the BiLSTM classifier. The difference between normally distributed labels and the tweaks made from the imbalance-learn library was just 1%.

With all the content generated in social media, it is more common that people would not insult or offend anyone on their first instinct and we believe that this is reflected on the datasets' creation. Having an imbalance between the labels offers a more realistic setting of the datasets and training and evaluating classifiers with such data would be beneficial in creating a stable and effective hate speech detection system.

Chapter 7

Conclusion

Online hate speech detection has proven to be a difficult task when considering multiple languages. The research being done for the problem in hand has been mainly focused on the English language while other languages remain threatened by the extensive use of hate speech online. Without having a proper mechanism of handling this detection in less widespread languages, we expand the horizon of hate speech into a multilingual setting by including languages such as Albanian, Danish and Turkish.

7.1 Contribution

7.1.1 An expanded Albanian Dataset

With this work we contribute by establishing the largest Albanian dataset of its kind, and the first one to be made publicly available for future research in the field. Seeing that the OffensEval Schema is being used more and more in the Natural Processing Language task of hate

speech, we make ground for the usage of Albanian as a language to train classifiers to detect hate speech.

7.1.2 Advantages and disadvantages of various multilingual HS detection

We restate the questions we have posed in the Introduction chapter of our paper and provide answers for each one based on the outcomes of the experiments we have carried out.

1. How do these models behave in the monolingual environment?
   Through the results of our experiments, we observe that the models achieve their best performances when employed in the monolingual setting. This results from the ability of the model to pick up general features in hate speech, but also language dependent ones. Therefore, whenever a monolingual hate speech detection tool is a viable option, we recommend it as the one that has statistically shown to perform a better classification.

2. How do these models behave when a mixture of languages is used to train and test them? Furthermore, is there a combination of languages which proves to be more effective?
   In our bilingual and multilingual experiments, we have seen how models trained and tested on a combination of languages manage to achieve quite high performances, and are a good choice in naturally multilingual environments. They pick up common characteristics of hate speech in the respective languages, and obtain good results in classification.
   To answer the second part of the question, we would argue that junctions of datasets that are comprised of similar languages from

the same language families or others that contain a large subset of common words and expressions actually give a better performance. It is necessary, however, for datasets to be of similar sizes, so that the model gains a somewhat equal knowledge from each language.

3. Can these models learn from additional languages added during training to provide a better detection on specific languages?
   During our knowledge transfer experiments, we saw that knowledge transfer isn't quite an easy task, as it can sometimes help the model into making good decisions, but oftentimes also induce more confusion in the classification process. We did observe however that for small datasets the addition of larger datasets of similar languages can truly improve a model's performance.

4. How effective are these models in detecting hate speech in languages they have not been trained on?
   This question was tackled in our 0-shot experiments, where we conducted the largest amount of trials for all combinations of languages in training and testing. As we add more languages to the training dataset, we notice that the F1 scores generally increase for the models, and that is because a common understanding of offensive speech is being created. However, the results available from these experiments show that at least for the time being "off-the-shelf" hate speech detection models will not produce an impressive result when used on languages they have not been trained on. However, a glimpse of hope still exists as some of the results for these experiments are comparable with the ones in previous experiments.

5. How robust are these classifiers when training and testing with many languages at the same time?
   Our highest F1 Score of 0,71 for the multilingual experiment is on par with the metrics given in many recent publications. This goes to show that truly multilingual hate speech detection is possible and can achieve very good results.

6. How can word embeddings contribute to improving detection across different languages?
   We believe that word embeddings provide a good common ground for multilingual hate speech detection. However, due to our decision to allow the models to train and therefore change the embeddings in the process, we cannot offer a decisive outcome for the question in hand.

The experiments conducted in this thesis, provide new insights on how different classifiers are build to handle detection in various languages and how well they perform in doing so. We believe that BERT is a reliable and efficient model to be used when dealing with the multilingual task.

7.2 Future Work

As future work, we would like to refine the Albanian dataset even further by including more representations of various forms of hate speech. We would also like to work towards the diversification of the dataset, be it for the chosen data sources, but also for repetitive or very similar comments that appear several times in the data corpus.

In terms of multilingual hate speech detection, we want to see if the classifiers can be tested with more languages in order to reach a more efficient a language-independent detection. We would also like to experiment with newer and more elaborate models presented in very recent works, that have shown remarkable performance even for settings similar to our zero shot experiment.

