Faculty of Engineering, Alexandria University
Computer and Communication Department

Hate speech Detection

Supervisors:

Prof. Dr. Mohamed Abougabal

Prof. Dr. Nagwa El-Makky

Dr. Nancy Diaa El Din

Authors:

Youssef Hany

Omar El Deeb

Zeyad El Abady

Khaled El Gewily

Mostafa Tarek

Karim Yehia

Acknowledgment

We cannot fully express our deepest gratitude towards our mentors and professors Prof. Dr. Mohamed S. Abougabal, Prof. Dr. Nagwa Elmakky, and Dr. Nancy Diaa Eldin who guided us in this project. Without their assistance, we would not have been able to provide any material with such high standards as this.

The least we can do is to give our professors the credit that they deserve. We are extremely thankful for the precious hours that you dedicated to helping us so that we can provide something that would be worthy enough of your standards despite the current situation with covid-19 that is happening around the world. Also, for all your valuable advice and comments that guided us on the right track.

Abstract

In the past few years, an incredible surge of usage of social media platforms has occurred. There is no denying that social media became a constant part of our everyday routine. Chatting with a friend or a colleague, surfing through Facebook, Twitter, or other social platforms, watching videos on YouTube has now become a stable part of our lives.

Social media was originally created to link people all over the world and close the gap of unfamiliarity between them providing any scientific materials on these platforms. Unfortunately, a dangerous side of social media emerged which is offensive or hate speech, in which misconduct and verbal abuse is performed by people practicing inappropriate behavior.

To pursue the healthy environment that was sought from the beginning, there is a need for automated detection of hate speech. This demand particularly arises when the content is written in complex languages (e.g. Arabic). Arabic text is known for its challenges and complexity. In this project, different techniques are proposed to detect hate speech in Arabic language Twitter posts. Hate speech contains insults or threats targeting specific groups based on their nationality, ethnicity, race, gender, political or sport affiliation, religious belief, or other common characteristics.

Table of Contents

List of Figures

List of Tables

Acronyms Table

Abbreviation	Meaning	Page
API	Application Programming Interface	23
BERT	Bidirectional Encoder Representations from Transformers	25
BGRU	Bi-directional Gated Recurrent	29
BLSTM	Bi-directional Long Short-Term Memory	29
BPTT	Backpropagation Through Time	17
CBOW	Continuous Bag of Words	24
CNN	Convolutional Neural Network	16
CPC	Category Pivoted Categorization	15
DNN	Deep Neural Network	28
DPC	Document Pivoted Categorization	15
EDA	Easy Data Augmentation	23
FNN	Feed-forward Neural Network	16
GBT	Gradient-Boosted Trees	28
GRU	Gated Recurrent Unit	17
ICWSM	International AAAI Conference on Web and Social Media	22
IDF	Inverse Data Frequency	26
LR	Label ranking	14
LR	Logistic Regression	28
LSTM	Long Short-term Memory	17
LTC	Liquid Time-Constant	30
MLC	Multi-label classification	14
MLM	Masked Language Model	24
MTL	Multi-task Learning	34
NLM	Neural Language Model	25
NLP	Natural Language Processing	16
OLID	Offensive Language Identification Dataset	22
*RF	Random Forest	28
RNN	Recurrent Neural Network	16
SSWE	Sentiment Specific Word Embedding	27
SVM	Support Vector Machine	27
TF	Term Frequency	26

Preface

This report is written as a part of our graduation project in the Computer and Communication Systems Program at the University of Alexandria, Faculty of Engineering.

As it is fairly a new area, hate speech detection became an important field almost instantly. By taking that into consideration, we decided to choose this area of work to be our project believing that we can try to improve the work that has been done already in other papers and try to include our own perspective. Without a doubt, it has been remarkably hard trying to improve other people's work, but at the same time, it was surprisingly thrilling and beneficial.

Disclaimer

Due to the nature of the project, some examples contain highly offensive language and hate speech. They don’t reflect the views of the authors in any way, and the point of the project is to help fight such speech.

Chapter 1

Introduction

‘’ The first step towards getting somewhere is to decide you’re not going to stay where you are ‘’ -J.P. Morgan

1.1 General

Information is spreading faster than ever nowadays because of social media which in turn allowed people who abuse it to spread fake news propaganda and hate speech as observed in figure 1.1.

Figure 1.1 hate speech growth over time across English language

The definition of hate speech from the Oxford Dictionary is speech or writing that attacks or threatens a particular group of people, especially on the basis of race, religion, or sexual orientation.

The genocide of Rohingya community, the anti-Muslim mob violence in Sri Lanka, and the Pittsburg shooting are all good examples of how hate speech crimes have developed over the years and how it has been increasing and causing issues in different countries.

The public expressions of hate speech and the frequent exposure to it have proved to desensitize the individual and affect the devaluation of minority members, the exclusion of minorities from society, and the discriminatory distribution of public resources.

Since English is the most widely used language all over the world, the majority of datasets and work in the field of detecting hate speech was performed upon it, while hate speech in other languages is not detected properly which causes problems for widely used social media platforms like Facebook which only detect it in (English,Spanish and Mandarin).

1.2 Motivation

Detecting hate speech for Arabic language is very challenging. This is mainly due to the very rich and complex nature of Arabic language. Moreover, there are many arabic dialects that differ from Modern Standard Arabic in lexical selection, morphology, and syntactic structures.

The major focus in hate speech detection and various proposed approaches were mainly for the English language. The fact that the Arabic language is the 4th most spoken language on the internet in 2020, as represented in figure 1.2, and the fact that it is our mother language, encouraged the project team to study hate speech detection for Arabic and to try to make improvements over the few related works in this area.

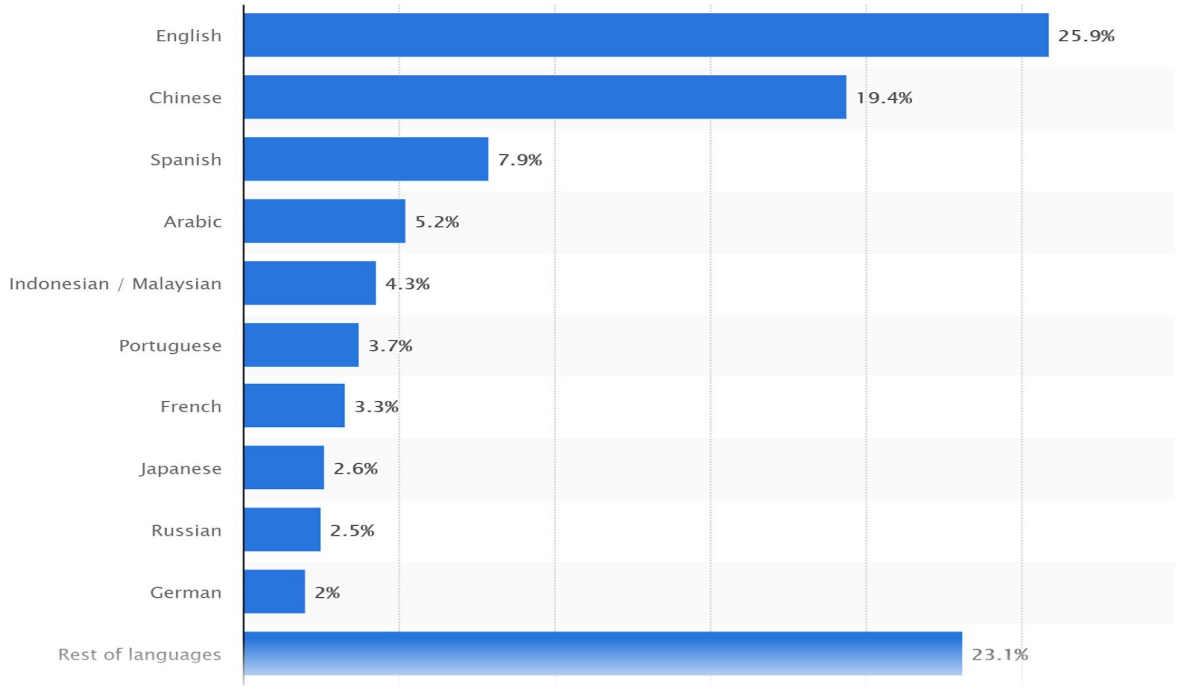


Figure 1.2 top 10 spoken languages on social media in 2020

1.3 Scope of Work

Online social media has developed rapidly in the last few years. With this growth the presence of hate speech and offensive language has grown too, making the manual methods to detect hate speech and offensive language very hard. A lot of researches approached techniques to detect hate speech and offensive language with high accuracy for the English language. Although Arabic language is the fourth most common language used on the internet and detecting hate speech in Arabic content is still nascent.

