People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

University Larbi Tébessi - Tébessa

Faculty of Exact Sciences and Natural and Life Sciences

Department: Mathematics and Computer Science

A Dissertation Submitted in Partial Fulfillment of the Requirement for Master's Degree in Computer Science

Specialty: Systems and Multimedia

An efficient Hate Speech Detection from Arabic Tweets

Presented by:

Haouaouchi Kheir Eddine

Supervised by :

Dr. Daouadi Kheir Eddine

Dr. Boualleg Yaakoub

In front of jury composed of :

Mr.Zemmar Ammar	MAA Larbi Tebessi University	President
Mr.Zebdi Abd El Moumen	MAA Larbi Tebessi University	Examiner

Year:

2021/2022

Abstract

Today, hate speech classification from Arabic tweets has drawn the attention of several researchers. Many systems and techniques have been developed to resolve this classification task. Nevertheless, two of the major challenges faced in this context are the limited performance and the problem of imbalanced data. In this study, we propose a novel approach that leveraged ensemble learning and semi-supervised learning based on previously manually labeled. We conducted experiments on a benchmark dataset by classifying Arabic tweets into 5 distinct classes: non-hate, general hate, racial, religious or sexism. Experimental results show that: (1) ensemble learning based on pre-trained language models outperforms existing related works; (2) Our proposed data augmentation improves the accuracy results of hate speech detection from Arabic tweets and outperforms existing related works. Our main contribution is the achievement of encouraging results in Arabic hate speech detection.

Résumé

Aujourd'hui, la classification des discours de haine à partir des tweets arabes a attiré l'attention de plusieurs chercheurs. De nombreux systèmes et techniques ont été développés pour résoudre cette tâche de classification. Néanmoins, deux des défis majeurs rencontrés dans ce contexte sont les performances limitées et le problème des données déséquilibrées. Dans cette étude, nous proposons une nouvelle approche qui tire parti de l'apprentissage d'ensemble et de l'apprentissage semi-supervisé basé sur des étiquetages manuels préalables. Nous avons mené des expérimentations sur un ensemble de données de référence en classant les tweets arabes en 5 classes distinctes : non-haine, haine générale, racial, religieux ou sexisme. Les résultats expérimentaux montrent que : (1) l'apprentissage d'ensemble basé sur des modèles de langage pré-entraînés surpasse les travaux connexes existants ; (2) Notre proposition d'augmentation des données améliore la précision des résultats de la détection des discours de haine à partir des tweets arabes et surpasse les travaux connexes existants. Notre principale contribution est l'obtention de résultats encourageants dans la détection des discours de haine en arabe.

ملخص

في أيامنا الحالية استحوذ تصنيف خطاب الكراهيَة من التغريدات العربية على اهتمام العديد من البحوث. حيث تم تطوير العديد من الأنظمة والتقنيات لحل مشكلة التصنيف هذه. ومع ذلك، فقد واجهت دراسات هذا السياق إثنين من التحديات الرئيسية هي الأداء المحدود ومشكلة البيانات غير المتوازنة. في هذه الدراسة، نقترح نهجًا جديدًا يعتمد تحديدا التعلم الجماعي والتعلم شبه الخاضع للإشراف بناءً على البيانات المصنفة يدويًا مسبقًا. لقد أجرينا تجارِب على مجموعة بيانات معيارية عن طرق تصنيف التغريدات العربية إلى 5 فئات متميزة: غير الكراهيَة أو الكراهيَة العامة أو العنصرية أو الدينية أو التمييز الجنسي. أظهرت نتائج التجارِب أن: (1) التعلم الجماعي القائم على نماذج اللغة المدربة مسبقًا يتفوق على الأعمال الحالية ذات الصلة ؛ (2) تعمل الطريقة المقترحة لزيادة البيانات على تحسين نتائج دِقَّة الكشف عن الكلام الذي يحض على الكراهيَة من التغريدات العربية وتتفوق في الأداء على الأعمال الحالية ذات الصلة. مساهمتنا الرئيسية هي تحقيق نتائج مشجعة في اكتشاف الكلام الذي يحض على الكراهيَة باللغة العربية.

This modest work is dedicated to :
The dearest being of my life, my mother (Fatiha).
The one who made me a man, my father (Tayeb).
My dear brother and sisters (Houssem), (Khaoula), (Sarah).
To the dearest person in my life, my dear wife (Amani).
All my friends: Rami, Nacer-Eddine, Abd-Elmounem...
To all my teachers who have supervised me throughout the years of my studies.
To my dear supervisors, dr.Daouadi Kheir eddine and dr.Boualleg Yaakoub who believed in my work and patiently supported me.
I dedicate this work to all those who participated in my success.

Acknowledgements

First and foremost, I would like to thank my parents who have encouraged me at every step in life.

I would like to thank my wife who has been by my side at all times, and thank her for everything she has done for me.

I would like to thank my supervisors, Dr.Daouadi Kheir Eddine and Dr.Boualleg Yaakoub for their time and investment in all aspects of my work.

I would also like to thank the members of the Jury: Mrs A.Zebdi and Mrs A.Zemmar.

I would also like to thank the Master 2 students that I had the pleasure of studying with.

I thank all the people I have been able to meet and with whom I have been able to exchange.

These thanks would not be complete without thanking all my teachers for the 2021/2022 school year.

Thanks to all my family.

Contents

List of Figures

List of Tables

List of Abbreviations

NLP	Natural Language Processing.
HS	Hate Speech.
ML	Machine Learning.
NB	Naive Bayes.
SVM	Support Vector Machine.
TF	Term Frequency.
IDF	Inverse Document Frequency.
BoW	Bag of Words.
LSTM	Long Short Term Memory.
CNN	Convolution Neural Network.
BERT	Bidirectional Encoder Representation from Transforms.
RNN	Recurrent Neural Network.
RF	Random Forest.
BiLSTM	Bidirectional Long Short Term Memory.
GRU	Gated Recurrent Unit.
MLM	Masked Language Model.

Chapter 1

Introduction

Nowadays, the use of social media has substantially increased in Arab countries, which has allowed more freedom for speech in different domains. Twitter for example is one of the leading social media where users can share short text of up to 280 characters optionally followed by a link, video, or photo, known as a tweet. This free micro-blogging allows users to subscribe, follow other users, share content, like other tweets, repost another tweet and reply to another tweet. Today, Twitter is booming, the service has more than hundreds of millions of users producing over five hundred million tweets per day[1]. Figure 1.1 shows the percentage of some Arabic populations that used Twitter in 2022. Notably, the population of Saudi Arabia and Oman are the most countries that used the platform. The Arab users generate 27.4 million tweets per day (Al-Hassan and Al-Dossari, 2021). From that big number, we can assume that hate speech can spread easily and quickly through this platform.

The General Policy Recommendation no. 15 of the European Commission[2] has described Hate Speech (HS) as "the advocacy, promotion or incitement, in any form, of the denigration, hatred or vilification of a person or group of persons, as well as any harassment, insult, negative stereotyping, stigmatization or threat in respect of such a person or group of persons and the justification of all the preceding types of expression, on the ground of race, color, descent, national or ethnic origin, age, disability, language, religion or belief, sex, gender, gender identity, sexual orientation, and other personal characteristics or status".

Initially, this sort of content was spread through the traditional platforms, but with worldwide accessibility through social media like Twitter more and more users are sharing their opinions. Unfortunately, sometimes these opinions could have negative psychological effects on social media users and even lead them to commit suicide (Hinduja and Patchin, 2010).

The great-uncontrolled content disseminated on social media is a known phenomenon that can raise social alarms, specifically when this content contains HS. One strategy proposed by the European Union to face this challenge is through legislation. The European Union Commission has pressured various platforms to sign an HS code. The media platforms have pledged to review the 'majority of notifications for removing of illegal HS' in less than 24h. However, this pledge is very difficult to accomplish due to the exact

[1] https://www.omnicoreagency.com/twitter-statistics/ Accessed 20/05/2020

[2] http://hudoc.ecri.coe.int/eng?i=REC-15-2016-015-ENG Accessed 20/05/2020

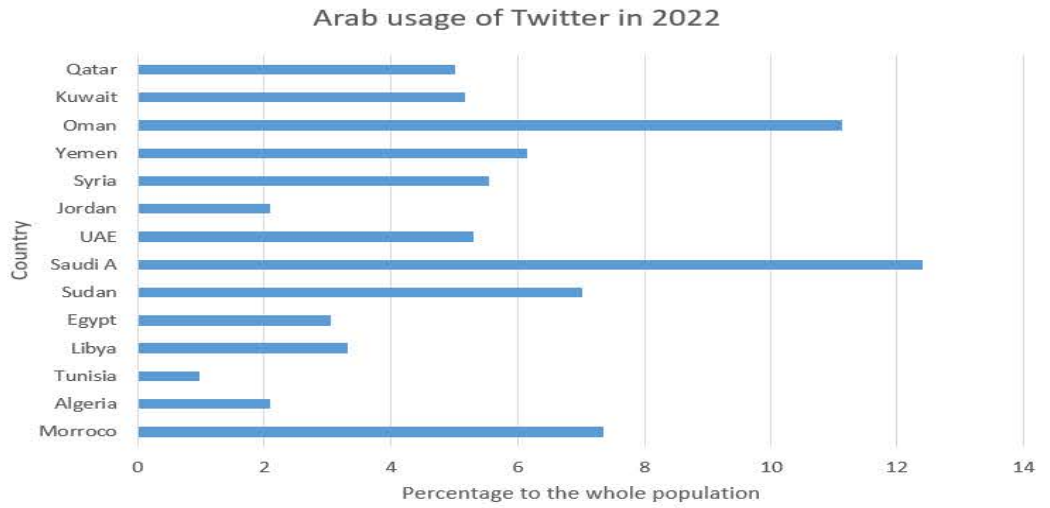

FIGURE 1.1: Arab usage of social media (2020).

scope remaining unclear because of the lack of data collection and systematic reporting on its occurrence. To resolve this issue they depend on their community to report HS content. This task for social media platforms is considered a complex one. Therefore, the severe consequences of this issue combined with the lack of available automatic systems and the huge amount of data disseminated daily encourage the text classification community to initiate research into HS detection.