Appendix A

Complete List of Results

A.1 Monolingual Experiment Results

Beginning of Monolingual Results' Table						
Language	**Model**	**Recall**	**Prec.**	**F1 Score**	**F2 Score**	**F0,5 Score**
Albanian	BERT	0.75	0.83	0.78	0.76	0.81
Albanian	BERT	0.76	0.84	0.79	0.77	0.82
Albanian	BERT	0.76	0.84	0.79	0.77	0.82
Albanian	BERT	0.76	0.84	0.79	0.77	0.82
Albanian	BERT	0.76	0.85	0.80	0.78	0.83
Albanian	Gaussian Naive Bayes	0.51	0.51	0.42	0.51	0.51
Albanian	Gaussian Naive Bayes	0.52	0.51	0.42	0.52	0.51
Albanian	LSTM	0.59	0.61	0.57	0.59	0.61
Continued in the next page ...						

Continuation of Monolingual Results' Table						
Language	**Model**	**Recall**	**Prec.**	**F1 Score**	**F2 Score**	**F0,5 Score**
Albanian	LSTM	0.59	0.66	0.57	0.6	0.64
Albanian	LSTM	0.59	0.68	0.57	0.61	0.66
Albanian	LSTM	0.61	0.61	0.61	0.61	0.61
Albanian	LSTM	0.63	0.74	0.66	0.65	0.72
Albanian	LSTM	0.66	0.69	0.67	0.67	0.68
Albanian	LSTM	0.69	0.73	0.71	0.7	0.72
Danish	BERT	0.66	0.85	0.71	0.69	0.8
Danish	Gaussian Naive Bayes	0.55	0.53	0.51	0.55	0.53
Danish	LSTM	0.61	0.65	0.63	0.62	0.64
Danish	LSTM	0.64	0.65	0.64	0.64	0.65
Danish	LSTM	0.65	0.6	0.61	0.64	0.61
Danish	LSTM	0.66	0.64	0.65	0.66	0.64
Danish	LSTM	0.68	0.71	0.69	0.69	0.7
English	BERT	0.75	0.75	0.75	0.75	0.75
English	BERT	0.75	0.75	0.75	0.75	0.75
English	BERT	0.76	0.78	0.77	0.76	0.78
English	Gaussian Naive Bayes	0.53	0.53	0.47	0.53	0.53
English	LSTM	0.5	0.84	0.40	0.54	0.74
English	LSTM	0.68	0.69	0.68	0.68	0.69
English	LSTM	0.68	0.69	0.69	0.68	0.69

Continued in the next page ...

Continuation of Monolingual Results' Table						
Language	**Model**	**Recall**	**Prec.**	**F1 Score**	**F2 Score**	**F0,5 Score**
English	LSTM	0.69	0.7	0.70	0.69	0.7
English	LSTM	0.7	0.69	0.70	0.7	0.69
English	LSTM	0.7	0.7	0.70	0.7	0.7
English	LSTM	0.72	0.72	0.72	0.72	0.72
English	LSTM	0.73	0.73	0.73	0.73	0.73
Turkish	BERT	0.74	0.79	0.76	0.75	0.78
Turkish	Gaussian Naive Bayes	0.57	0.55	0.54	0.57	0.55
Turkish	LSTM	0.67	0.67	0.67	0.67	0.67
Turkish	LSTM	0.68	0.7	0.69	0.68	0.7
Turkish	LSTM	0.69	0.68	0.69	0.69	0.68
Turkish	LSTM	0.7	0.71	0.70	0.7	0.71

Table A.1: Complete list of results for the Monolingual Experiment

End of Monolingual Results' Table

A.2 Bilingual Experiment Results

Beginning of Bilingual Results' Table						
Language	**Model**	**Recall**	**Prec.**	**F1 Score**	**F2 Score**	**F0,5 Score**
Albanian, Danish	BERT	0.76	0.83	0.79	0.77	0.81
Albanian, Danish	Gaussian Naive Bayes	0.49	0.5	0.42	0.49	0.5
Albanian, Danish	LSTM	0.63	0.72	0.66	0.65	0.7
Albanian, Danish	LSTM	0.65	0.72	0.68	0.66	0.7
Albanian, Danish	LSTM	0.66	0.71	0.68	0.67	0.7
Albanian, Danish	LSTM	0.68	0.68	0.68	0.68	0.68
Albanian, Danish	LSTM	0.68	0.69	0.69	0.68	0.69
Albanian, Danish	LSTM	0.69	0.7	0.70	0.69	0.7
Albanian, Danish First Vector	LSTM	0.65	0.74	0.68	0.67	0.72
Continued in the next page ...						