In this work, different methods are proposed to detect hate speech for Arabic Tweets. Working on Arabic language is challenging. Moreover, tweets are represented in informal Arabic and dialectal Arabic, which is a greater challenge on it's own as the irregularities increase dramatically.

1.4 Report Organization

The current report is organized into 5 chapters. The necessary overview of the problem is introduced in chapter 1 and related works are presented in chapter 2.

Chapter 2

Background and Related Work

“ The only true wisdom is in knowing you know nothing.” -Socrates

2.1 Introduction

A general introduction along with the project motivations and scope of work are presented in the previous chapter. In this chapter, the necessary background about hate speech detection as well as a survey of related work are introduced.

In section 2.2 the difference between hate speech and offensive speech will be discussed. In section 2.3 a few applications of hate speech detection are detailed. section 2.4 outlines the essence of our problem, which is an automated text categorization problem. In section 2.5 some general definitions of machine learning and deep learning are shown while also presenting two different techniques in the context of our problem. Ensemble classifiers are introduced in a very simple way in section 2.6. In section 2.7, hate speech and offensive language detection datasets are presented. In section 2.8, minority class augmentation techniques are explained. In section 2.9 different techniques of word representation are exhibited.Section 2.10 presents a survey conducted for some of the previous works. Section 2.11 outlines the need to extend the related works. In section 2.12 the scope of work is presented. Finally, the topics left for future work are listed in section 2.13.

2.2 Hate speech vs offensive speech

In this section, the differences between hate speech and offensive speech will be discussed. As the presence of social media in our lives increases, people get the freedom to express their feelings and thoughts without supervision. This allows the use of offensive and toxic language, either towards a person or a group of people with certain beliefs.

2.2.1 Offensive speech definition [1]

In general, offensive language is the type of language that proposes or allots negative effects when criticizing someone without involving his ethnicity, religion, sex, and many other things that can be considered as hate speech. The structure of offensive language can contain insults such as curse/profanity words, normal insults such as comparing someone to an animal (donkey, dog, etc.), negative sarcastic comments, for example (@USER you look ugly as s**t) or even threats that can be directed towards an individual or a group.

2.2.2 Hate speech definition [2]

Hate speech is a type of offensive language but has more harm to people receiving it than ordinary offensive speech. The structure of a hate speech sentence contains negative criticism but towards groups or individuals, targeting something common between them like their sex, ethnicity, religion, nationality, or anything that defines a person or his beliefs.

In figure 2.1 we examine different types of hate speech content removed from platforms that have signed the EU Code of Conduct (January 2018).

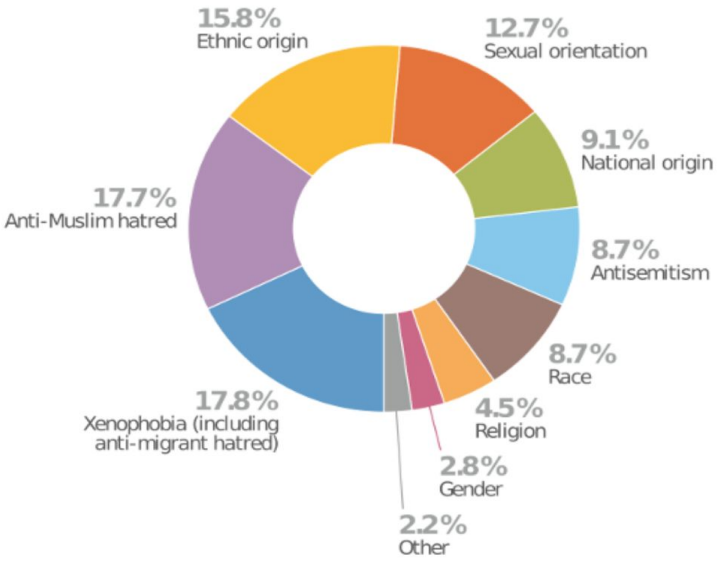


Figure 2.1 Different types of hate speech

2.3 Applications of Offensive/ Hate Speech Detection

Offensive/Hate speech detection is useful in different environments where there is unmonitored interaction using words; such as comments and tweets.

The most common platforms where offensive/hate speech detection is used are Facebook, Twitter, Instagram and Youtube in addition to all social media platforms, where the tweets and comments in posts, photos, and videos might contain offensive language, so, Offensive/Hate speech detection algorithms can be used to detect and remove those comments/tweets.

Offensive/Hate speech is also frequently present in online gaming platforms such as Playstation Network and Xbox Live where players can send messages to each other, and these messages are likely to include offensive language or hate speech due to a portion of players tending to have a toxic behavior when losing.

Any other website or application that wants to monitor the interaction between users may also need to make use of offensive speech detection. Some applications like fitness applications may have a social club where members can socialize and interact through the application, so comments and posts may need to be filtered through detecting offensive language. Since offensive/ hate speech detection is a classification problem, automated text categorization is discussed in the next section.

2.4 Automated text categorization

In this section, the definition and the main aspects of text categorization are presented.

2.4.1 Definition of Text Categorization

Text classification (a.k.a. text categorization or text tagging) is the process of assigning a set of predefined categories to open-ended text. This classification process can be used to organize, structure, and categorize mostly any text – from documents, medical studies and files all over the web. For instance, firms can organize support tickets according to urgency, new articles can be organized according to topics, brand mentions can be organized by sentiment, applications can organize chat conversations according to language, and so on.

2.4.2 Single-label vs Multi-label Text Categorization [3]

Classification is one of the most important topics in the supervised learning field. Although the most widely used classification deals with single-label datasets, where every example is assigned to only one category, the multi-label datasets are becoming more and more popular due to their increasing applicability to real problems. Multi-label datasets are used when the examples can be assigned to more than one category (different labels).

The two main tasks when learning from multi-label data are multi-label classification (MLC) that returns a subset of labels to be associated with a given example (it can be considered as a bipartition of the label set considering relevant and irrelevant elements)

and label ranking (LR) that returns an ordering of the labels according to their relation with the example.

2.4.3 Category Pivoted vs Document Pivoted Text Categorization [4]

A Text Classifier can be used in two ways. The first one is Document Pivoted Categorization (DPC) and the second one is Category Pivoted Categorization (CPC). The primary difference between both is that the document is given in the former while the category is given in the latter.

DPC is when given a document D, we need to find all categories under which it should be filed. This means that the document is searched under all categories and the required corresponding category will be found which contains the given document.
CPC is when given a category C, we need to find all documents under which it should be filed. This means that the category is searched under all documents and the required corresponding document will be found which contains the given category.

However, DPC is used more often than CPC, as the former situation is more common.