Natural language processing (NLP) is a field of research that deals with machine learning (ML) algorithms applied to natural textual human languages. NLP applications consist to automatically processing written human languages, including HS detection, sentiment analysis, and text classification.

The Arabic texts are known by their complexity, where it is very difficult to understand the intent of the user (Salem, 2017). Furthermore, some tricky variances can be found such as the usage of the neglection of diacritics and writing from right to left (Alshutayri and Atwell, 2018). In addition, social media data is generally considered as unstructured text, which includes regular natural language used in our daily life. Thus it is very hard to extract insights from textual data since they are context-dependent sentences. However, NLP methods can interpret the variability of unstructured data (Irfan et al., 2015). Also, previous studies report multiple challenges in working to analyze hate speech : racial bias, code-switched language complexity, datasets, and context limitation.

Today, through Twitter, researchers propose approaches for Arabic HS detection. Nevertheless, two of the major challenges faced in this context are the limited performance and the problem of imbalanced data, which make them suffer from severe over-fitting (Founta et al., 2018). Automatic HS detection from Arabic tweets using traditional machine learning classifiers like Naïve Bayes (NB), Support Vector Machine (SVM) and Random Forest (RF) have

shown reasonable accuracy results. However, they are based on the hand-crafted features calculated based on some pre-defined methods like Term Frequency (TF), Term Frequency-Inverse Document Frequency (TF-IDF), Bag of Word (BoW), etc. Recently, Long Short Term Memory (LSTM) and Convolution Neural Network (CNN) and Bidirectional Encoder Representations from Transformers (BERT) have already shown good results for HS classification from Arabic tweets.

In summary, leveraging a single classifier produces poor performance results. In addition, the imbalanced nature of data often has been disregarded in the major related works, which might have a substantial impact on the classification results. This problem has occurred in the HS detection where most labeled datasets are highly imbalanced. Here comes the role of the ensemble learning approach and data augmentation method to obtain better predictive performance than single and imbalance learning algorithms.

In our study, we leveraged ensemble learning that relies on a pre-trained Bidirectional Encoder Representation from Transformer (BERT) language models for the Arabic language (Antoun et al., 2020; Abdul-Mageed et al., 2020) to perform HS classification. Specifically, we fine-tuned various Arabic BERT models trained on various Twitter data. In addition, we proposed a semi-supervised learning method based on previously labeled data. The research questions investigated in this research are described as follows:

- How do increase the accuracy results of the Arabic hate speech detection task?
- Can data augmentation improve the accuracy results of the Arabic hate speech detection?

Our research contributions can be summarized as follows:

- We leveraged various Arabic BERT models via transfer learning and fine-tuning to build our baseline classifier.
- We evaluated ensemble learning based on the leveraged Arabic BERT models.
- We propose a data augmentation method based on semi-supervised learning and previously labeled data.

The rest of this thesis is organized as follows: Chapter 2 reviews the most recent related works on Arabic hate speech detection. In Chapter 3, we provide details of our methodology. Chapter 4 presents and discusses the results of the conducted experiments. Conclusion and future works are presented in Chapter 5.

Chapter 2

Related Work

2.1 Introduction

Chapter 1 detailed the general context and background for this thesis, provided a clear emphasis on importance of the topic, and present the research objectives and our research questions. Before initiating any piece of original research, it is critical to review existing literature thoroughly. To allow the identification of some key ideas and methodologies in the field where an addition to knowledge can be achieved.

The rest of this chapter is managed as follows. Section 2.2 provides a background of feature extraction techniques. Section 2.3 provides a background of classification algorithms. Section 2.4 provides an overview about existing Arabic HS detection approaches. Section 2.5 provides an overview of the existing data augmentation method for Arabic HS detection, and finally, Section 4.6 provides a background of data augmentation methods.

2.2 Background of Feature Extraction Techniques

Thanks to a wide range of feature extraction techniques, researchers through Twitter have followed three major techniques to capture features from the textual content of a tweet: Bag of Words, Term Frequency Inverse Document Frequency, and Word Embedding. The rest of this section is dedicated to present a background of the aforementioned techniques.

2.2.1 Bag of Words

Bag of Words (BoW) (Sousa Pereira Amorim et al., 2018) is the most popular strategy that has mainly aimed used for NLP. It focuses on the frequency of word occurrences without taking the order or sequence of words. This generates a vocabulary of specific words present in all tweets in the learning set, and these will be used as feature vectors representing the presence or absence of each word in the vocabulary as shown in Figure 2.1.

2.2.2 Term Frequency Inverse Document Frequency

TF-IDF (Vadivukarassi et al., 2018) is a weighting scheme based on the combination of TF and IDF. This is often used in Information Retrieval and Text

FIGURE 2.1: Bag of Words example.

Mining (Leskovec et al., 2020), which is computed by combining the product of a function of term frequency ($f_{w,t}$) and a function of the inverse of document frequency ($1/N_t$) as follows.

- TF measures how frequently a word w occurs in a tweet t by dividing it by the maximum number of occurrences of any word w in the same tweet t (Luhn, 1957), and is calculated as follows:

$$TF_{w,t} = F_{w,t}/(max(F_{w,t})) \tag{2.1}$$

- IDF measures how important a word w is by computing the logarithm of the number of tweets in the corpus divided by the number of tweets in which a given word appears (Jones, 1972), and is calculated as follows:

$$IDF_{w,t} = log(N/df_w) \tag{2.2}$$

The combination of the aforementioned Eqs is formulated as follows:

$$TF - IDF_{w,t} = TF_{w,t} * IDF_{w,t} \tag{2.3}$$

where: N is the total number of tweets in the corpus, df_w is the number of tweets in the corpus that contain the word w, and $F_{w,t}$ is the term frequency in which the word w appears in a tweet t.

2.2.3 Word Embedding

Word Embedding (Yang et al., 2018) is another variant of feature extraction techniques, where words are presented as vectors in an unbroken space. Word Embedding have recently attracted much attention in many NLP tasks as in (Barhoumi et al., 2019; Laatar et al., 2017), because of its capability to capture semantic and grammatical relations between words from a large

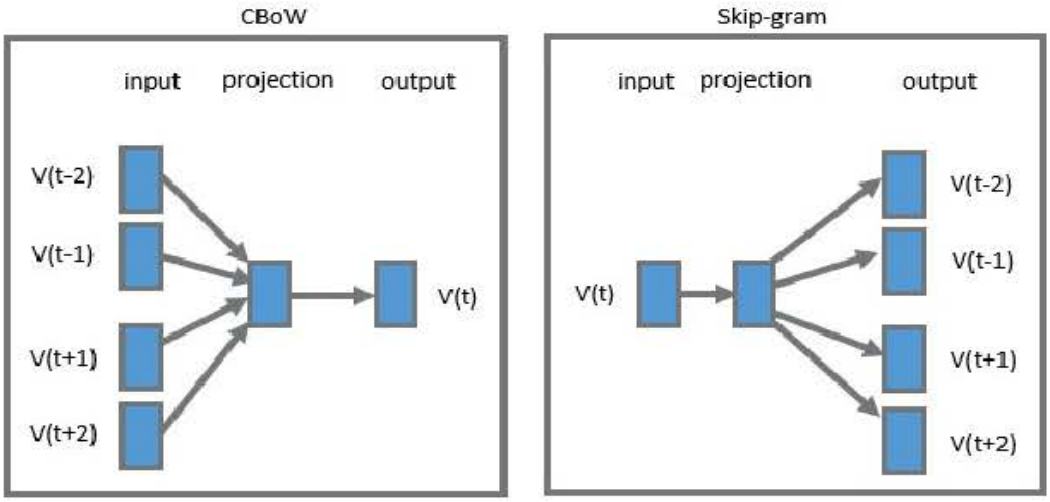

FIGURE 2.2: Continuous Bags of Words and skip-gram model architecture.

amount of text. In the literature, researchers has followed three main methods to train the Word Embedding model are described in the following:

- Word2vec (Mikolov et al., 2013) is a neural network model that yields a real-valued vector of words showing up in a text corpus. Word2vec is a technique that can identify the words that are similar in terms of their syntactic and semantic relationships. It uses a window to define the context in which these words are placed. To train the Word2vec model, the researchers used two main common techniques. The first uses adjacent words to predict a word goal known as a continuous bag of words, and the second uses a word to predict adjacent words in a sentence, the so-called Skip-gram. The illustration is shown in Figure 2.2

- Global Vectors for word representation (Glove) (Pennington et al., 2014) is an extension of word2vec, which instead of using a window to determine the local context, uses statistical computation across the entire ensemble to create a clear co-occurrence of the word matrix. Word2vec offers better word vector representations with a limited dimensional semantic space when compared to Glove. For example, the publicly accessible Glove model, which has 300-dimension vector representations of 400K words and was trained on Wikipedia.

- Fasttext (Joulin et al., 2017) is a modification of word2vec that represents each word as an n-gram of characters rather than learning vectors directly from words. For example, the Fasttext representation of the word "political" with n=4 is poli, olit, liti, itic, tica, ical >. This aids in the comprehension of shorter words. For example, the publicly accessible Fasttext model, which has 300-dimension vector representations of 1 million words and was trained on Wikipedia (Joulin et al., 2017).

2.3 Background of Classification Algorithms

Thanks to a wide range of classification algorithms, researchers through Twitter have leveraged two principal type of classification algorithms: a traditional and a deep learning. The rest of this section is dedicated to present a background of the aforementioned algorithms.

2.3.1 Traditional Algorithms

In this context, traditional learning classifiers like Multinomial Naïve Bayes and Support Vector Machine have shown good results for classifying Twitter data. The rest of this subsection is dedicated to present a brief overview of traditional learning classifiers.

2.3.1.a Multinomial Naïve Bayes

Multinomial Naïve Bayes (MNB) (McCallum, Nigam, et al., 1998) is an extension of Naïve Bayes classifier that takes into account the repetition of words to produce the distribution of data in a polynomial manner. The BoW method is used to compute the tweet vectors, and the Naïve Bayes technique is used to classify them. This classifier has been widely used in studies involving Arabic HS detection as in (Husain, 2020a).