Continuation of Bilingual Results' Table						
Language	**Model**	**Recall**	**Prec.**	**F1 Score**	**F2 Score**	**F0,5 Score**
Albanian, Danish First Vector	LSTM	0.71	0.7	0.71	0.71	0.7
Albanian, English	BERT	0.77	0.79	0.78	0.77	0.79
Albanian, English	Gaussian Naive Bayes	0.53	0.52	0.46	0.53	0.52
Albanian, English	LSTM	0.68	0.71	0.69	0.69	0.7
Albanian, English	LSTM	0.7	0.71	0.71	0.7	0.71
Albanian, English	LSTM	0.7	0.72	0.70	0.7	0.72
Albanian, English	LSTM	0.71	0.71	0.71	0.71	0.71
Albanian, English One Vector	LSTM	0.7	0.73	0.71	0.71	0.72
Albanian, Turkish	BERT	0.73	0.79	0.76	0.74	0.78
Albanian, Turkish	BERT	0.74	0.8	0.77	0.75	0.79

Continued in the next page ...

Continuation of Bilingual Results' Table						
Language	**Model**	**Recall**	**Prec.**	**F1 Score**	**F2 Score**	**F0,5 Score**
Albanian, Turkish	BERT	0.74	0.81	0.76	0.75	0.79
Albanian, Turkish	Gaussian Naive Bayes	0.54	0.52	0.51	0.54	0.52
Albanian, Turkish	LSTM	0.68	0.72	0.70	0.69	0.71
Albanian, Turkish	LSTM	0.69	0.7	0.69	0.69	0.7
Albanian, Turkish	LSTM	0.7	0.71	0.70	0.7	0.71
Albanian, Turkish one vector	LSTM	0.67	0.69	0.68	0.67	0.69
Danish, English	BERT	0.49	0.45	0.43	0.48	0.46
Danish, English	BERT	0.5	0.35	0.41	0.46	0.37
Danish, English	BERT	0.52	0.54	0.49	0.52	0.54
Danish, English	BERT	0.75	0.76	0.75	0.75	0.76
Danish, English	Gaussian Naive Bayes	0.53	0.52	0.48	0.53	0.52

Continued in the next page ...

Continuation of Bilingual Results' Table						
Language	**Model**	**Recall**	**Prec.**	**F1 Score**	**F2 Score**	**F0,5 Score**
Danish, English	LSTM	0.68	0.73	0.70	0.69	0.72
Danish, English	LSTM	0.69	0.69	0.69	0.69	0.69
Danish, English	LSTM	0.69	0.7	0.69	0.69	0.7
Danish, English	LSTM	0.72	0.72	0.72	0.72	0.72
Danish, English Only One vector	LSTM	0.68	0.68	0.68	0.68	0.68
Danish, English Only One vector	LSTM	0.71	0.73	0.72	0.71	0.73
Danish, English Only One vector	LSTM	0.73	0.74	0.73	0.73	0.74
Danish, Turkish	BERT	0.72	0.77	0.74	0.73	0.76
Continued in the next page ...						

Continuation of Bilingual Results' Table

Language	**Model**	**Recall**	**Prec.**	**F1 Score**	**F2 Score**	**F0,5 Score**
Danish, Turkish	BERT	0.73	0.76	0.74	0.74	0.75
Danish, Turkish	Gaussian Naive Bayes	0.55	0.54	0.54	0.55	0.54
Danish, Turkish	LSTM	0.66	0.72	0.68	0.67	0.71
English, Turkish	BERT	0.75	0.77	0.76	0.75	0.77
English, Turkish	Gaussian Naive Bayes	0.57	0.55	0.54	0.57	0.55
English, Turkish	LSTM	0.68	0.7	0.69	0.68	0.7
English, Turkish	LSTM	0.71	0.7	0.70	0.71	0.7

Table A.2: Complete list of results for the Bilingual Experiment

End of Bilingual Results' Table

A.3 Multilingual Experiment Results

Beginning of Multilingual Results' Table						
Language	**Model**	**Recall**	**Prec.**	**F1 Score**	**F2 Score**	**F0,5 Score**
Turkish, Danish, Albanian, English	Gaussian Naive Bayes	0.53	0.52	0.48	0.53	0.52
Turkish, Danish, Albanian, English	Gaussian Naive Bayes	0.54	0.53	0.49	0.54	0.53
Turkish, Danish, Albanian, English	Gaussian Naive Bayes	0.54	0.53	0.49	0.54	0.53
Turkish, Danish, Albanian, English	LSTM	0.69	0.69	0.69	0.69	0.69
Turkish, Danish, Albanian, English	LSTM	0.69	0.7	0.69	0.69	0.7
Continued in the next page ...						