2.5 Machine learning vs Deep learning

The main differences between machine learning and deep learning are discussed in the following subsections.

2.5.1 Machine learning

Machine learning is a field that focuses on the learning aspect of artificial intelligence by developing algorithms that best represent a set of data. In contrast to classical programming, in which an algorithm can be explicitly coded using known features, ML uses subsets of data to generate an algorithm that may use novel or different combinations of features and weights that can be derived from first principles**[5]**.
There are two broad categories of machine learning problems, supervised and unsupervised learning. Our problem is a supervised one since classes of tweets will be provided in our dataset as we will discuss in later sections.
Machine learning literally means “machine teaching.” We know what the machine needs to learn, so our task is to create a learning framework and provide properly-formatted, relevant, clean data for the machine to learn from. A machine learning model is the sum of the learning that has been acquired from its training data. The model changes as more learning is acquired.

2.5.2 Deep learning

Deep Learning is a machine learning technique that constructs artificial neural networks to mimic the structure and function of the human brain. In practice, deep learning uses a

large number of hidden layers and refers to applying deep neural networks to massive amounts of data to learn a procedure aimed at handling a task. The task can range from simple classification to complex reasoning.

Over the last few years, deep learning has been considered to be one of the best methods and has become very common for its ability to handle a huge amount of data. The interest in having deeper hidden layers has recently begun to surpass classical methods performance in different fields; especially in computer vision and natural language processing. The most popular deep neural networks are the Feed Forward Neural Networks (FNNs), the Convolutional Neural Networks (CNNs) and the Recurrent Neural Networks(RNNs) **[6]**.

Feed-forward Neural Network

A feed-forward neural network is an artificial neural network where connections between the nodes do not form a cycle. This is the main difference from its descendant the recurrent neural networks. The feed-forward neural network was the first and most simplistic type of artificial neural network devised. In this network, the information moves in only one direction, it moves forward from the input nodes, through the hidden nodes (if any) and to the output nodes. There are no cycles or loops in the network.

Convolutional Neural Networks

Convolutional neural networks (CNNs), whose architecture is inspired by the human visual cortex, are a subclass of feed-forward neural networks.CNNs are named after the underlying mathematical operation “convolution” which yields a measure of the interoperability of its input functions. Convolutional neural networks are usually employed in situations where data needs to be represented with a 2 dimensional (2D) or 3 dimensional (3D) data map. In the data map representation, the proximity of data points usually corresponds to their information correlation.

For using CNNs for NLP, the inputs are sentences represented as matrices. Each row of the matrix is associated with a language element such as a word or a character. Most CNN architectures learn word or sentence representations in their training phase. Different CNN architectures were used in various classification tasks such as Sentiment Analysis and Topic Categorization.

Recurrent Neural Network

Recurrent Neural Networks (RNNs) have shown great promise in natural language processing tasks. Unlike feedforward neural networks, RNNs can handle a variable-length sequence input by having a recurrent hidden state whose activation at

each time is dependent on that of the previous time. Figure 2.2 shows what a typical RNN looks like **[7]**.

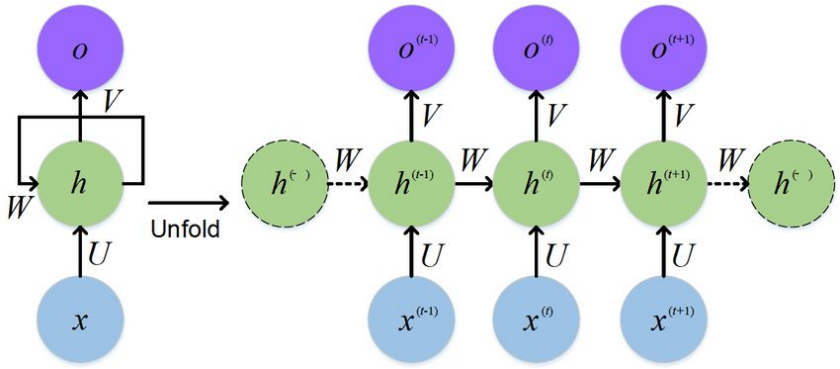


Figure 2.2 Recurrent neural network

Similar to a traditional neural network, a twisted backpropagation algorithm called Backpropagation Through Time (BPTT) is used to train an RNN. Unfortunately, it is difficult to train RNN to capture long-term dependencies because the gradients tend to either vanish or explode . In 1997 long short-term memory (LSTM) unit has been proposed, and in 2014 gated recurrent unit (GRU) has been proposed to deal with the problem effectively.

LSTM

The Long Short-Term Memory (LSTM) was first proposed by Hochreiter and Schmidhuber in 1997 allow RNNS to learn long-term dependencies. All RNNs have feedback loops in the recurrent layer. This lets them maintain information in 'memory' over time. But, it can be difficult to train standard RNNs to solve problems that require learning long-term temporal dependencies. This is because the gradient of the loss function decays exponentially with time (called the vanishing gradient problem). LSTM networks are a type of RNN that use special units in addition to standard units. LSTM units include a 'memory cell' that can maintain information in memory for long periods of time. A set of gates is used to control when information enters the memory, when it's output, and when it's forgotten. This architecture, displayed in figure 2.3, lets LSTM networks learn longer-term dependencies.

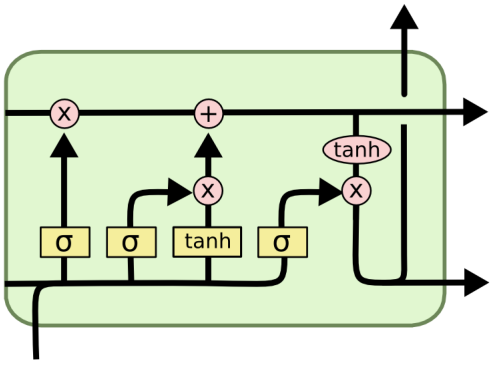


Figure 2.3 LSTM Unit

GRU

A gated recurrent unit (GRU) was initially proposed by Cho et al.in 2014 to make each recurrent unit adaptively capture dependencies of different time scales**[7]**.
GRU is related to LSTM as both are utilizing different ways of gating information to prevent vanishing gradient problem. Unlike LSTM ,GRU has only two gates which are reset and update gates.The reset gate determines how to combine new inputs with the previous memory, and the update gate defines how much of the previous memory remains. The following is an illustration of the GRU unit figure 2.4.

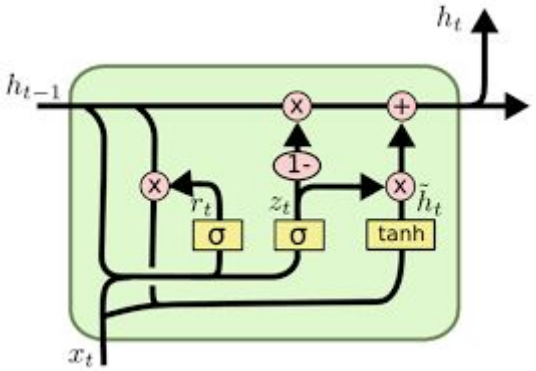


Figure 2.4 GRU unit

In general, The main difference between CNN and RNN is the ability to process temporal information or data that comes in sequences, such as a sentence for example. Moreover, convolutional neural networks and recurrent neural networks are used for completely different purposes, and there are differences in the structures of the neural networks themselves to fit those different use cases. CNNs employ filters within convolutional layers to transform data, while RNNs reuse activation functions from other data points in the sequence to generate the next output in a series.