2.3.1.b Support Vector Machine

Support Vector Machine (SVM) is one of the most popular classifiers because it is very accurate and efficient in classifying text. The main advantage of this classifier is that it usually performs well even with a small amount of training data. SVM has been widely used in studies involving Arabic HS detection as in (Chowdhury et al., 2020a).

2.3.2 Deep Learning Algorithms

In this context, deep learning classifiers such as Recurrent Neural Network, Convolutional Neural Network, and Bidirectional Encoder Representation for Transformers have shown good results for classifying Twitter data. The rest of this subsection is dedicated to present a brief overview of the aforementioned classifiers.

2.3.2.a Convolution Neural Network

Convolution Neural Network (CNN) (Krizhevsky et al., 2012) is one of the most famous and successful deep neural network models. CNN is an alternative way to feed a forward neural network where the different layers, are sparsely connected. This is accomplished by connecting a local region to an input layer by neurons in the next layer. When processing text data, it must first be converted into numerical values, by converting text words into word vectors known as word embedding. A 2-dimensional array corresponding to

the text sentence is formed where each row corresponds to a symbol or word. Then the various CNN steps consist of a convolution layer, a pooling layer and a fully connected layer. CNN has been very popular in research involving Arabic HS detection as in (Abuzayed and Elsayed, 2020; Faris et al., 2020; Alsafari et al., 2020a; Alsafari et al., 2020b).

2.3.2.b Recurrent Neural Network

Recurrent neural network (RNN) (Mikolov et al., 2010) is another solution to address the sequential learning problem posed by the traditional neural network. Because of the inverse propagation of the error that required the network to calculate the gradient to update the weights, RNN had difficulty remembering Long Short term memories. Thus, RNNs faced the problem of disappearing/exploding gradients. Long Short Term Memory (LSTM) (Glorot et al., 2011) is leveraged instead of RNN to solve this problem by employing a memory cell at each time step t. Thus, the algorithm iterates through a function F to update the network's hidden states and create the outputs. LSTM has been very popular in research involving Arabic HS detection as in (Faris et al., 2020; Duwairi et al., 2021).

2.3.2.c Bidirectional Encoder Representation for Transformers

Bidirectional Encoder Representation for Transformers (BERT) (Devlin et al., 2018) is a powerful transformer-based architecture that provides advanced results in various natural language processing tasks. The BERT framework is divided into two parts, each of which requires pre-training and fine-tuning. BERT is pre-trained as a mask language model, as opposed to left-to-right or right-to-left models. During this process, some input tokens are randomly hidden and then predicted. Its also pre-trained to pick up the connection in dual relationships. The Arabic version of BERT has been widely used in research including Arabic HS detection as in (Elmadany et al., 2020; Aldjanabi et al., 2021; Alghanmi et al., 2020).

2.4 Arabic Hate Speech Detection

Thanks to its importance, Tweet classification problem has drawn the attention of several studies such as topic classification (Daouadi et al., 2021) bot detection (Daouadi et al., 2019a; Daouadi et al., 2020; Daouadi et al., 2019b) organization detection (Daouadi et al., 2018a; Daouadi et al., 2018b) and hate speech detection (Husain, 2020b). In particular, HS detection from Arabic tweets has drawn the attention of several researchers. In the literature, many systems have been developed to resolve this classification problem. They follow two main principal approaches: a traditional approach, and a deep learning approach.

2.4.1 Traditional Approaches

In this context, researchers have focused on manual features engineering based on local metadata and rely on the presence of tweet features like hashtags. Some examples of traditional approaches are briefly described in the following.

Besides, authors in (Mubarak et al., 2020) focuse on the problem of detecting offensive tweets. They use features from the Arabic word2vec model with SVM, this yielded an Accuracy result of 88.6%.

Likewise, authors in (Husain, 2020b) classify tweets being offensive or not. The best experimental results are obtained using the trained ensemble learning in offensive language detection, and this yielded F1 score of 88%, which exceeds the score obtained by the best single learner classifier by 6%.

Furthermore, authors in (Husain, 2020c) examine the effect of the preprocessing step on offensive language and HS classification. They claimed that an intensive preprocessing technique demonstrates its significant impact on the classification rate. The optimal experimental results are obtained using BoW and SVM, yielding F1 score results of 89% and 95% for offensive language and HS classification.

In a different strategy, authors in (Mubarak et al., 2017) classify user accounts being abusive or not. The best experimental results are obtained using list-based methods (Seed Words + Log Odds Ratio), yielded F1-score of 60%.

2.4.2 Deep Learning Approaches

In this context, researchers follow two principal approaches are deep learning from scratch approaches and fine-tuning approaches.

2.4.2.a Deep Learning from Scratch Approaches

In this context, researchers have focused on automatic feature engineering. They used deep learning models such as CNN and LSTM. Some examples of deep learning approaches are briefly described in the following.

Authors in (Alharbi and Lee, 2020) classify Arabic tweets being offensive or not. They use features from n-gram and word embedding model. The best experimental results are obtained using LSTM. This yielded Accuracy result of 74%.

In a similar, authors in (Farha and Magdy, 2020) investigate the impact of using Multitask learning on offensive and HS detection. The showed that Multitask learning based on CNN-LSTM improve the accuracy results, yielded macro F1-score of 90.4% and 73.7% for classifying offensive and HS tweets, respectively.

Likewise, authors in (Rachid et al., 2020) classify tweets being cyberbullying or not. They use features from the Arabic Word2vec model achieving. The best experimental result is obtained using a simple CNN-LSTM model, yielded F1-score of 84%.

Furthermore, authors in (Husain et al., 2020) use features based on TF-IDF to classify tweets from those of offensive or not. The best experimental

results are obtained using a Bidirectional gated recurrent unit classifier, and this yielded a macro F1-score of 83%.

In a similar, authors in (Abuzayed and Elsayed, 2020) classify tweets being HS or not based on the Arabic Word2vec model. They conducted a series of experiments using CNN, RNN and SVM. This yielded a macro F1-score of 73%.

Besides, authors in (Faris et al., 2020) classify tweets from those of hateful and normal ones. They used CNN-LSTM with different versions of Arabic Word2vec, which yielded an F1-score of 71.68%.

Likewise, authors in (Alsafari et al., 2020b) investigate the effect of word embedding models and neural network architectures on the accuracy rate using (Offensive and Hate vs. Clean), (Hate vs. Offensive vs. Clean) and (Nationality hate vs. Clean vs. Religion Hate vs. Gender Hate vs. ethnicity hate vs. Offensive) classification tasks. The best results are achieved using CNN and word2vec based on the Skip-gram model, which yielded F1-score results of 70.80% 75.16% 87.22%, and for six-class, three-class, and two-class classification tasks, respectively.

Besides, authors in (Alsafari et al., 2020a) use the AraBERT embedding model with an ensemble of CNNs and Bidirectional LSTM (BiLSTMs) classifiers. The best results are achieved using the average-based ensemble approach, which yielded F1-score results of 91.12% (CNNs), 84.01% (CNNs), and 80.23%(BiLSTMs), for two-class, three-class, and six-class classification tasks, respectively.

In a similar, authors in (Alsafari et al., 2020c) use CNN with the Multilingual BERT embedding model, which yielded F1 score results of 87.03% 78.99%, and 75.51% for two-class, three-class, and six-class classification tasks, respectively.

Furthermore, authors in (Duwairi et al., 2021) conducted a series of experiments using CNN, BiLSTM-CNN, and CNN-LSTM. They reported three types of experiment: Binary classification (Hate or Normal), Ternary classification (Hate, Abusive, or Normal), and Multi-class classification (Misogyny, Racism, Religious Discrimination, Abusive, and Normal). In the binary classification task, the CNN model outperformed other models and achieved an accuracy of 81%. In the ternary classification task, both the CNN, and BiLSTM-CNN models achieved the best accuracy of 74%. While in multi-class classification task, the best results are achieved by CNN-LSTM, and BiLSTM-CNN, yielded an Accuracy of 73%.

In similar, authors in (Alshaalan and Al-Khalifa, 2020) classify tweets being hate or not, they compared four models: CNN, GRU, CNN-GRU, and BERT. Their experimental results show that CNN model gives the best performance with an F1-score of 79%.

Similarly, authors in (Al-Hassan and Al-Dossari, 2021) classify tweets into five distinct classes: none, religious, racial, sexism and general hate. They used four deep learning models: LSTM, CNN-LSTM, GRU, and CNN-GRU. Their experimental results show that the hybrid CNN-LSTM model gives the best performance with an F1-score of 73%.

Besides, authors in (Hassan et al., 2020) evaluate the combination of SVM, CNN, CNN-BiLSTM on offensive language and HS classification tasks. These models showed a significant performance with 90.51% macro F1-score for offensive language detection and 80.63% macro F1-score for hate speech detection.

In a different strategy, authors in (Alghanmi et al., 2020) use both Arabic BERT and Arabic Word2vec) embedding models classify tweets being normal, hateful, or abusive. The best accuracy result is achieved using CNN, which yielded an F1 score of 72.1%.

2.4.2.b Transfer Learning Approaches

In this context, researchers have focused on transfer learning based on fine-tuning. The majority of transfer learning approaches focus mainly on the Arabic versions of BERT. Some examples of transfer learning approaches are briefly described in the following:

Authors in (Elmadany et al., 2020) evaluate Arabic BERT on offensive language and HS classification tasks. This yielded a macro F1-score of 82.31% and 70.51% for classifying HS and offensive language classification task.

Likewise, authors in (Abdellatif and Elgammal, 2020) evaluate the Universal Language Model Fine-tuning on the offensive language and HS classification task, this yielded macro F1-score of 77% and 58% for classifying offensive and HS tweets.

Besides, authors in (Aldjanabi et al., 2021) studied the impact multitask learning model built on top of a pre-trained Arabic BERT on offensive and HS classification tasks, which yielded F1-score of 92.34% and 88.73% for classifying offensive and HS tweets.

2.5 Data Augmentation Methods

In this context, researchers have focused on generating synthetic data in order to face the challenge of imbalanced learning. Some examples of data augmentation methods are briefly described in the following:

Authors in (Husain, 2020c) investigate the impact of upsampling technique (duplicate tweets of the minority class to balance the class label) on offensive and HS classification tasks. They showed that upsampling decrease the result of F1-score results.