Continuation of Multilingual Results' Table						
Language	**Model**	**Recall**	**Prec.**	**F1 Score**	**F2 Score**	**F0,5 Score**
Turkish, Danish, Albanian, English	LSTM	0.7	0.69	0.69	0.7	0.69
Turkish, Danish, Albanian, English	LSTM	0.7	0.69	0.69	0.7	0.69
Turkish, Danish, Albanian, English	LSTM	0.7	0.69	0.70	0.7	0.69
Turkish, Danish, Albanian, English	LSTM	0.7	0.7	0.70	0.7	0.7
Turkish, Danish, Albanian, English	LSTM	0.7	0.72	0.71	0.7	0.72
Continued in the next page ...						

Continuation of Multilingual Results' Table						
Language	**Model**	**Recall**	**Prec.**	**F1 Score**	**F2 Score**	**F0,5 Score**
Turkish, Danish, Albanian, English	LSTM	0.71	0.7	0.71	0.71	0.7
Turkish, Danish, Albanian, English	LSTM	0.71	0.71	0.71	0.71	0.71
Turkish, Danish, Albanian, English	LSTM	0.72	0.72	0.72	0.72	0.72

Table A.3: Complete list of results for the Multi1ingual Experiment

End of Multilingual Results' Table

A.4 Knowledge Transfer Experiment Results

Beginning of Knowledge Transfer Results' Table							
Training Languages	**Testing Language**	**Model**	**Recall**	**Prec.**	**F1 Score**	**F2 Score**	**F0,5 Score**
Albanian, Danish	Albanian	BERT	0.5	0.57	0.47	0.51	0.55
Albanian, English	Albanian	BERT	0.5	0.51	0.47	0.5	0.51
Albanian, English	Albanian	BERT	0.5	0.52	0.47	0.5	0.52
Albanian, English	Albanian	BERT	0.78	0.78	0.78	0.78	0.78
Albanian, English	Albanian	Gaussian Naive Bayes	0.51	0.5	0.41	0.51	0.5
Albanian, English	Albanian	LSTM	0.69	0.71	0.70	0.69	0.71
Albanian, English	Albanian	LSTM	0.69	0.71	0.70	0.69	0.71
Albanian, English	Albanian	LSTM	0.7	0.71	0.70	0.7	0.71
Albanian, English	Albanian	LSTM	0.71	0.69	0.70	0.71	0.69
Albanian, English	Albanian	LSTM	0.71	0.69	0.70	0.71	0.69
Continued in the next page ...							

Continuation of Knowledge Transfer Results' Table							
Training Languages	**Testing Language**	**Model**	**Recall**	**Prec.**	**F1 Score**	**F2 Score**	**F0,5 Score**
Albanian, English	Albanian	LSTM	0.71	0.73	0.72	0.71	0.73
Albanian, English	English	BERT	0.5	0.5	0.40	0.5	0.5
Albanian, English	English	BERT	0.51	0.51	0.51	0.51	0.51
Albanian, English	English	BERT	0.75	0.75	0.75	0.75	0.75
Albanian, English	English	Gaussian Naive Bayes	0.52	0.52	0.47	0.52	0.52
Albanian, English	English	LSTM	0.5	0.49	0.44	0.5	0.49
Albanian, English	English	LSTM	0.5	0.49	0.46	0.5	0.49
Albanian, English	English	LSTM	0.5	0.49	0.48	0.5	0.49
Albanian, English	English	LSTM	0.5	0.5	0.43	0.5	0.5
Albanian, English	English	LSTM	0.5	0.51	0.45	0.5	0.51
Albanian, English	English	LSTM	0.5	0.51	0.46	0.5	0.51

Continued in the next page ...