2.6 Ensemble classifiers [8] [9] [10]

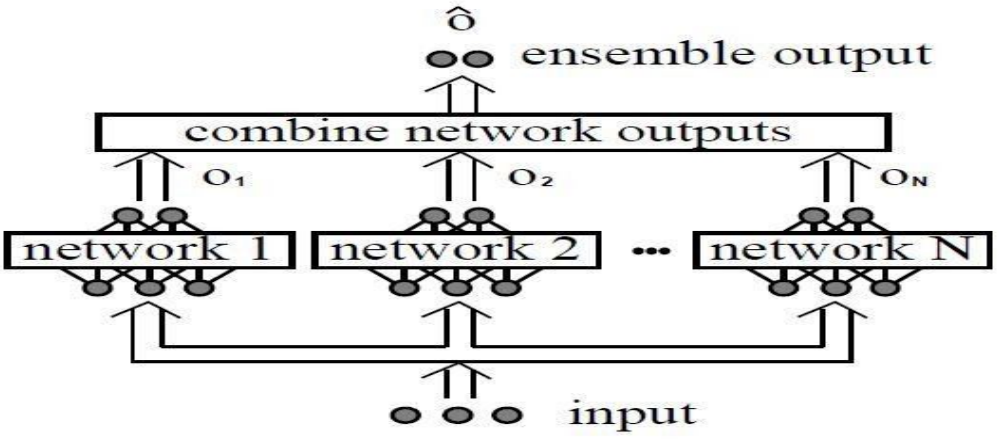
Ensemble classifiers consist of different individual trained classifiers; these classifiers can be based on different learning algorithms (such as neural networks, decision trees, or regression). The idea behind the ensemble is mixing different classifiers to make a new classifier that has an error rate less than the composing classifiers' error rate. For example, if there are three classifiers each with an error rate of 30%, then the probability that one of them misclassified one input is 0.3 but the probability that two of them misclassify the same input is less than 0.3.

It was proved that if the component classifiers have an error rate of for example 50%, then the error rate of the ensemble classifier is zero if the number of the components goes to infinity however this assumption can't hold in real life. Later it was proved that the error rate can be divided into two terms: average error of component classifiers and a term measures the disagreements between individual classifiers. A good ensemble is made of accurate classifiers that each classifier makes errors of different input space because if all classifiers misclassify the same input then the main idea of the ensemble is not standing.

The ideal ensemble is made of classifiers with high accuracy and these classifiers disagree as much as possible. A higher weight could be given to some component's classifiers outputs than other classifiers based on each one's accuracy. For example figure 2.4 shows an ensemble that uses neural networks as a basic classifier (could be different algorithms). The output of classifiers (classifier one through N) is mixed to produce one output. One scheme found effective is to average the output of different networks. Different neural network architecture, Initial weights and hyperparameters lead to different network results.

Figure 2.5 ensemble classifier of neural networks

For the most of the time, an Ensemble of classifiers has a higher accuracy than a single classifier.The most three common ensemble algorithms are bagging, boosting and stacking. Bagging is based on manipulating the dataset, each component classifier is



trained on randomly drawn with replacement dataset of a size usually the same size as the original dataset. All the training examples come from the original dataset so some examples can appear more than one time in component classifiers' training data and some examples may not appear. Bagging can easily be implemented in a parallel manner by training each classifier through different computational units.

The idea around boosting is to make a series of classifiers each with a training set that changes based on the previous classifier's performance. Adaboost is an example of boosting algorithms where all training examples of the first classifier have the same weight. After the first classifier training, the misclassified examples are given higher weight and correctly classified examples weights are decreased. The second classifier is trained on a dataset with updated weights so it gives a higher priority to misclassified examples from the previous classifier. This procedure is repeated until a specific number of iterations are completed or specific accuracy is achieved.
Stacking is concerned with mixing component classifiers with different learning algorithms (heterogeneous models) unlike bagging and boosting which are based on data manipulating. Stacking consists of two phases, base-level classifiers and a meta-level classifier. A meta-dataset is created that contains the same number of instances found in the original dataset. However, instead using the original input attributes, it uses the base classifiers output as input features. The predicted attribute remains the same as in the original dataset. A test instance is first predicted by each of the individual base classifiers. These predictions are then fed into the meta-model which provides the final output.

2.7 Hate Speech and Offensive Language Detection Datasets [11]

Since hate speech detection is a new topic in the natural language processing field, there were no large datasets to be used for training and testing classifiers. In the beginning, the majority of work and studies in this area forced those working on it to collect the data they need themselves and create their own datasets, and annotating them to be able to use them for their intended work. There were prominent or go-to platforms that people collected the data from including Twitter, Facebook and yahoo message boards.

During the course of the past few years, several datasets emerged that were collected and publicly published to be used, here are a few of them. The first dataset published in this field of work was BURNAP Dataset, the dataset was collected by **[12]** in English and had four different characteristics including religion and sexual orientation. Then came the WASEEM dataset which was published by **[13]**. The dataset consisted of 16k tweets and was categorized into three different classes which are racism, sexism, or neither. Later on, this dataset was extended again through adding additional 4k tweets to it by the same publisher.

Similar to the datasets before, the DAVIDSON Dataset was published by **[14]** and labeled 24k tweets with either hate speech, offensive, or neither. The publishers said they collected the dataset using a lexicon from HateBAse and used a crowdsourcing platform called Figure-Eight for annotating the tweets. The Founta Dataset collected by

[15] boasted a number of 80k tweets which was at this time larger than any previous dataset collected in this field. The dataset was made publicly available using TweetIDS. . The authors use a boosted random sampling technique to generate the final dataset. As part of ICWSM challenge, they released a new version of this dataset containing 100k tweets in text format.

There were more targeted datasets to a specific group of people for example the WARNER Dataset which was collected by **[16]** using data from Yahoo News Group and URLs from the American Jewish Society. It was classified into 7 categories including anti-muslim, anti-sematic and anti-black. There was also the ZHANG dataset **[17]**created using Muslim and refugee-specific words from Twitter. The dataset was initially publicly available but not anymore.

Not all datasets used the traditional platforms for collecting the data, for example, the QIAN Dataset **[18]** used Reddit and Gab including intervention responses written by humans. For providing context for more accurate annotations they leave the conversational threads intact which means that each entry in this dataset is a conversation of several indexed comments.

Unlike the majority of the datasets which were only in English, the HATEVAL Dataset **[19]** that was used for detecting hate speech against women and immigrants contains English and Spanish tweets labelled into hateful or not hateful. In other languages, hate speech detection research has also progressed.

Regarding offensive language detection , OLID **[20]** is one of the most widely used datasets in this field. In the OLID dataset, the authors use a hierarchical annotation schema split into three levels to distinguish between whether the language is offensive or not , its type , and its target. OLID was used in the OffensEval: Identifying and Categorizing Offensive Language in Social Media (SemEval 2019 - Task 6) shared task.

Offensive language and hate speech detection Arabic datasets are very scarce. Recently,the authors of [21] built the largest Arabic offensive language dataset to date that includes special tags for vulgar language. This dataset was used in SemEval 2020. More recently, the authors of [21] annotated the hate speech records in this dataset to be used for both offensive language and hate speech detection. Subsequently, this dataset was used in OSACT4 shared task on offensive language detection. OSACT4 Shared Task on Offensive Language Detection has 2 tasks. Task A is to detect offensive tweets containing explicit or implicit insults or attacks against other people, or inappropriate language. Task B is to detect hate speech. If a tweet has insults or threats targeting a group based on their nationality, ethnicity, gender, political or sport affiliation,

religious belief, or other common characteristics, this is considered as Hate Speech. More details, about how this dataset was collected, are given in chapter 3.

2.8 Imbalanced Dataset problem

This section outlines the problem of an imbalanced dataset declaring its definition. Then, the different techniques for minority class augmentation are discussed.