In similar, the authors in (Haddad et al., 2020) propose two data augmentation methods for offensive and HS classification. The first one uses an external augmenting technique by adding some offensive tweets from another labeled data. While the second one uses the random oversampling method by shuffling the words into HS tweets to create new samples.

Likewise, authors in (Elmadany et al., 2020) investigate the impact of seed words and tweet emotion for automatic annotate tweets for offensive and HS classification tasks. They showed that the examined method has a positive and negative impact on the offensive and HS classification tasks, respectively.

In a different strategy, authors in (Alsafari and Sadaoui, 2021) explore the role of semi-supervised built on unlabeled tweets for classifying tweets being offensive, hateful, and normal. They showed that an improvement of up to 7% was achieved from using additional pseudo-labeled tweets.

2.6 Conclusion

This chapter has reviewed the literature review; it has set out the background information from feature extraction techniques and classification algorithms. Furthermore, the chapter also reviewed existing works on Arabic HS detection and data augmentation methods. The next chapter will overview our proposed approach for Arabic HS detection from tweets.

Chapter 3

Proposed Approach

3.1 Introduction

Chapter 2 detailed the background of feature extraction techniques, provided a background of classification algorithms, presented state-of-the-art of Arabic HS detection and data augmentation methods.

HS classification task presented in this research is aimed to accurately classify tweets from those of Normal, General HS, Religious, Sexism, and Racism. Previous state-of-the-art approaches in this context have faced two major challenges owing to the limited performance results and the problem of imbalanced data. In this study, we proposed a novel approach for HS detection by leveraging ensemble learning and data augmentation based on semi-supervised learning built on previously manually labeled tweets.

The rest of this chapter is organized as follows: Section 3.2 present the problem formulation. Section 3.3 discuss our proposed approach, and finally Section 4.6 conclude the chapter.

3.2 Problem Formulation

The HS task presented in this thesis is a multi-class classification problem, which tries to uncover various characteristics of tweets. As with any other supervised classification problem, we need a labeled dataset. Thus, we use the taxonomy presented by (Al-Hassan and Al-Dossari, 2021), below we provide the definition and examples in Figures 3.1 for each of these HS types:

- **Religious:** Any Religious discrimination, such as: Islamic sects "Sunni, Sheie, Alrafdhah ...etc." Also, anti-Judaism or anti-Hinduisand, anti-Christian and their respective denominations, calling for atheism or other religions. Also attaching relations of following or not following a particular religious group, these groups include but not limited to: ISIS and Al-qaedah, Muslim Brotherhood. Al-Houthi and many others..

- **Racism:** Any Racial offense or tribalism, regionalism, prejudice against particular tribe or region, xenophobia (especially for migrant workers) and nativism (hostility against immigrants and refugees). Also, offending the appearance and color of individual or offending particular country leader or country politics.

FIGURE 3.1: Example of hate speech tweets. (A: Religious hate speech, B: Racism, C: Sexism, D: General hate speech, E: Non-hate speech).

- **Sexism:** Any post that offense particular gender using any form of hostility or devaluation based on person's gender. In addition, any form of misogyny tendency.
- **General hate speech:** Any general type of hate which is not mentioned in the previous classes. Whether it contains: general hatred, obscene, offensive and abusive words that are not related to religion, race or sex.
- **Non-hate (Normal):** If the tweet does not contain any form of hatred.

3.3 Proposed Approach

The main focus of this work will be to train a model with a multiclass dataset for the data augmentation purpose and then preprocess the concerned data

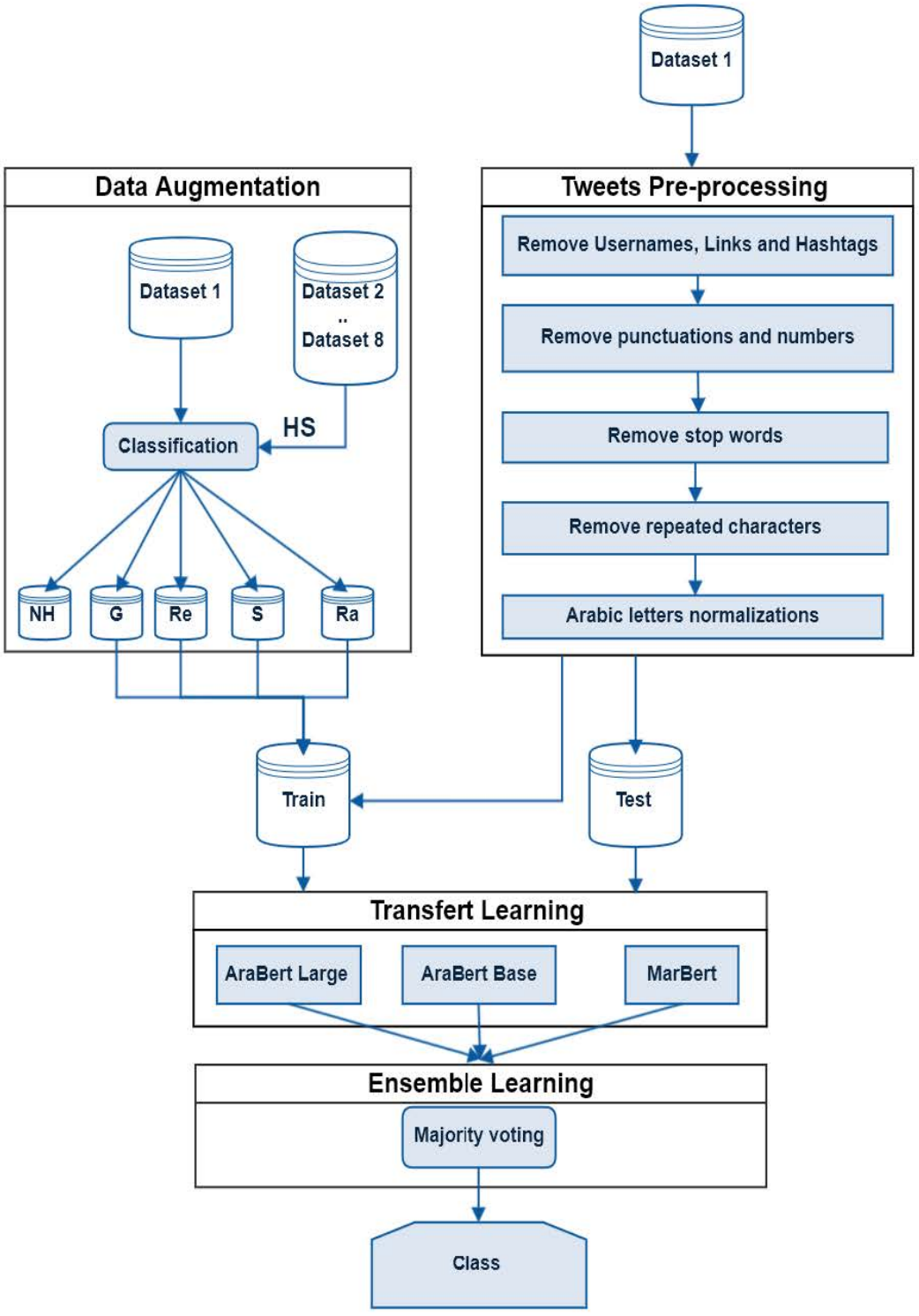

FIGURE 3.2: The process of transfer learning Corpus (NH: Normal, G: General hate speech, Re: Religious, S: Sexism, Ra: Racism).

through multiple phases to be fit to the transfer learning ensemble we will be working with. As shown in Figure 3.2, our proposed approach consists of four main steps are data augmentation, tweets pre-processing, transfer learning and ensemble learning.

3.3.1 Data Augmentation

The imbalance nature in data often has been disregarded in the major related works, which might have a substantial impact on the classification results. This problem has occurred in the HS detection where most labeled datasets are highly imbalanced. To face this issue, we propose semi-supervised learning built on previously manually labeled tweets. Firstly, we train the model using the state-of-the-arts datasets (Al-Hassan and Al-Dossari, 2021) labeled as (Non-hate, General HS, Sexism, Racial and Religious HS). The tweets labeled as religious HS in (Albadi et al., 2019; Alsafari et al., 2020c) are added directly to the to the datasest of (Al-Hassan and Al-Dossari, 2021). While the tweets previously labeled as HS in (Sun et al., 2019; Mulki et al., 2019; Haddad et al., 2019; Alshaalan and Al-Khalifa, 2020; Alsafari et al., 2020c; Ousidhoum et al., 2019) are classified with the trained model, and the new label is used balance the first datasets.

3.3.2 Tweets Preprocessing

The inputs of our proposition are the textual content of tweets composed of raw tweet content. In this step, our main objective is to clean the textual content produce a more consistent and standard tweet. We performed some pre-processing tasks as follows:

- Removing the tweet features: user mentions '@', URLs, the word RT, #, punctuation, special characters (emoticons), and numerical characters.
- Removing repeated characters: such as (مبروووك) to be (مبروك).
- Removing Arabic stop words, non-Arabic letters, new lines as well as diacritics.
- Arabic letters normalization: in the Arabic language, they are different variations for representing some letters which are:
  - Letter (ۀ) which can be mistaken and written as (ه), we normalized it to (ه).

- Letter (أ) which has the forms (ا, آ, إ, أ), all these four letters are normalized into (ا).
- The Arabic dash that is used to expand the word (مرحبا) to (مرحبـــا) has been removed.
- Letter (ى) has been normalized to (ي).

All tweets are padded to the length of the longest tweet and will be used as an input for the following step.

3.3.3 Transfer Learning

The second step aims to use the pre-trained language models (AraBert-Large, AraBert-Base, and MarBert).