Continuation of Knowledge Transfer Results' Table							
Training Languages	**Testing Language**	**Model**	**Recall**	**Prec.**	**F1 Score**	**F2 Score**	**F0,5 Score**
Albanian, English	English	LSTM	0.51	0.52	0.45	0.51	0.52
Albanian, English	English	LSTM	0.51	0.53	0.45	0.51	0.53
Albanian, English One Vector	Albanian	LSTM	0.68	0.71	0.69	0.69	0.7
Albanian, English One vector	Albanian	LSTM	0.69	0.72	0.70	0.7	0.71
Albanian, Turkish	Albanian	BERT	0.74	0.8	0.76	0.75	0.79
Albanian, Turkish	Albanian	BERT	0.75	0.79	0.77	0.76	0.78
Albanian, Turkish	Albanian	Gaussian Naive Bayes	0.52	0.51	0.42	0.52	0.51
Albanian, Turkish	Albanian	LSTM	0.69	0.69	0.69	0.69	0.69
Albanian, Turkish	Turkish	BERT	0.5	0.41	0.45	0.48	0.43
Albanian, Turkish	Turkish	BERT	0.73	0.79	0.76	0.74	0.78

Continued in the next page ...

Continuation of Knowledge Transfer Results' Table							
Training Languages	**Testing Language**	**Model**	**Recall**	**Prec.**	**F1 Score**	**F2 Score**	**F0,5 Score**
Albanian, Turkish	Turkish	Gaussian Naive Bayes	0.55	0.54	0.53	0.55	0.54
Albanian, Turkish	Turkish	LSTM	0.5	0.5	0.50	0.5	0.5
Albanian, Turkish	Turkish	LSTM	0.5	0.5	0.50	0.5	0.5
Albanian, Turkish one vector	Albanian	LSTM	0.69	0.71	0.70	0.69	0.71
Albanian, Turkish one vector	Turkish	LSTM	0.51	0.51	0.51	0.51	0.51
Albanian,Turkish	Albanian	LSTM	0.7	0.69	0.69	0.7	0.69
Danish, English	Danish	BERT	0.78	0.81	0.80	0.79	0.8
Danish, English	Danish	BERT	0.79	0.81	0.80	0.79	0.81
Danish, English	Danish	BERT	0.79	0.81	0.80	0.79	0.81
Danish, English	Danish	Gaussian Naive Bayes	0.53	0.52	0.51	0.53	0.52
Danish, English	Danish	Gaussian Naive Bayes	0.54	0.52	0.50	0.54	0.52

Continued in the next page ...

Continuation of Knowledge Transfer Results' Table							
Training Languages	**Testing Language**	**Model**	**Recall**	**Prec.**	**F1 Score**	**F2 Score**	**F0,5 Score**
Danish, English	Danish	LSTM	0.46	0.48	0.45	0.46	0.48
Danish, English	Danish	LSTM	0.47	0.48	0.45	0.47	0.48
Danish, English	Danish	LSTM	0.47	0.48	0.46	0.47	0.48
Danish, English	Danish	LSTM	0.48	0.49	0.46	0.48	0.49
Danish, English	Danish	LSTM	0.49	0.49	0.47	0.49	0.49
Danish, English	Danish	LSTM	0.49	0.49	0.48	0.49	0.49
Danish, English	Danish	LSTM	0.49	0.49	0.48	0.49	0.49
Danish, English	Danish	LSTM	0.49	0.5	0.47	0.49	0.5
Danish, English	Danish	LSTM	0.5	0.5	0.48	0.5	0.5
Danish, English	Danish	LSTM	0.53	0.52	0.49	0.53	0.52
Danish, English	English	BERT	0.75	0.75	0.75	0.75	0.75

Continued in the next page ...

Continuation of Knowledge Transfer Results' Table							
Training Languages	**Testing Language**	**Model**	**Recall**	**Prec.**	**F1 Score**	**F2 Score**	**F0,5 Score**
Danish, English	English	Gaussian Naive Bayes	0.52	0.52	0.47	0.52	0.52
Danish, English	English	LSTM	0.67	0.66	0.62	0.67	0.66
Danish, English	English	LSTM	0.67	0.67	0.67	0.67	0.67
Danish, English	English	LSTM	0.68	0.66	0.66	0.68	0.66
Danish, English	English	LSTM	0.68	0.67	0.65	0.68	0.67
Danish, English	English	LSTM	0.68	0.68	0.68	0.68	0.68
Danish, English	English	LSTM	0.68	0.68	0.68	0.68	0.68
Danish, English	English	LSTM	0.68	0.69	0.69	0.68	0.69
Danish, English	English	LSTM	0.69	0.67	0.67	0.69	0.67
Danish, English	English	LSTM	0.69	0.69	0.69	0.69	0.69
Danish, English	English	LSTM	0.69	0.69	0.69	0.69	0.69

Continued in the next page ...