2.8.1 Class Imbalance meaning[22]

Most real-world classification problems display some level of class imbalance which means there is one class in the dataset which largely outnumbers the other classes, for example, suppose you have two classes, class A and class B, class A is 90% of the dataset and class B is the other 10%. But we are most interested in identifying instances of class B.

That is the exact situation of hate speech and offensive language detection, most of the dataset is clean. Models that train on a balanced dataset have better results than models that do not, so the dataset needs to be more balanced.

Balancing the dataset is achieved by augmentation of minority (rare) classes.

2.8.2 Data augmentation of minority (rare) classes [23]

Data augmentation of minority (rare) classes is increasing the size of rare classes in the dataset through generating new variants by adding slightly modified copies of already existing data.

There are many methods for data augmentation as discussed below.

Back translation

In this method, the text data is translated into some language and then translated back to the original language. This can help to generate textual data with different words while preserving the context of the text data. Language translation APIs like google translate, Bing, Yandex are used to perform the translation.

Easy Data Augmentation

Easy data augmentation uses traditional and very simple data augmentation methods. EDA consists of four simple operations that do a surprisingly good job of preventing overfitting and helping train more robust models.

1-Synonym Replacement
2-Random Insertion
3-Random Swap
4-Random Deletion

NLP Albumentation

1-Shuffle Sentences Transform, in this transformation, if the given text sample contains multiple sentences these sentences are shuffled to create a new sample.

2-Exclude duplicate transform, in this transformation, if the given text sample contains multiple sentences with duplicate sentences, these duplicate sentences are removed to create a new sample.

2.9 Word representation

In this section, the explanation of the two main techniques of word representation is discussed.

2.9.1 Word embeddings

A word embedding is considered to be a learned representation for text where words that have identical meanings have close representation. Word embeddings are classified as a list of tools that are used to represent words in a vector form with real values in a predefined vector space. Every word has a corresponding vector mapped to it and the vector values are learned in a way that resembles a neural network, therefore, this technique has commonly appeared with deep learning **[24]**.

To achieve this objective, the best approach is employing a densely distributed representation for every word, having a real-valued vector mapped to its corresponding word, often tens many dimensions.

Some models for English word embeddings are Word2vec, Glove, FastText, BERT. There are also other models for multilingual word embeddings like mBERT and MUSE .

Word2VEC is a word embedding model built using neural networks that uses a large corpus of the dataset to learn word associations. This model learns the embeddings by predicting the current word using the context as an input **[26]**.

There are two different learning models for Word2Vec namely; Skip Gram and Continuous Bag of Words (CBOW). The continuous bag of words model is the most widely used technique for sentiment analysis, it uses an enormous lexicon that may contain duplication of words and some repetitions. To build this lexicon, people are needed to provide positive and negative word lists using the sentiment of the word that can be deduced by manual observation.**[26]**

GLOVE denotes Global Vectors due to the fact that global corpus statistics are captured directly by the model **[27]**.

FastText is an approach based on the skip gram model, where each word is represented as a bag of character n-grams. A vector representation is associated with each character n-gram; words being represented as the sum of these representations. This method is fast, allowing to train models on large corpora quickly and allows us to compute word representations for words that did not appear in the training data **[26]**.

BERT is a multi-layer bidirectional transformer encoder trained on the English Wikipedia and the Book Corpus containing 2,500M and 800M tokens, respectively. Bert uses a masked language model (MLM) pre-training objective. The masked language model randomly masks some of the tokens from the input, and the objective is to predict the original vocabulary identifier of the masked word based only on its context. It also has the ability to fuse the left and the right context, unlike left to right language model pre-training, which enables us to pre-train a deep bidirectional Transformer **[28]**.

In addition, BERT is capable of encoding sentence-level properties even with single-word embeddings. Contextual BERT is a Neural Language Model (NLM) that can encode a wide range of linguistic information in a hierarchical manner, also it can give support to the process of extraction of dependency parse trees using its different layers, which can not be done using contextual-independent Neural Language model like Word2VEC **[29]**.

mBERT is the multilingual version of BERT which is trained on Wikipedia consisting of 104 languages. mBERT is simply trained on a multilingual corpus with no language IDs, but it encodes language identities. mBERT is used to train hate speech detection models in different languages **[29]**.

Multilingual word embeddings are used to train models with a multilingual setting such as Muse embeddings. Muse denotes Multilingual Unsupervised and Supervised Embeddings. It constructs a bilingual dictionary between two languages without the help of any parallel corpora, using the method of aligning monolingual word embedding spaces in an unsupervised manner.

For Arabic word embeddings, there are Aravec, Mazajak and AraBERT.

Aravec is a distributed word representation open source project whose objective is to give strong word embedding models to the Arabic NLP society for different NLP tasks. The total number of tokens used to build the models amounts to more than 3,300,000,000. The models were built carefully using multiple different Arabic text resources to provide wide domain coverage. Specifically, the models were built using web pages collected from World Wide Web, text harvested from social

platforms and text obtained from encyclopedia entries. AraVec provides six different word embedding models, where each text domain (Tweets, WWW and Wikipedia) has two different models; one built using CBOW technique and the other using the Skip-Gram technique **[30].**

In Mazajak, word embeddings were created using the word2vec . The skip-gram architecture was used. The embeddings were built using a corpus of 250M unique Arabic tweets; this makes it a larger Arabic word embeddings set when compared to the available AraVec built using a corpus of 67M tweets.t. The word embeddings were built using word2vec utilising the skip-gram architecture **[31].**

AraBert is a model created solely for word embeddings of Arabic language. It achieves state-of-the-art performances in many NLP tasks compared to other models that use multilingual datasets.The final size of the pre-training dataset of AraBERT, after removing duplicate sentences, is 70 million sentences, corresponding to ~24GB of text. This dataset covers news from different media in different Arab regions, and therefore can be representative of a wide range of topics discussed in the Arab world. **[32]**.

2.9.2 Statistical Bag-of-Words Representations (e.g., TF-IDF) [33]

Another strategy of representing words is to score the relative importance of words using Term Frequency-Inverse Document Frequency (TF-IDF). It is commonly used with classical classifiers.

Term Frequency (TF)

It is the number of times a word appears in a document divided by the total number of words in the document. Every document has its own term frequency.

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{i,j}} \qquad (2.1)$$

Tfi,j = frequency of word i in document j.

ni,j = the number of occurence of word i in document j

$\sum_k ni,j = total\ number\ of\ words\ in\ document\ j$

Inverse Data Frequency (IDF)

It is the log of the number of documents divided by the number of documents that contain the word ***w***. Inverse data frequency determines the weight of rare words across all documents in the corpus.

$$idf(w) = log(\frac{N}{df_t}) \quad (2.2)$$

N = total number of documents in corpus
Dft = number of documents where term t appears.

Lastly, the TF-IDF is simply the TF multiplied by the IDF.

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right) \quad (2.3)$$

In the upcoming section, some previous related works are going to be illustrated with some details.

2.10 Related work

The summary of surveying several papers related to offensive language and hate speech detection for English and Arabic languages is presented in this section.

2.10.1 Offensive and Hate speech detection for English language

Davidson, et al. 2017 [34]
The objective of this paper was the separation of hate speech from other instances of offensive speech, for this reason, the authors chose to compare the performance of different classical machine learning classifiers that incorporate distributional TF-IDF features, part-of-speech tags, and other linguistic features. Logistic Regression and Linear SVM tended to perform significantly better than other classifiers, as a final model logistic regression with L2 regularization was chosen.

Linguistic features also helped to identify hate speech by distinguishing between different usages of the term, but the model still suffers from some subtleties, such as when offensive terms are used positively.

As future work, the authors wanted to look more closely at different social contexts in which hate speech is used and also to examine the individual characteristics of people using this term, their motivations, and social structures they are embedded in.