Traditional Machine Learning (ML) technology has matured to a point where it may be used in a variety of practical applications with considerable success. ML does, however, have certain limits in some real-world circumstances. For example, gathering sufficient training data is prohibitively expensive, time-consuming, or impossible, necessitating the use of transfer learning methods. Transfer learning is a method of training artificial neural networks that depend on pre-trained models on specific tasks and data. Transfer learning assumes that if a pre-trained network solves one issue well, it may be utilized to tackle a similar but distinct problem with a little extra training. The reliance on a large amount of training data can be reduced in this way. Text classification (Duwairi et al., 2021), sentiment analysis (Ghahramani, 2007), and image classification (Wu and Dietterich, 2004; Boualleg et al., 2019; Boualleg and Farah, 2018; Boualleg et al., 2020; Hafdhellaoui et al., 2019; Amiri et al., 2020) are only a few of the applications of transfer learning.

The main goal of transfer learning is to adapt the acquired weights from Ds to learn the target conditional probability distribution $P(Yt|Xt)$ Dt with $Ds \neq Dt$ or $Ts \neq Tt$, given a source domain Ds and its accompanying source task Ts, as well as a target domain Dt and a target task Tt. Figure 3.3 show the process of transfer learning. Transfer learning is undeniably one of the most important aspects of language models. In the NLP world, the advent of Bidirectional Encoder Representation from Transformers (BERT) (Devlin et al., 2018) sparked a revolution. BERT is a deep learning model that has produced best in class performance on a wide range of NLP tasks.

BERT was taught using a technique known as Masked Language Modeling (MLM). The MLM operates by randomly concealing part of the existing tokens from the input. As a result, the masked word's original vocabulary id may be anticipated from the word's context. As a result, the MLM will be able to merge the left and right contexts. In addition to the MLM, BERT employs the Next Sentence Prediction task to train a competent language model that recognizes and understands sentence relationships. The BERT model architecture is a multi-layered bidirectional transformer encoder based on the (Devlin et al., 2018) original implementation. The transformers, it is suggested,

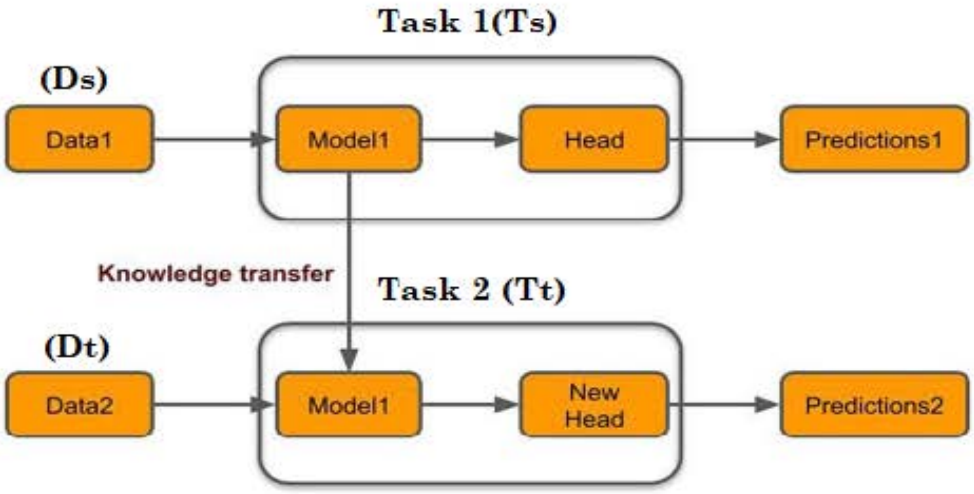

FIGURE 3.3: The process of transfer learning.

are a collection of numerous nested layers (or blocks). An 'attention' layer is present in each layer/block. For each block, BERT uses twelve (12) different attention mechanisms, allowing tokens from the input sequence (e.g., sentences made up of word or sub-word tokens) to focus on the other token. In their paper, (Devlin et al., 2018) (Vaswani et al., 2017) offered the following two architectures:

- **BERT Base:** 12 layers (Transformer blocks), 12 attention heads, and 110 million parameters.
- **BERT Large:** 24 layers, 16 attention heads, and 340 million parameters.

A preprocessing phase, consisting of tokenization, should be conducted before feeding a raw sentence to BERT. The sub-word tokenization technique WordPiece employs a vocabulary initialization to cover all characters in the training data. WordPiece added the appropriate merge rules for tokenization later on. WordPiece learns positional embeddings with a sequence length of up to 512 tokens and creates embeddings with a 30,000 character vocabulary. English and BooksCorpus are used to train BERT models. The single sentence categorization using the BERT model is shown in Figure 3.4. In Figure 3.4, E stands for input embedding, Ti for contextual representation of token i (Tok I), and [CLS] for classification output, which summarizes all the hidden states outputted from all tokens in the input sentence.

3.3.4 Ensemble Learning

Ensemble learning is a broad machine learning meta-approach that tries to improve predictive performance by mixing predictions from several models. In our research, we compared two ensemble learning methods are majority voting and average voting described as follows.

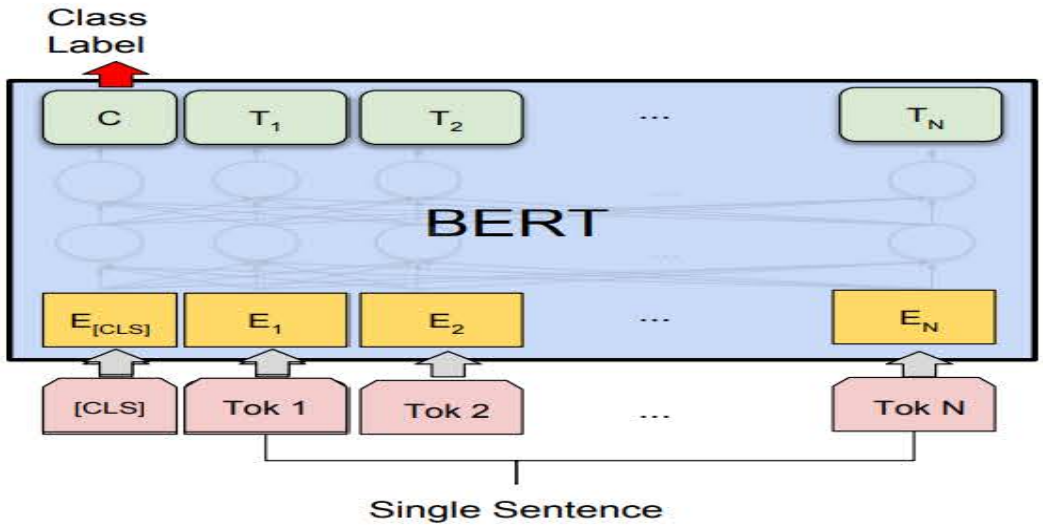

FIGURE 3.4: Single sentence classification using BERT.

Majority voting (also known as Hard Voting), each classifier votes for a class, and the class with the most votes win. The ensemble's anticipated target label is the mode of the distribution of individually predicted labels in statistical terms. This strategy is intentionally used to increase model performance, to out performs any one model in the ensemble.

Average voting (also known as Soft voting), where each classifier assigns a probability value to each data point that it belongs to. The predictions are totaled and weighted according to the relevance of the classifier. The vote is then cast for the target label with the largest sum of weighted probability.

3.4 Conclusion

This chapter has reviewed the proposed approach; it has set out the problem formulation. Furthermore, the chapter also reviewed our proposed method for data balancing. In addition, the chapter provides a detailed description of the proposed classification approach. The next chapter will present the experimental and evaluation results.

Chapter 4

Experimental Results

4.1 Introduction

Chapter 3 presented a detailed review of our proposition. This chapter is dedicated to presenting the experimental and evaluation results.

The rest of this chapter is organized as follows. Section 4.2 present an overview of the development environment and setup. Section 4.3 describe the performance measures used to evaluate our proposed approach. Section 4.4 describe our interesting datasets. Section 4.5 discuss the results achieved using our proposed approach.

4.2 Development Environment and Setup

We used Python 3.7 as the programming language for all of the experiments in our research, specifically, we used code editor Kaggle, which is an online community platform for data scientists and machine learning enthusiasts. Kaggle offers a no-setup, customizable, python notebooks environment by accessing Graphics Processing Unit at no cost, while allowing users to collaborate with others. The libraries used in our experiments are described as follows:

- **Pandas** is an open-source library that offers easy-to-use data structures and data analysis tools[1].
- **Numpy** is the most important Python module that includes a multidimensional array object, various derived objects (such as hidden arrays and matrices), and a variety of routines for performing fast array operations, such as math, logic, shape manipulation, sorting and more [2].
- **Scikit-learn** is undoubtedly Python's most helpful machine learning library. It contains many machine learning and statistical modeling algorithms, such as classification, and regression[3].
- **Pyarabic:** A Python library for manipulating Arabic letters and text, Pyarabic contains fundamental operations such as recognizing Arabic

[1] https://pandas.pydata.org/docs/Accessed 20/05/2020
[2] https://numpy.org/doc/stable/Accessed 20/05/2020
[3] https://scikit-learn.org/stable/user_guide.html/Accessed 20/05/2020

letters, Arabic letter groups and characteristics, deleting diacritics, and so on[4].

- **ktrain** is a Python library that makes deep learning and AI more accessible and easier to apply [5].
- **HuggingFace** is a platform provider and an open-source of machine learning technologies, that contains pre-trained language models[6].

To fine-tune the transfer learning model, authors in (Sun et al., 2019) have recommended selecting from the values of the following parameters: learning rate, batch size, number of epochs described as follows:

- The learning rate is a hyper-parameter that controls how much the model changes each time.
- The number of epochs is a hyper-parameter that specifies how many times the learning algorithm will iterate over the whole training dataset.
- The batch size is a hyper-parameter that specifies how many samples must be processed before the internal model parameters are updated.

4.3 Performance Measures

The tenfold cross-validation approach was used to evaluate the performance of our proposed approach. The dataset was split into ten equal-sized segments while maintaining the balance of each class in the corresponding datasets. One of these parts was used as a testing and the remainder were used as training. This procedure was repeated 10 times and the averaging of the performance results was obtained across the ten repetitions of cross-validation. The performance measures used to evaluate our proposed approach are as described follows.