Continuation of Knowledge Transfer Results' Table							
Training Languages	**Testing Language**	**Model**	**Recall**	**Prec.**	**F1 Score**	**F2 Score**	**F0,5 Score**
Danish, English	English	LSTM	0.69	0.7	0.69	0.69	0.7
Danish, English	English	LSTM	0.69	0.71	0.70	0.69	0.71
Danish, English	English	LSTM	0.7	0.69	0.70	0.7	0.69
Danish, English	English	LSTM	0.7	0.7	0.70	0.7	0.7
Danish, English	English	LSTM	0.7	0.71	0.71	0.7	0.71
Danish, English one vector	Danish	LSTM	0.46	0.48	0.45	0.46	0.48

Table A.4: Complete list of results for the Knowledge Transfer Experiment

End of Knowledge Transfer Results' Table

A.5 Zero Shot Experiment Results

A.5.1 Zero shot 1-1

Beginning of Zero Shot 1-1 Results' Table							
Training Language	**Testing Language**	**Model**	**Recall**	**Prec.**	**F1 Score**	**F2 Score**	**F0,5 Score**
Albanian	**Danish**	**BERT**	**0.57**	**0.65**	**0.59**	**0.58**	**0.63**
Albanian	Danish	Gaussian Naive Bayes	0.49	0.5	0.43	0.49	0.5
Albanian	Danish	LSTM	0.55	0.53	0.53	0.55	0.53
Albanian	**English**	**BERT**	**0.58**	**0.6**	**0.58**	**0.58**	**0.6**
Albanian	English	Gaussian Naive Bayes	0.5	0.5	0.50	0.5	0.5
Albanian	English	LSTM	0.54	0.54	0.54	0.54	0.54
Albanian	**Turkish**	**BERT**	**0.54**	**0.56**	**0.54**	**0.54**	**0.56**
Albanian	Turkish	Gaussian Naive Bayes	0.47	0.48	0.40	0.47	0.48
Albanian	Turkish	Gaussian Naive Bayes	0.49	0.49	0.32	0.49	0.49
Albanian	Turkish	LSTM	0.53	0.52	0.52	0.53	0.52
Danish	**Albanian**	**BERT**	**0.53**	**0.56**	**0.54**	**0.54**	**0.55**
Danish	Albanian	Gaussian Naive Bayes	0.43	0.47	0.36	0.44	0.46
Danish	Albanian	LSTM	0.5	0.51	0.50	0.5	0.51
Danish	**English**	**BERT**	**0.59**	**0.69**	**0.58**	**0.61**	**0.67**
Continued in the next page ...							

Continuation of Zero Shot 1-1 Results' Table							
Training Language	**Testing Language**	**Model**	**Recall**	**Prec.**	**F1 Score**	**F2 Score**	**F0,5 Score**
Danish	English	Gaussian Naive Bayes	0.5	0.5	0.48	0.5	0.5
Danish	English	LSTM	0.52	0.55	0.49	0.53	0.54
Danish	Turkish	BERT	0.51	0.52	0.49	0.51	0.52
Danish	Turkish	Gaussian Naive Bayes	0.48	0.49	0.35	0.48	0.49
Danish	**Turkish**	**LSTM**	**0.5**	**0.5**	**0.50**	**0.5**	**0.5**
English	**Albanian**	**BERT**	**0.55**	**0.56**	**0.56**	**0.55**	**0.56**
English	Albanian	Gaussian Naive Bayes	0.41	0.46	0.34	0.42	0.45
English	Albanian	LSTM	0.52	0.53	0.52	0.52	0.53
English	**Danish**	**BERT**	**0.64**	**0.63**	**0.63**	**0.64**	**0.63**
English	Danish	Gaussian Naive Bayes	0.49	0.49	0.37	0.49	0.49
English	Danish	LSTM	0.57	0.53	0.49	0.56	0.54
English	**Turkish**	**BERT**	**0.54**	**0.54**	**0.54**	**0.54**	**0.54**
English	Turkish	LSTM	0.54	0.53	0.52	0.54	0.53
Turkish	**Albanian**	**BERT**	**0.56**	**0.58**	**0.57**	**0.56**	**0.58**
Turkish	Albanian	Gaussian Naive Bayes	0.42	0.46	0.32	0.43	0.45
Turkish	Albanian	LSTM	0.54	0.52	0.51	0.54	0.52

Continued in the next page ...