S. Agrawal and A. Awekar 2018 [35]

The authors conducted this paper in order to detect cyberbullying on social media across datasets from different social media platforms (Twitter, FormSpring and Wikipedia), experimenting with the usage of traditional machine learning models against deep neural network models using a variety of representations methods for words as bag of character n-gram, bag of word unigram, GloVe embeddings and Sentiment Specific Word Embedding (SSWE).

It was found that deep-learning based models outperformed traditional Machine Learning models for the task, the vocabulary of words and terms used for cyberbullying differs across different platforms, cyberbullying posts on social media are few so most datasets were imbalanced and also the authors used DNN based models coupled with transfer learning which achieved higher scores.

As for future work, the improvement of models using extra data as the profile and social graph of users was suggested, accompanied by providing info about the severity of bullying in upcoming datasets.

Younghun Lee et al. 2018 [36]

Previously the datasets used for abusive language detection were relatively small ranging from 10k to 35k in size, this quantity was insufficient to train the parameters of neural network models, due to this reason most studies addressed the problem using traditional machine learning classifiers. Most recently Founta et al. (2018) **[37]** introduced Hate and Abusive Speech on Twitter, a dataset containing 100K tweets that has great potential for deep learning models.

In this paper, the authors investigate the efficacy of the most frequent machine learning model as well as recent neural network classifiers. Additionally, the effect of character-level feature representation was discussed, together with the possibility to improve more and more using ensemble classifiers.

concerning results, neural models were more accurate than classical models except for the logistic regression model which had an equal F1 score as the highest CNN classifier, character-level representation of words improved the performance of SVM and RF classifiers but it significantly decreases the accuracy of CNN and RNN model, as for ensemble classifiers GBT and RF classifiers followed the LR classifiers in context of traditional machine learning classifiers.

As for future work, the authors suggest focusing more on the usage of ensemble classifiers to address this problem as more improvements would be expected.

M. Ibrahim et al. 2018 [38]

The authors used data from Wikipedia talk page edits to train multi-label classifiers to detect different types of toxicity. The dataset has several toxicity classes toxic, severe toxic, obscene, threat, insult and identity hate. The dataset suffers from being imbalanced. More than 85% of the data is not toxic and toxicity classes have skewed distribution, where 3 of the 6 classes were represented by less than 7%, so the authors used data augmentation to overcome this problem. Unique Words Augmentation, Random mask and Synonyms Replacement were used as data augmentation techniques. The prediction of the toxic class is approached over two steps: the first step is to classify the input whether toxic or not, then the second step if the input is toxic is to classify it into one of the toxicity classes. That was done by training different deep learning models: convolutional neural networks (CNN), bidirectional long-short term memory (LSTM), and bidirectional gated unit (GRU) and ensemble model that use all of them.

Zampieri et al. 2019 [39]

This paper presents the results of SemEval-2019 Task 6 on identifying and classifying offensive language on social media. The task was based on the OLID dataset we presented in previous sections and it featured three subtasks.

Subtask A, which is offensive language identification, had a top result by using BERT-base-uncased classifier with default parameters but with a max sentence length of 64 and trained on 2 epochs, also an ensemble of CNN and BLSTM + BGRU had significant results on the task.

Subtask B, which is automatic categorization of offense types, had ensemble classifiers used among 5 of the top 10 teams, ensembles of deep learning and non-neural machine learning models were used by those teams.

Subtask C, which is offense target identification, had BERT used by the 1st ranked team together with pre-trained word embeddings based on GloVe, and an ensemble of deep learning models such as OpenAI Finetune, LSTM, Transformer, and non-neural machine learning models such as SVM and Random Forest used by the team ranked in 2nd place.

In future works, the authors wanted to increase the size of the OLID dataset, address the problem of small class size and imbalance dataset, and also the need to expand the task to other languages rather than English.

The following table summarizes the previously mentioned papers for English language offensive and hate speech detection.

Paper	Platform	Goals	Best model	Year
[34]	Twitter	Answering the questions listed below. 1-will linguistic features help to detect hate speech? 2-Compare the performance of different classical machine learning classifiers.	Logistic regression .	2017
[35]	Twitter Formspring Wikipedia	1-detect cyberbullying across datasets collected from different platforms. 2-use a variety of word representation techniques, to know which one would perform better. 3-Experiment the usage of machine learning against deep learning. 4-try transfer learning techniques.	BLSTM with attention and feature level transfer learning.	2018
[36]	Twitter	1-improve performance using ensemble classifiers. 2-compare performance of classical machine learning classifiers and neural network classifiers when trained over a large dataset (100k tweets). 3-Discuss the effect	RNN-LTC model and CNN.	2018

		of character-level representations .		
[38]	Wikipedia	1-Reduce dataset imbalance by developing a data augmentation technique, and how will it improve performance and accuracy. 2-Trying different deep learning models such as CNN, LSTM and GRU. 3-Try an ensemble model that uses all deep learning models used in the project.	ensemble model	2018
[39]	Twitter	This paper discusses the results of SemEval 2019 task 6, it provides a summary of different techniques used in 3 different tasks: offensive language identification, automatic categorization of offense types and offense target identification.	Ensemble classifiers and BERT model .	2019

Table 2.1 Hate speech and Offensive language detection for English language

2.10.2 Hate speech detection for Arabic language

Mubarak et al. 2020 [21]

In this paper, the authors introduced the largest Arabic offensive language dataset and described the specific methodology on which it was built, and experiment the usage of SVM classifiers with different word representations as lexical features, Pre-trained static embeddings (AraVec, Mazajak and FastText) ,embeddings trained on data (FastText) and deep contextualized embeddings (BERT).

As a result it was found that among all representations used with the SVM classifier, Mazajak yield the best result as it is suspected that it outperformed the BERT setup due to the fact that Mazajak embeddings were trained on in-domain data as opposed to BERT.

For completeness a comparison between SVM and 7 other classical machine learning classifiers were conducted, as all of them were using Mazajak embeddings ,using SVM yield the best results.

As a future work it was planned to explore target specific offensive language where attacks against an entity or a group may employ certain expressions that are only offensive within the context of that target and completely innocuous otherwise. Second, it plans to examine the effectiveness of cross dialectal and cross lingual learning of offensive language.

A. Abuzayed & T. Elsayed 2020 [40]

In this paper, the authors aim to answer two research questions in the context of hate speech detection

1-Is distributed word representation (e.g., Word2Vec embeddings) more effective than standard statistical bag-of-words representation (e.g., Tf-Idf) ?

2-Are neural models more effective than classical machine learning models ?

The results showed that tf-idf representation was more effective than word embeddings representation when used with 7 different classical machine learning models and that the extra trees model gave the highest performance.

Also when training different neural models with the word embedding feature, they outperformed the extra tree model and CNN + LSTM gave the highest accuracy.

Oversampling was tried to address the imbalance of the dataset but it didn’t give a significant effect.

As for future work, using tf-idf with neural models was suggested since it gave a good performance when used with machine learning models, trying transfer learning techniques were also recommended. Since BERT yielded the state of art performance in several natural language processing tasks it was also suggested to be used.

I. Abu-Farha and W. Magdy 2020 [41]

In this paper, the authors explored various approaches to detect hate-speech and offensive language, which include transfer learning, deep learning and multitask learning .As for word representation, Mazajak word embedding were used, and the paper also describes several data preprocessing approaches as letter normalization, elongation removal and cleaning by the removal of different unwanted character.

The results showed that multitask learning had the highest performance, which shows that the extra information learned through learning multiple objectives was effective to improve the performance of the model.

As for future work it was suggested to improve the results using other resources such as lexicons and experimenting more multitask learning settings.