- Recall, because it concentrates on the good examples, we should pay greater attention to that measure. For each class label, recall refers to the proportion of properly classified tweets to the number of tweets that belonged to that class but were mistakenly classified by the model. For example, the Recall of the Racial class is calculated as described in the following equation:

$$Recall = \frac{Racial\_tweets\_correctly\_classified}{Total\_number\_of\_Racial\_tweets} \tag{4.1}$$

- Precision, this metric is likewise useful in our instance; it reflects the proportion of relevant retrieved tweets from a specific class. For example, the Precision of religious class is calculated as described in the

[4] https://pyarabic.sourceforge.io/ Accessed 20/05/2020
[5] https://github.com/amaiya/ktrain/ Accessed 20/05/2020
[6] https://www.crunchbase.com/organization/hugging-face/ Accessed 20/05/2020

following equation:

$$Precision = \frac{Religious\_tweets\_correctly\_classified}{Tweets\_classified\_as\_"Religious"} \tag{4.2}$$

- F1-Score is more useful than accuracy in our instance of unequal class distribution since it is the weighted average of precision and recall.

$$F1 - Score = 2 \times \frac{Precision \times Recal}{Precision + Recall} \tag{4.3}$$

- Micro Calculate measures globally by counting the total true positives, false negatives, and false positives.
- Macro Calculate measures for each label, and find their unweighted mean, this does not take class imbalance into account.
- Weighted Calculate measures for each label and find their average weighted by support (the number of true instances for each label). This alters 'macro' to account for label imbalance.

4.4 Dataset Description

To validate our proposition, we used different Arabic HS datasets. In the following, a brief description of the corresponding datasets is presented.

4.4.1 Multi-class Hate Speech Classification Data

We use the dataset annotated by (Al-Hassan and Al-Dossari, 2021) to evaluate our proposed classification approach. Table 4.1 shows an overview of the dataset we utilized in our research. The authors utilized the Especial library with a list of hashtags that activate and attract Twitter nasty content to collect the tweets. Because of this, balancing the amount of non-hate and hate tweets is unrealistic and does not reflect the true situation. To maintain a realistic and natural environment, the writers simply picked a list of hashtags that undoubtedly contains both non-hateful and hateful information. After that, two annotators manually annotated the obtained tweets to eliminate any annotator bias. A guide was supplied to the annotators to follow to separate the tweet classifications, resulting in a dataset of 11634 tagged tweets out of 37 k collected tweets. Table 1 shows that the tiny subset of tweets belongs to the racial hate speech category, whereas the large subset belongs to the non-hate category.

4.4.2 Binary and Ternary Hate Speech Classification data

The statistics about the datasets used to evaluate our proposed method for data augmentation are presented in Table 4.2 and described as follows:

	NH	GH	Re	Ra	Se	Total
Number of tweets	8332	1397	722	526	657	11634
Word count	96.9 K	17.2 K	9.2 K	6.6 K	8.3 K	138.3 K
Unique words	29 K	8.7 K	4.9 K	4 K	4.4 K	37.9 K
Average words per tweet	11.6	12.3	12.7	12.7	12.6	11.9

TABLE 4.1: Overview of the datasets used to evaluate our proposed classification approach (NH: Non-Hate speech, Se: Sexism hate speech, Re: Religious hate speech, GH: General hate speech, Ra: Racism hate speech, k denotes one thousand)

- **OSACT** (Sun et al., 2019): This dataset comprises 10K tweets that have been labeled for offensiveness (OFF or NOT OFF) and hate speech (HS or NOT HS).
- **L-HSAB** (Mulki et al., 2019): L-HSAB is a collection of 5,846 Syrian and Lebanese political tweets categorized as normal, abusive, or hateful.
- **T-HSAB** (Haddad et al., 2019): T-HSAB brings together 6,024 Tunisian comments that are classified as normal, abusive, or hateful.
- **GHDS** (Alshaalan and Al-Khalifa, 2020): A total of 9,316 tweets were classed as hateful, abusive, or normal in the dataset (not hateful or abusive).
- **RHS** (Albadi et al., 2019): These include 6000 Arabic tweets, 1000 for each of the six religious groups.
- **MCH** (Alsafari et al., 2020c): This dataset contains 1100 tweets that were gathered utilizing contentious accounts and hashtags.
- **MLMA** (Ousidhoum et al., 2019): This consists of multi-language and multi-aspect, the Arabic tweets include 3353 of hate and non-hate speech.

Dataset	Hate tweets
MLMA	460
GHDS	180
RHS	2759
MCH	1417
OSACT	491
T-HSAB	1078
L-HSAB	468
All data	6853

TABLE 4.2: Overview of the datasets used to evaluate our proposed method for data augmentation.

4.5 Experimental Results and Evaluation

This section presents the experiments conducted in our study. The first experiment demonstrates the choice of hyper-parameter, the second experiment delegated to ensemble learning, the third experiment proves data augmentation, the fourth experiment study demonstrates comparison with state-of-the-art and the last one delegated to comparison with classification algorithms.

4.5.1 Fine-tuning

The first set of our experiment is dedicated to determining the best hyper-parameter. The initial parameters used in our experiment are 2 epochs, 8 batch size, and 1e-5 learning rate.

The experiment presented in Table 4.3 shows the effect of the number of epochs. The F1-score metric is increased when the number of epochs is increased from 2 to 4 by 4.99% in bert-base-arabertv02-twitter, but it is decreased when the number of epochs is further increased. Then, using bert-large-arabertv02-twitter, The F1-score metric is increased from 2 epochs, but it is decreased when the number of epochs is further increased. While when using MARBERT, the F1-score is increased by 0.16 when the number of epochs is increased from 2 epochs to 4 epochs, but it is decreased when the number of epochs is further increased

Epochs	2	3	4	5	10
bert-base-arabertv02-twitter	79.65	84.60	**84.64**	84.28	84.14
bert-large-arabertv02-twitter	**84.59**	83.78	81.74	83.24	79.68
MARBERT	83.98	**84.14**	83.91	83.58	83.11

TABLE 4.3: The effect of the number of epochs on micro F1-score.

The experiment presented in Table 4.4 shows the effect of the number of batch size. The F1-score is increased when the number of batch size is increased from 8 to 16 by 0.02 using bert-base-arabertv02-twitter, but it is decreased when the number of batch size is further increased. However, the F1-score metric is decreased when the number of batch size is further increased from 8 to 64 using bert-large-arabertv02-twitter and MARBET.

Batch size	8	16	32	64
bert-base-arabertv02-twitter	84.64	**84.66**	84.11	83.76
bert-large-arabertv02-twitter	**84.59**	84.08	83.66	81.88
MARBERT	**84.14**	83.87	84.07	82.92

TABLE 4.4: The effect of the number of batch size on micro F1-score.

The experiment presented in Table 4.5 shows the effect of the learning rate. The F1 score is decreased when the value of learning rate is increased from 1 e-5 to 5 e-5.

Learning rate	1e-5	2e-5	3e-5	4e-5	5e-5
bert-base-arabertv02-twitter	**84.66**	84.62	84.53	84.20	83.17
bert-large-arabertv02-twitter	**84.59**	82.25	78.29	76.83	83.66
MARBERT	**84.14**	83.42	81.83	80.55	79.68

TABLE 4.5: The effect of the number of learning rate on micro F1-score.

In the summary, the optimal parameters of the corresponding models are shown in table 4.6.

	Epoch	Batch size	Learning Rate
bert-base-arabertv02-twitter	4	16	1e-5
bert-large-arabertv02-twitter	2	8	1e-5
MARBERT	3	8	1e-5

TABLE 4.6: Optimal parameters of the corresponding pre-trained language model.

4.5.2 Ensemble Learning

The second set of our experiments is dedicated to evaluating the ensemble learning methods (majority voting average voting). Table 4.7 present a comparison between single and ensemble models. It is important to note that ensemble learning models achieve the highest results, yielding weighted-average F1-score results of 85.48 and 85.10 using majority voting and averaged voting, respectively. In summary, the Macro-Averaged, Micro-Averaged F1-scores are reached 72.66%, 85.47% and 85.48%, respectively.

	NH	**Se**	**Re**	**GH**	**Ra**	**Ma**	**Mi**	**W**
BBAT	92.42	70.26	79.89	60.72	50.48	70.76	84.66	84.69
BLAT	92.47	67.60	79.74	60.82	47.81	69.69	84.59	84.46
MARBERT	92.17	68.83	78.42	59.23	50.35	69.80	84.14	84.15
AV	92.60	70.86	81.53	61.21	52.37	71.71	85.08	85.10
MV	**92.73**	**72.27**	**82.56**	**61.89**	**53.85**	**72.66**	**85.47**	**85.48**

TABLE 4.7: The effect of ensemble learning (Ma: Macro-Average, Mi: Micro-Average, W: Weighted-Average, BBAT: bert-base-arabertv02-twitter, BLAT: bert-large-arabertv02-twitter, AV: Average voting, MV: Majority Voting).

4.5.3 Evaluate Data Augmentation Method

The third set of our experiments is delegated to evaluating our proposed method for data augmentation. Firstly, the tweets labeled as religious hate speech in (Albadi et al., 2019) and (Alsafari et al., 2020c) datasets are directly added to the training data. Second, we used semi-supervised learning using the trained model with the datasets presented in Table 4.2. Table 4.8 present the results of the classification task.

Predicted class	**GH**	**Re**	**Se**	**Ra**
Number tweets	1034	950	863	926

TABLE 4.8: Prediction results.

The experiments presented in Table 4.9 shows a comparison of the proposed approach using data augmentation and without data augmentation. However, the F1-score result is increased from 85.48 to 85.65 when the training data is augmented through or proposed method.

	NH	**Se**	**Re**	**GH**	**Ra**	**Ma**	**Mi**	**W**
WDA	**92.73**	72.27	82.56	61.89	53.85	72.66	85.47	85.48
Ours	92.41	**72.58**	**83.50**	**63.27**	**57.25**	**73.80**	**85.60**	**85.65**

TABLE 4.9: The effect of our proposed data augmentation method (WDA: Without Data Augmentation).