Continuation of Zero Shot 1-1 Results' Table

Training Language	**Testing Language**	**Model**	**Recall**	**Prec.**	**F1 Score**	**F2 Score**	**F0,5 Score**
Turkish	**Danish**	**BERT**	**0.51**	**0.56**	**0.50**	**0.52**	**0.55**
Turkish	Danish	Gaussian Naive Bayes	0.49	0.49	0.40	0.49	0.49
Turkish	Danish	LSTM	0.52	0.51	0.49	0.52	0.51
Turkish	English	BERT	0.51	0.59	0.43	0.52	0.57
Turkish	English	Gaussian Naive Bayes	0.51	0.51	0.48	0.51	0.51
Turkish	**English**	**LSTM**	**0.52**	**0.54**	**0.50**	**0.52**	**0.54**

Table A.5: Complete list of results for the Zero Shot 1-1 Experiment

End of Zero Shot 1-1 Results' Table

A.5.2 Zero shot 2-1

Beginning of Zero Shot 2-1 Results' Table							
Training Languages	**Testing Language**	**Model**	**Recall**	**Prec.**	**F1 Score**	**F2 Score**	**F0,5 Score**
Albanian, Danish	English	BERT	0.63	0.69	0.63	0.64	0.68
Albanian, Danish	English	Gaussian Naive Bayes	0.49	0.49	0.48	0.49	0.49
Albanian, Danish	English	LSTM	0.53	0.57	0.50	0.54	0.56
Albanian, Danish	Turkish	BERT	0.52	0.56	0.52	0.53	0.55
Albanian, Danish	Turkish	Gaussian Naive Bayes	0.47	0.48	0.40	0.47	0.48
Albanian, Danish	Turkish	LSTM	0.51	0.51	0.51	0.51	0.51
Albanian, English	Danish	BERT	0.64	0.61	0.62	0.63	0.62
Albanian, English	Danish	Gaussian Naive Bayes	0.51	0.5	0.42	0.51	0.5
Albanian, English	Danish	LSTM	0.59	0.54	0.51	0.58	0.55
Albanian, English	Turkish	BERT	0.56	0.57	0.56	0.56	0.57
Albanian, English	Turkish	Gaussian Naive Bayes	0.48	0.49	0.36	0.48	0.49
Continued in the next page ...							

Continuation of Zero Shot 2-1 Results' Table							
Training Languages	**Testing Language**	**Model**	**Recall**	**Prec.**	**F1 Score**	**F2 Score**	**F0,5 Score**
Albanian, English	Turkish	LSTM	0.53	0.53	0.53	0.53	0.53
Albanian, Turkish	Danish	Gaussian Naive Bayes	0.52	0.51	0.40	0.52	0.51
Albanian, Turkish	Danish	LSTM	0.56	0.53	0.52	0.55	0.54
Albanian, Turkish	English	Gaussian Naive Bayes	0.49	0.49	0.47	0.49	0.49
Albanian, Turkish	English	LSTM	0.55	0.56	0.54	0.55	0.56
Danish, English	Albanian	LSTM	0.51	0.52	0.51	0.51	0.52
Danish, English	Turkish	LSTM	0.52	0.54	0.52	0.52	0.54
Danish, Turkish	Albanian	LSTM	0.53	0.52	0.53	0.53	0.52
Danish, Turkish	English	LSTM	0.54	0.59	0.52	0.55	0.58
English, Danish	Albanian	BERT	0.53	0.59	0.54	0.54	0.58
English, Danish	Albanian	Gaussian Naive Bayes	0.41	0.46	0.35	0.42	0.45
Continued in the next page ...							

Continuation of Zero Shot 2-1 Results' Table							
Training Languages	**Testing Language**	**Model**	**Recall**	**Prec.**	**F1 Score**	**F2 Score**	**F0,5 Score**
English, Danish	Turkish	BERT	0.54	0.57	0.54	0.55	0.56
English, Danish	Turkish	Gaussian Naive Bayes	0.48	0.49	0.30	0.48	0.49
English, Turkish	Albanian	BERT	0.55	0.56	0.55	0.55	0.56
English, Turkish	Albanian	Gaussian Naive Bayes	0.42	0.46	0.35	0.43	0.45
English, Turkish	Albanian	LSTM	0.56	0.54	0.55	0.56	0.54
English, Turkish	Danish	BERT	0.64	0.63	0.63	0.64	0.63
English, Turkish	Danish	Gaussian Naive Bayes	0.46	0.48	0.41	0.46	0.48
English, Turkish	Danish	LSTM	0.57	0.54	0.52	0.56	0.55
Turkish, Albanian	Danish	BERT	0.57	0.67	0.59	0.59	0.65
Turkish, Albanian	English	BERT	0.54	0.67	0.49	0.56	0.64
Turkish, Danish	Albanian	BERT	0.54	0.59	0.55	0.55	0.58

Continued in the next page ...