Mubarak et al. 2020 (OSACT4) [42]

In this paper the authors provide an overview of the offensive language detection shared task at the 4th workshop on Open-Source Arabic Corpora and Processing Tools (OSACT4). There were two main subtasks, namely, Subtask A, involving the detection of offensive language, which contains unacceptable or vulgar content in addition to any kind of explicit or implicit insults or attacks against individuals or groups; and

Subtask B, involving the detection of hate speech, which contains insults or threats targeting a group based on their nationality, ethnicity, race, gender, political or sport affiliation, religious belief, or other common characteristics. For the datasets the two subtasks used the SemEval 2020 Task 12 Arabic offensive language dataset, which contains 10000 tweets that were manually labeled. A variety of methods for preprocessing were used such as: character normalization, removal of punctuation, diacritics, repeated letters, and non-Arabic tokens, extensive preprocessing including normalizing emoticons (translating their English description to Arabic), dialectal to MSA conversion, word category identification, removal of dialectal stopwords and hashtag segmentation. For learning methods the models used were SVM, logistic regression, Convolutional Neural Network, Recurrent Neural Network including LSTM biLSTM and GRU and fine tuning of contextual embeddings such as BERT and AraBERT. The highest ranking submissions used an ensemble of different learning methods that combined both traditional machine learning and Deep Learning approaches.

In the following table, the Arabic offensive language and hate speech detection papers mentioned in this subsection are summarized.

Paper	Dataset	Goals	Best model	Year
[21]	Mubarak et al. **[21]**	1-Build the largest Arabic offensive language dataset . 2-Experiment the usage of SVM classifiers with different word representations and to find out which representation would have best results . 3-Comparison between SVM and 7 other classical machine learning classifiers, all of them using Mazajak embeddings .	SVM with Mazajak embeddings.	2020
[40]	Mubarak et al. **[21]**	1-Is distributed word representation more effective than standard statistical bag-of-word representation ? 2-Are neural models more effective than classical machine learning models ? 3-try ensemble classifiers .	CNN + LSTM	2020
[41]	Mubarak et al. **[21]**	1-Experiment the usage of deep learning ,transfer learning and multitask learning .	Multitask learning	2020

[42]	Mubarak et al. **[21]**	Find the best results for subtasks A and B for the shared task at the 4th workshop on Open-Source Arabic Corpora and Processing Tools (OSACT4).	an ensemble of different learning methods that combined both traditional machine learning and Deep Learning approaches.	2020

Table 2.2 Hate Speech and Offensive Language detection for Arabic language

A comparison between the Arabic offensive language/ hate speech detection papers is presented in Table 2.3.

paper	Deep learning	Dataset available	Balanced dataset	Code available	Suitable preprocessing	Word embeddings
[21]	□	✓	□	□	✓	✓
[40]	✓	✓	□	✓	✓	✓
[41]	✓	✓	□	□	✓	✓
[42]	✓	✓	□	-	✓	✓

Table 2.3 Comparison between the Papers Surveyed in Section 2.10.2

2.10.3 Arabic Preprocessing for Papers Surveyed in Section 2.10.2

Arabic preprocessing needs much more efforts than English preprocessing due to the challenges of the Arabic language. Among these challenges; Arabic is highly inflectional and derivational. Inflectional means that each word consists of a root and zero or more affixes (prefix, infix, suffix). Derivational means that all the Arabic words have root verbs of three or four characters. Also, Arabic is characterized by diacritical

marks . The same word with different diacritics can express different meanings. Diacritics are usually omitted which greatly increases ambiguity.

This section describes the preprocessing operations done in the papers surveyed in section 2.10.2

	[41], 2020	[21], 2020	[40], 2020	[42],2020
Letter normalization	✓	✓	✓ (AraVec2.0)	✓
Elongation removal	✓	✓	✓	✓
Remove unk characters	✓	✓	✓	✓
Remove punctuation	✓	✓	✓	✓
Remove URLs	✓	✓	✓	✓
Remove diacritics (tashkeel)	✓	✓	✓	✓
Remove stopwords	□	□	□	□
Tokenizing	-	Farsa arabic NLP toolkit	-	-
attached words separation	□	✓	□	-
Remove repeated letters	✓	✓	✓	✓
Remove kashida	✓	✓	✓	✓

Table 2.4 Arabic preprocessing for Papers Surveyed in Section 2.10.2

2.10.4 Feature Representation for Papers Surveyed in Section 2.10.2

The following table summarizes the feature representation techniques used by the papers surveyed in section 2.10.2.

representation	[42],2020	[40], 2020	[41], 2020	[42], 2020
Lexical Features	□	NileULex	□	✓
fastText embeddings	□	✓	□	✓
AraVec embeddings	□	✓	AraVec2.0	✓
Mazajak embeddings	✓	✓	□	✓
Trained fastText	□	✓	□	✓
Deep contextualized embeddings	□	BERT	□	BERT
Bag-of-words	□	□	**tf-idf**	**tf-idf**

Table 2.5 Feature Representation for the Papers Surveyed in Section 2.10.2

2.11 The Need to Extend the Related Work

From the previous sections, it is clear that the care given to hate speech detection for the Arabic Language is far less than the care given for other languages, especially the English language. This is evident from the many papers that address this topic for English language and the different English datasets introduced for this task. On the contrary, few papers address this topic for Arabic language and very scarce Arabic datasets were introduced for this task. From the surveyed papers, it is clear that there is a need to extend the related work since there is much room for improvement over the current studies. The points listed below summarize the techniques that can be used to extend the related work.

1. Data Augmentation.
   Since Arabic hate speech detection is a fairly new field, data augmentation for minority classes is still to be used as a technique to solve the problem of imbalanced datasets.

2. Word embeddings
   Experimenting with different Arabic word embeddings is necessary for this project as the Arabic language has multiple structures and wordings. Trying the contextual word embedding AraBert which gave state-of-the-art

performance on many NLP tasks is expected to give good performance results.

3. Ensemble Classifiers.
   Using ensemble classifiers will assist the project by providing better results as a machine learning technique.Using an ensemble which combines different classification techniques is expected to improve the performance.

4. Multitask Learning
   Multi-task learning (MTL) is a subfield of machine learning in which multiple tasks are simultaneously learned by a shared model.Such approach offers advantages like improved data efficiency, reduced overfitting through shared representations, and fast learning by leveraging auxiliary information. Previous works related to the Arabic language suggested that using this technique would probably improve performance.

5. Transfer Learning
   Transfer learning is the improvement of learning in a new task through the transfer of knowledge from a related task that has already been learned. While most machine learning algorithms are designed to address single tasks, the development of algorithms that facilitate transfer learning is a topic of ongoing interest in the machine-learning community. This technique was recommended also in many papers in the previous section **[43]**.

2.12 Scope of work

This section is used for stating the topics that will be used to extend the related work in this project. The rest of the topics will be included in the future work section of the chapter.

1. Data Augmentation
2. Word Embeddings
3. Ensemble Classifiers

2.13 Future Work

The following topics that are mentioned in section 2.11 will not be included in this project and can be used for future work

1. Multitask Learning
2. Transfer Learning

References

[1] Da.Costa Abreu, Marjory and A. De Souza . “Automatic offensive language detection from Twitter data using machine learning and feature selection of metadata”. In: IEEE World Congress on Computational Intelligence (IEEE WCCI). IEEE ,March,2020.

[2] Anna Schmidt, Michael Wiegand . “A Survey on Hate Speech Detection using Natural Language Processing”. In: The Fifth International Workshop on Natural Language Processing for Social Media, 2017, pp. 1-10

[3] Carmona-Cejudo, J.M., “Feature extraction for multi-label learning in the domain of email classification,” Computational Intelligence and Data Mining (CIDM),IEEE, April, 2011, pp. 30-36.