4.5.4 Comparison with Existing Data Augmentation Methods

The fourth set of our experiments is dedicated to comparing our proposed methods with the following related as shown in the table 4.10. When compared our proposed method for data augmentation with the following state-of-the-art:

- **W** (Husain, 2020c) which used Up-sampling technique.
- **X** (Alsafari and Sadaoui, 2021) which used semi-supervised learning technique on the unlabeled data.
- **Y** (Liu et al., 2020) which used Generative Pre-trained Transformer technique.
- **Z** (Cao and Lee, 2020) which used GAN technique Generative adversarial networks.

Notably, from the experiment shown in Table 4.8, the F1-score results of different classes is presented as follows:

- **Normal:** The F1-score of the latest state-of-the-art data augmentation methods is ranging from [91.65% – 92.36%] while the F1-score of our proposed data augmentation methods yielded 92.41%.

- **General hate speech:** The achieved results from our proposed data augmentation method is increased by about 3.19% when compared with the latest state-of-the-art.
- **Sexism:** The F1-score of the latest state-of-the-art data augmentation methods ranges from [66.38% – 70.52%], whereas our proposed data augmentation method achieved 72.58%.
- **Racial:** Our proposed data augmentation method increases the F1-score from [44.15% - 50.99%] in comparison with the latest state-of-the-art, which yielded 57.25%.
- **Religious hate speech:** The F1-score of the latest state-of-the-art data augmentation methods is ranging from [78.72% - 79.89%] while our proposed data augmentation method yielded F1-score of 83.50%.

In a summary, the Macro-Averaged, Micro-Averaged, and Micro-Averaged F1-scores are reached 73.80%, 85.60%, and 85.65%, respectively.

	N	S	Re	G	Ra	Ma	Mi	W
WDA	**92.73**	72.27	82.56	61.89	53.85	72.66	85.47	85.48
W	92.04	70.12	79.89	59.45	50.99	70.50	84.20	84.28
X	92.21	70.52	79.42	60.08	50.60	70.57	84.29	84.45
Y	92.36	66.38	78.72	59.48	44.15	68.22	84.08	83.92
Z	91.65	67.50	79.20	57.01	49.07	68.89	83.52	83.43
Ours	92.41	**72.58**	**83.50**	**63.27**	**57.25**	**73.80**	**85.60**	**85.65**

TABLE 4.10: Comparison of our proposed data augmentation method with the latest state-of-the-art methods.

4.5.5 Comparison with Existing Classification Approaches

The last set of our experiment is dedicated to comparing our proposed classification approach with the latest following related works:

- **A** (Al-Hassan and Al-Dossari, 2021), which used Naïve Bayes and N-grams.
- **B** (Husain, 2020b), which used TF–IDF, Bag of Word, and Bagging.
- **C** (Chowdhury et al., 2020b), which use SVM and TF-IDF.
- **D** (Al-Hassan and Al-Dossari, 2021), which learned word embedding use the hybrid CNN-LSTM and using the training data for classification.
- **E** (Alsafari et al., 2020c), which used Multilingual BERT embedding model with CNN.
- **F** (Haddad et al., 2020), which use bidirectional GRU augmented with attention layer and AraVec.

- **G** (Alghanmi et al., 2020), which used AraVec embedding with CNN and AraBERT.
- **H** (Faris et al., 2020), which used hybrid CNN-LSTM and AraVec.
- **I** (Alsafari et al., 2020a), which used AraBERT embedding and ensemble CNNs.
- **J** (Mubarak et al., 2020), which fine-tuned the pre-trained AraBERT language model.

While comparing our proposed approach with the other baselines in Table 4.11, the accuracy rate of our proposition is the highest. This is due to our use of ensemble learning and the the data augmentation method. When compared with CNN, LSTM, current DL hate speech classification models require to huge amount of labeled data to achieve good performance results. In contrast, does not require a huge amount of labeled data to achieve good performance results. When compared to SVM, NB, and Bagging, ours has the benefit of not requiring handcrafted features to be designed and extracted.

Work	NH	Se	Re	GH	Ra	Ma	W
A	0.85	0.21	0.10	0.12	0.09	0.27	0.64
B	0.86	0.42	0.50	0.25	0.24	0.45	0.71
C	0.87	0.41	0.54	0.30	0.29	0.48	0.73
D	0.86	0.43	0.59	0.31	0.25	0.49	0.73
E	0.85	0.42	0.56	0.45	0.20	0.50	0.73
F	0.87	0.50	0.64	0.49	0.27	0.55	0.76
G	0.86	0.44	0.59	0.47	0.22	0.52	0.74
H	0.87	0.37	0.47	0.28	0.26	0.45	0.72
I	0.87	0.49	0.56	0.47	0.24	0.53	0.75
J	0.88	0.49	0.67	0.41	0.37	0.57	0.77
Ensemble	**0.93**	0.72	0.83	0.62	0.54	0.73	0.85
Ours	0.92	**0.73**	**0.84**	**0.63**	**0.57**	**0.74**	**0.86**

TABLE 4.11: Comparison between our proposed classification approach and the latest related works.

4.6 Conclusion

This chapter has presented the experimental results; it has set out the development environment and setup and describes the performance measures and our interesting datasets. Furthermore, the chapter also presented the evaluation results. The next chapter will conclude the thesis and present our future works.

Chapter 5

Conclusion and Future Work

Today, the increase in the spread of HS on social media is a global alarm. Efforts are now taking place to control and avoid HS. This research investigates the impact of ensemble learning that incorporates three different pre-trained Arabic language models with semi-supervised learning built on previously labeled datasets. The leveraged models are used to cover classical Arabic texts and their dialects. Moreover, using pre-trained language model will facilitate the process of learning content representations. Experiments results show that: (1) ensemble learning based on pre-trained language models outperforms existing related works; (2) Our proposed data augmentation improves the accuracy results of hate speech detection from Arabic tweets and outperforms existing related works.

As future works, we plan to follow numerous directions. First, we want to focus on the contextual embedding model and try to modify its vocabulary to support the HS detection task. Second, we plan to address the task of classifying HS in the Algerian dialect.

From a research perspective, we will leverage our proposed systems to classify and analyze Arabic Twitter discussions to determine the extent of HS conversations with public discourse and to understand how their sophistication and capabilities evolve over the time.

Bibliography

Abdellatif, Mohamed and Ahmed Elgammal (2020). "Offensive language detection in Arabic using ULMFiT". In: *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pp. 82–85.

Abdul-Mageed, Muhammad, AbdelRahim Elmadany, and El Moatez Billah Nagoudi (2020). "ARBERT & MARBERT: deep bidirectional transformers for Arabic". In: *arXiv preprint arXiv:2101.01785*.

Abuzayed, Abeer and Tamer Elsayed (2020). "Quick and simple approach for detecting hate speech in Arabic tweets". In: *Proceedings of the 4th workshop on open-source Arabic Corpora and processing tools, with a shared task on offensive language detection*, pp. 109–114.

Al-Hassan, Areej and Hmood Al-Dossari (2021). "Detection of hate speech in Arabic tweets using deep learning". In: *Multimedia Systems*, pp. 1–12.

Albadi, Nuha, Maram Kurdi, and Shivakant Mishra (2019). "Investigating the effect of combining GRU neural networks with handcrafted features for religious hatred detection on Arabic Twitter space". In: *Social Network Analysis and Mining* 9.1, pp. 1–19.

Aldjanabi, Wassen, Abdelghani Dahou, Mohammed AA Al-qaness, Mohamed Abd Elaziz, Ahmed Mohamed Helmi, and Robertas Damaševičius (2021). "Arabic Offensive and Hate Speech Detection Using a Cross-Corpora Multi-Task Learning Model". In: *Informatics*. Vol. 8. 4. Multidisciplinary Digital Publishing Institute, p. 69.

Alghanmi, Israa, Luis Espinosa-Anke, and Steven Schockaert (2020). "Combining BERT with static word embeddings for categorizing social media". In.

Alharbi, Abdullah I and Mark Lee (2020). "Combining character and word embeddings for the detection of offensive language in Arabic". In: *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pp. 91–96.

Alsafari, Safa and Samira Sadaoui (2021). "Semi-Supervised Self-Training of Hate and Offensive Speech from Social Media". In: *Applied Artificial Intelligence*, pp. 1–25.

Alsafari, Safa, Samira Sadaoui, and Malek Mouhoub (2020a). "Deep learning ensembles for hate speech detection". In: *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, pp. 526–531.

— (2020b). "Effect of Word Embedding Models on Hate and Offensive Speech Detection". In: *arXiv preprint arXiv:2012.07534*.

— (2020c). "Hate and offensive speech detection on Arabic social media". In: *Online Social Networks and Media* 19, p. 100096.

Alshaalan, Raghad and Hend Al-Khalifa (2020). "Hate speech detection in saudi twittersphere: A deep learning approach". In: *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, pp. 12–23.

Alshutayri, Areej and Eric Atwell (2018). "Creating an Arabic dialect text corpus by exploring Twitter, Facebook, and online newspapers". In: *OSACT 3 Proceedings*. LREC.

Amiri, Khitem, Yaakoub Boualleg, and Mohamed Farah (2020). "Radiometric indices-based spectro-spatial approach for hyperspectral image classification". In: *The Egyptian Journal of Remote Sensing and Space Science* 23.3, pp. 287–301.

Antoun, Wissam, Fady Baly, and Hazem Hajj (2020). "AraBERT: Transformer-based Model for Arabic Language Understanding". In: *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pp. 9–15.

Barhoumi, Amira, Nathalie Camelin, Chafik Aloulou, Yannick Estève, and Lamia Hadrich Belguith (2019). "An empirical evaluation of arabic-specific embeddings for sentiment analysis". In: *International Conference on Arabic Language Processing*. Springer, pp. 34–48.

Boualleg, Yaakoub and Mohamed Farah (2018). "Enhanced interactive remote sensing image retrieval with scene classification convolutional neural networks model". In: *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, pp. 4748–4751.

Boualleg, Yaakoub, Mohamed Farah, and Imed Riadh Farah (2019). "Remote sensing scene classification using convolutional features and deep forest classifier". In: *IEEE Geoscience and Remote Sensing Letters* 16.12, pp. 1944–1948.

— (2020). "TLDCNN: A triplet low dimensional convolutional neural networks for high-resolution remote sensing image retrieval". In: *2020 Mediterranean and Middle-East Geoscience and Remote Sensing Symposium (M2GARSS)*. IEEE, pp. 13–16.