Continuation of Zero Shot 2-1 Results' Table							
Training Languages	**Testing Language**	**Model**	**Recall**	**Prec.**	**F1 Score**	**F2 Score**	**F0,5 Score**
Turkish, Danish	Albanian	Gaussian Naive Bayes	0.44	0.47	0.36	0.45	0.46
Turkish, Danish	English	BERT	0.61	0.73	0.60	0.63	0.7
Turkish, Danish	English	Gaussian Naive Bayes	0.49	0.49	0.47	0.49	0.49

Table A.6: Complete list of results for the Zero Shot 2-1 Experiment

End of Zero Shot 2-1 Results' Table

A.5.3 Zero shot 3-1

Beginning of Zero Shot 3-1 Results' Table							
Training Languages	**Testing Languages**	**Model**	**Recall**	**Prec.**	**F1 Score**	**F2 Score**	**F0,5 Score**
Albanian, Danish, English	Turkish	Gaussian Naive Bayes	0.47	0.48	0.36	0.47	0.48
Albanian, Danish, English	Turkish	BERT	0.55	0.57	0.55	0.55	0.57
Albanian, Danish, English	Turkish	BERT	0.56	0.58	0.56	0.56	0.58
Albanian, Danish, English	Turkish	LSTM	0.52	0.52	0.52	0.52	0.52
Albanian, Danish, English	Turkish	LSTM	0.52	0.52	0.52	0.52	0.52
Albanian, Danish, Turkish	English	BERT	0.57	0.73	0.54	0.6	0.69
Albanian, Danish, Turkish	English	BERT	0.59	0.72	0.58	0.61	0.69
Continued in the next page ...							

Continuation of Zero Shot 3-1 Results' Table							
Training Languages	**Testing Languages**	**Model**	**Recall**	**Prec.**	**F1 Score**	**F2 Score**	**F0,5 Score**
Albanian, Danish, Turkish	English	Gaussian Naive Bayes	0.5	0.5	0.48	0.5	0.5
Albanian, Danish, Turkish	English	LSTM	0.5	0.49	0.43	0.5	0.49
Albanian, Danish, Turkish	English	LSTM	0.5	0.5	0.43	0.5	0.5
Albanian, Danish, Turkish	English	LSTM	0.5	0.51	0.43	0.5	0.51
Albanian, Danish, Turkish	English	LSTM	0.5	0.51	0.44	0.5	0.51
Albanian, Danish, Turkish	English	LSTM	0.5	0.51	0.44	0.5	0.51
Albanian, English, Turkish	Danish	BERT	0.63	0.65	0.64	0.63	0.65
Continued in the next page ...							

Continuation of Zero Shot 3-1 Results' Table							
Training Languages	**Testing Languages**	**Model**	**Recall**	**Prec.**	**F1 Score**	**F2 Score**	**F0,5 Score**
Albanian, English, Turkish	Danish	Gaussian Naive Bayes	0.49	0.49	0.38	0.49	0.49
Albanian, English, Turkish	Danish	LSTM	0.48	0.48	0.48	0.48	0.48
Albanian, English, Turkish	Danish	LSTM	0.49	0.49	0.49	0.49	0.49
Danish, English, Turkish	Albanian	BERT	0.57	0.58	0.57	0.57	0.58
Danish, English, Turkish	Albanian	Gaussian Naive Bayes	0.41	0.46	0.35	0.42	0.45
English, Danish, Turkish	Albanian	LSTM	0.54	0.52	0.51	0.54	0.52
English, Danish, Turkish	Albanian	LSTM	0.54	0.52	0.51	0.54	0.52

Continued in the next page ...

Continuation of Zero Shot 3-1 Results' Table							
Training Languages	**Testing Languages**	**Model**	**Recall**	**Prec.**	**F1 Score**	**F2 Score**	**F0,5 Score**
English, Danish, Turkish	Albanian	LSTM	0.55	0.53	0.51	0.55	0.53

Table A.7: Complete list of results for the Zero Shot 3-1 Experiment

End of Zero Shot 3-1 Results' Table