[4] A. Faraz , “A Comparison of Text Categorization Methods”, International Journal on Natural Language Computing (IJNLC), vol. 5, no. 1, February 2016, pp. 31-44.

[5]R. Y. Choi, et al,”Introduction to Machine Learning, Neural Networks and Deep Learning”, Translational Vision Science and Technology, vol. 9, 14, February 2020.

[6] S. Albawi, T. A. Mohammed and S. Al-Zawi, "Understanding of a convolutional neural network," 2017 International Conference on Engineering and Technology (ICET), Antalya, 2017, pp. 1-6.

[7]X.Wang, W.Jiang and Z.Luo,”Combination of Convolutional and Recurrent Neural Network for Sentiment Analysis of Short Texts”,in Proc. of the 26th international conference on computational linguistics: technical papers,december, 2016, pp. 2428-2437.

[8] D. Optiz and R.Maclin, “ Popular Ensemble Methods: An Empirical Study”,Journal Of Artificial Intelligence Research, vol. 11, August, 1999, pp. 169-198.

[9]O. Sagi and L. Rokach. "Ensemble learning: A survey." Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 8, no. 4, February,2018.

[10] S. Dzeroski and B. Zanko, “Is Combining Classifiers with Stacking Better than Selecting the Best One?”,Journal of Machine Learning, vol. 54, March, 2004, pp. 255-273.

[11] K.Madukwe, X. Gao and B. Xue,“In Data We Trust: A Critical Analysis of Hate Speech Detection Datasets”,in Proc.of the Fourth Workshop on Online Abuse and Harms, November, 2020, pp. 150-161.

[12] P. Burnap and M.L. Williams, “Us and them: identifying cyber hate on Twitter across mul-tiple protected characteristics”, EPJ Data Science, 2016.

[13]Z. Waseem and D. Hovy,”Hateful sym-bols or hateful people? predictive features for hate speech detection on Twitter”. in Proc. of theNAACL Student Research Workshop, June, 2016, pp. 88–93.

[14]T. Davidson, et al.” Automated Hate Speech Detection and the Problem of Offensive Language”. In Proc. of the 11th International AAAI Conference on Web and Social Media, ICWSM ’17, 2017, pages 512–515.

[15]A. Founta, et al, ”Large Scale Crowdsourcing and Characterization of Twitter Abusive Behavior”, In Proc. of AAAI International Conference on Web and Social Media (ICWSM), 2018.

[16]W. Warner and J. Hirschberg, “DetectingHate Speech on the World Wide Web”, In Proc. of the 2012 Workshop on Language in SocialMedia, June, 2012, pp. 19–26

[17]Z. Zhang, D. Robinson and J. Tepper, ”Detecting hate speech on twitter using aconvolution-gru based deep neural network”. In TheSemantic Web: European Semantic Web Conference, 2018, pp. 745–760,

[18]J, Qian, et al, “A benchmark dataset for learning to intervene in online hate speech.” In Proc. of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019, pp. 4757–4766

[19]V. Basile,et al, “Multilingual detection of hate speech against immigrants and women in twitter”, In Proc. of the 13th International Workshop on Se-mantic Evaluation (SemEval-2019). Association for Computational Linguistics, 2019.

[20]M. Zampieri et al,”Predicting the Type and Target of Offensive Posts in Social Media”, InProc. Of the 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL), April, 2019.

[21]H. Mubarak, et al, “Arabic offensive language on twitter: Analysis and experiments.”, May, 2020, arXiv:2004.02192.

[22] B. Santoso, et al. "Synthetic over sampling methods for handling class imbalanced problems: a review." In IOP conference series: earth and environmental science, vol. 58, no. 1, 2017.

[23]Z. Zhong, et al,”Random Erasing Data Augmentation.”, In Proc. of 34th AAAI conference on Artificial Intelligence, 2020, pp. 13001-13008.

[24] O. Levy and Y. Goldberg,”Dependency-Based Word Embeddings”. In Annual Meeting of the Association for Computational Linguistics (ACL),June , 2014, pp. 302-308.

[26]T. Mikolov, et al. "Enriching word vectors with subword information." Transactions of the Association for Computational Linguistics, vol.5 ,2017, pp.135-146.

[27] Pennington et al.. "Glove: Global vectors for word representation." Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014.

[28]H. Jwa, et al, “exBAKE: Automatic Fake News Detection Model Based on Bidirectional Encoder Representations from Transformers (BERT)”, Applied Sciences, vol.9, no.19, 2019.

[29]Miaschi, Alessio, and Felice Dell’Orletta. "Contextual and Non-Contextual Word Embeddings: an in-depth Linguistic Investigation." Association for Computational Linguistics,July, 2020.

[30] Pires, et al. "How Multilingual is Multilingual BERT?." Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2019.

[31] Farha, Ibrahim Abu, and W. Magdy. "Mazajak: An online Arabic sentiment analyser." Proceedings of the Fourth Arabic Natural Language Processing Workshop, 2019.

[32] Baly, Fady, and H. Hajj. "AraBERT: Transformer-based Model for Arabic Language Understanding." Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection. May 2020, pp.9-15.

[33]S.Qaiser and R. Ali, “Text mining: use of TF-IDF to examine the relevance of words to documents.” International Journal of Computer Applications,vol. 181. no. 1, 2018, pp.25-29.

[34] T. Davidson,et al. “Automated hate speech detection and the problem of offensive language”, Eleventh international aaai conference on web and social media,2017

[35] S. Agrawaland and A. Awekar,” Deep learning for detecting cyberbullying across multiple social media platforms”, European Conference on Information Retrieval,Springer,2018 ,pp.141–153.

[36] Y. Lee, S. Yoon, and K. Jung.,” Comparative studies of detecting abusive language on twitter”,Computing Research Repository, Association for Computational Linguistics, October, 2018.

[37]A.M. Founta et al.” A unified deep learning architecture for abuse detection.”, In Proc. of the 10th ACM conference on web science, 2019.

[38] M. Ibrahim, M. Torki and N. El-Makky, "Imbalanced Toxic Comments Classification Using Data Augmentation and Deep Learning," 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), Orlando, FL, 2018, pp. 875-878.

[39]Marcos Zampieri et al.” Identifying and categorizing offensive language in social media” In Proc. of the 13th International Workshop on Semantic Evaluation, March, 2019

[40]A. Abuzayed and T. Elsayed , ”Quick and Simple Approach for Detecting Hate Speech in Arabic Tweets,” osact2020 workshop,Marseille, France. May, 2020 ,pp.109-114.

[41]I. Abu-Farha and W. Magdy , “Multitask Learning for Arabic offensive Language and Hate-Speech Detection,” Language Resources and Evaluation Conference (LREC 2020),Marseille, France. May, 2020, pp.86-90.

[42] Mubarak, Hamdy, et al. "Overview of osact4 arabic offensive language detection shared task." Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection. 2020.

[43] S. J. Pan and Q. Yang, "A Survey on Transfer Learning," in IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 10, Oct. 2010, pp. 1345-1359, doi: 10.1109/TKDE.2009.191.

Figures links

Figure (1.1), (2.1)
https://www.rcmediafreedom.eu/Dossiers/Hate-speech-what-it-is-and-how-to-contrast-it (L.V 16/3/2021)

Figure 1.2
https://www.statista.com/statistics/262946/share-of-the-most-common-languages-on-the-internet/#:~:text=As%20of%20January%202020%2C%20English,with%20a%2019.4%20percent%20share.(L.V 16/3/2021)

[26] Heyman, Geert, et al. "Learning unsupervised multilingual word embeddings with incremental multilingual hubs." Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,
Vol. 1, june, 2019.