Cao, Rui and Roy Ka-Wei Lee (2020). "Hategan: Adversarial generative-based data augmentation for hate speech detection". In: *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6327–6338.

Chowdhury, Shammur Absar, Hamdy Mubarak, Ahmed Abdelali, Soon-gyo Jung, Bernard J Jansen, and Joni Salminen (2020a). "A multi-platform Arabic news comment dataset for offensive language detection". In: *Proceedings of the 12th Language Resources and Evaluation Conference*, pp. 6203–6212.

— (2020b). "A multi-platform Arabic news comment dataset for offensive language detection". In: *Proceedings of the 12th Language Resources and Evaluation Conference*, pp. 6203–6212.

Daouadi, Kheir Eddine, Rim Zghal Rebaï, and Ikram Amous (2018a). "Organization vs. Individual: Twitter User classification." In.

— (2019a). "Bot detection on online social networks using deep forest". In: *Computer science on-line conference*. Springer, pp. 307–315.

— (2019b). "Organization, bot, or human: Towards an efficient twitter user classification". In: *Computación y Sistemas* 23.2, pp. 273–279.

Daouadi, Kheir Eddine, Rim Zghal Rebaï, and Ikram Amous (2020). "Real-Time Bot Detection from Twitter Using the Twitterbot+ Framework." In: *J. Univers. Comput. Sci.* 26.4, pp. 496–507.

— (2021). "Optimizing semantic deep forest for tweet topic classification". In: *Information Systems* 101, p. 101801.

Daouadi, Kheir Eddine, Rim Zghal Rebaï, and Ikram Amous (2018b). "Towards a Statistical Approach for User Classification in Twitter". In: *International Conference on Machine Learning for Networking*. Springer, pp. 33–43.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805*.

Duwairi, Rehab, Amena Hayajneh, and Muhannad Quwaider (2021). "A deep learning framework for automatic detection of hate speech embedded in Arabic tweets". In: *Arabian Journal for Science and Engineering* 46.4, pp. 4001–4014.

Elmadany, AbdelRahim, Chiyu Zhang, Muhammad Abdul-Mageed, and Azadeh Hashemi (2020). "Leveraging affective bidirectional transformers for offensive language detection". In: *arXiv preprint arXiv:2006.01266*.

Farha, Ibrahim Abu and Walid Magdy (2020). "Multitask learning for Arabic offensive language and hate-speech detection". In: *Proceedings of the 4th workshop on open-source Arabic corpora and processing tools, with a shared task on offensive language detection*, pp. 86–90.

Faris, Hossam, Ibrahim Aljarah, Maria Habib, and Pedro A Castillo (2020). "Hate Speech Detection using Word Embedding and Deep Learning in the Arabic Language Context." In: *ICPRAM*, pp. 453–460.

Founta, Antigoni Maria, Constantinos Djouvas, Despoina Chatzakou, Ilias Leontiadis, Jeremy Blackburn, Gianluca Stringhini, Athena Vakali, Michael Sirivianos, and Nicolas Kourtellis (2018). "Large scale crowdsourcing and characterization of twitter abusive behavior". In: *Twelfth International AAAI Conference on Web and Social Media*.

Ghahramani, Zoubin (2007). *Proceedings of the 24th international conference on Machine learning*.

Glorot, Xavier, Antoine Bordes, and Yoshua Bengio (2011). "Domain adaptation for large-scale sentiment classification: A deep learning approach". In: *ICML*.

Haddad, Bushr, Zoher Orabe, Anas Al-Abood, and Nada Ghneim (2020). "Arabic offensive language detection with attention-based deep neural networks". In: *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pp. 76–81.

Haddad, Hatem, Hala Mulki, and Asma Oueslati (2019). "T-HSAB: A Tunisian Hate Speech and Abusive Dataset". In: *Arabic Language Processing: From Theory to Practice*. Ed. by Kamel Smaïli. Cham: Springer International Publishing, pp. 251–263. ISBN: 978-3-030-32959-4.

Hafdhellaoui, Sabrine, Yaakoub Boualleg, and Mohamed Farah (2019). "Collaborative clustering approach based on dempster-shafer theory for bag-of-visual-words codebook generation". In: *Canadian Conference on Artificial Intelligence*. Springer, pp. 263–273.

Hassan, Sabit, Younes Samih, Hamdy Mubarak, Ahmed Abdelali, Ammar Rashed, and Shammur Absar Chowdhury (2020). "ALT submission for OSACT shared task on offensive language detection". In: *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pp. 61–65.

Hinduja, Sameer and Justin W Patchin (2010). "Bullying, cyberbullying, and suicide". In: *Archives of suicide research* 14.3, pp. 206–221.

Husain, Fatemah (2020a). "Arabic offensive language detection using machine learning and ensemble machine learning approaches". In: *arXiv preprint arXiv:2005.08946*.

— (2020b). "Arabic offensive language detection using machine learning and ensemble machine learning approaches". In: *arXiv preprint arXiv:2005.08946*.

— (2020c). "OSACT4 shared task on offensive language detection: Intensive preprocessing-based approach". In: *arXiv preprint arXiv:2005.07297*.

Husain, Fatemah, Jooyeon Lee, Sam Henry, and Ozlem Uzuner (2020). "Salam-NET at SemEval-2020 Task 12: Deep learning approach for Arabic offensive language detection". In: *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pp. 2133–2139.

Irfan, Rizwana, Christine K King, Daniel Grages, Sam Ewen, Samee U Khan, Sajjad A Madani, Joanna Kolodziej, Lizhe Wang, Dan Chen, Ammar Rayes, et al. (2015). "A survey on text mining in social networks". In: *The Knowledge Engineering Review* 30.2, pp. 157–170.

Jones, Karen Sparck (1972). "A statistical interpretation of term specificity and its application in retrieval". In: *Journal of documentation*.

Joulin, Armand, Édouard Grave, Piotr Bojanowski, and Tomáš Mikolov (2017). "Bag of Tricks for Efficient Text Classification". In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pp. 427–431.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems* 25.

Laatar, Rim, Chafik Aloulou, and Lamia Hadrich Belguith (2017). "Word Sense Disambiguation of Arabic Language with Word Embeddings as Part of the Creation of a Historical Dictionary." In: *LPKM*.

Leskovec, Jure, Anand Rajaraman, and Jeffrey David Ullman (2020). *Mining of massive data sets*. Cambridge university press.

Liu, Ruibo, Guangxuan Xu, and Soroush Vosoughi (2020). "Enhanced offensive language detection through data augmentation". In: *arXiv preprint arXiv:2012.02954*.

Luhn, Hans Peter (1957). "A statistical approach to mechanized encoding and searching of literary information". In: *IBM Journal of research and development* 1.4, pp. 309–317.

McCallum, Andrew, Kamal Nigam, et al. (1998). "A comparison of event models for naive bayes text classification". In: *AAAI-98 workshop on learning for text categorization*. Vol. 752. 1. Citeseer, pp. 41–48.

Mikolov, Tomas, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur (2010). "Recurrent neural network based language model." In: *Interspeech*. Vol. 2. 3. Makuhari, pp. 1045–1048.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean (2013). "Distributed representations of words and phrases and their compositionality". In: *Advances in neural information processing systems* 26.

Mubarak, Hamdy, Kareem Darwish, and Walid Magdy (2017). "Abusive language detection on Arabic social media". In: *Proceedings of the first workshop on abusive language online*, pp. 52–56.

Mubarak, Hamdy, Ammar Rashed, Kareem Darwish, Younes Samih, and Ahmed Abdelali (2020). "Arabic offensive language on twitter: Analysis and experiments". In: *arXiv preprint arXiv:2004.02192*.

Mulki, Hala, Hatem Haddad, Chedi Bechikh Ali, and Halima Alshabani (Aug. 2019). "L-HSAB: A Levantine Twitter Dataset for Hate Speech and Abusive Language". In: *Proceedings of the Third Workshop on Abusive Language Online*. Florence, Italy: Association for Computational Linguistics, pp. 111–118. DOI: 10.18653/v1/W19-3512. URL: https://aclanthology.org/W19-3512.

Ousidhoum, Nedjma, Zizheng Lin, Hongming Zhang, Yangqiu Song, and Dit-Yan Yeung (2019). "Multilingual and multi-aspect hate speech analysis". In: *arXiv preprint arXiv:1908.11049*.

Pennington, Jeffrey, Richard Socher, and Christopher D Manning (2014). "Glove: Global vectors for word representation". In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543.

Rachid, Benaissa Azzeddine, Harbaoui Azza, and Hajjami Henda Ben Ghezala (2020). "Classification of cyberbullying text in arabic". In: *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, pp. 1–7.

Salem, Fadi (2017). "Social media and the internet of things towards data-driven policymaking in the Arab world: potential, limits and concerns". In.

Sousa Pereira Amorim, Brunna de, André Luiz Firmino Alves, Maxwell Guimarães de Oliveira, and Cláudio de Souza Baptista (2018). "Using Supervised Classification to Detect Political Tweets with Political Content". In: *Proceedings of the 24th Brazilian Symposium on Multimedia and the Web*, pp. 245–252.

Sun, Chi, Xipeng Qiu, Yige Xu, and Xuanjing Huang (2019). "How to fine-tune bert for text classification?" In: *China national conference on Chinese computational linguistics*. Springer, pp. 194–206.

Vadivukarassi, M, N Puviarasan, and P Aruna (2018). "A comparison of supervised machine learning approaches for categorized tweets". In: *International Conference on Intelligent Data Communication Technologies and Internet of Things*. Springer, pp. 422–430.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, ukasz Kaiser, and Illia Polosukhin (2017). "Attention is all you need". In: *Advances in neural information processing systems* 30.

Wu, Pengcheng and Thomas G Dietterich (2004). "Improving SVM accuracy by training on auxiliary data sources". In: *Proceedings of the twenty-first international conference on Machine learning*, p. 110.

Yang, Xiao, Craig Macdonald, and Iadh Ounis (2018). "Using word embeddings in twitter election classification". In: *Information Retrieval Journal* 21.2, pp. 183–207.