HAL
open science

Hate speech and offensive language detection using transfer learning approaches

Marzieh Mozafari

To cite this version:

Marzieh Mozafari. Hate speech and offensive language detection using transfer learning approaches. Document and Text Processing. Institut Polytechnique de Paris, 2021. English. NNT : 2021IPPAS007 . tel-03276023

HAL Id: tel-03276023

https://theses.hal.science/tel-03276023v1

Submitted on 1 Jul 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

INSTITUT POLYTECHNIQUE DE PARIS

NNT : 2021IPPAS007

Thèse de doctorat

Hate Speech and Offensive Language Detection using Transfer Learning Approaches

Thèse de doctorat de l'Institut Polytechnique de Paris
préparée à Télécom SudParis

École doctorale n°626 Ecole doctorale de l'Institut Polytechnique de Paris (ED IP Paris)
Spécialité de doctorat : Informatique

Thèse présentée et soutenue à Évry, le 28/05/2021, par

MARZIEH MOZAFARI

Composition du Jury :

Daqing Zhang Directeur d'études, IMT, Télécom SudParis - France	Président
Gabriella Pasi Professeure, University of Milano-Bicocca - Italy	Rapporteuse
Ioan Marius Bilasco Maître de Conférences, Lille 1 University - France	Rapporteur
Elena Cabrio Assistant Professor, Université Côte d'Azur - France	Examinateur
Christophe Cerisara Chargé de recherche, CNRS - France	Examinateur
Daqing Zhang Directeur d'études, IMT, Télécom SudParis - France	Examinateur
Noel Crespi Professeur, IMT, Télécom SudParis - France	Directeur de thèse
Reza Farahbakhsh Maître de Conférences Associé, IMT, Télécom SudParis - France	Co-directeur de thèse

Doctor of Philosophy (PhD) Thesis
Institut-Mines Télécom, Télécom SudParis
& Institut Polytechnique de Paris (IP Paris)

Specialization

Computer Science

presented by

Marzieh Mozafari

Hate Speech and Offensive Language Detection using Transfer Learning Approaches

Committee:

Gabriella Pasi	Reviewer	Professor, University of Milano-Bicocca - Italy
Ioan Marius Bilasco	Reviewer	Associate Professor, Lille 1 University - France
Elena Cabrio	Examiner	Assistant Professor, Université Côte d'Azur - France
Christophe Cerisara	Examiner	Researcher, CNRS - France
Daqing Zhang	Examiner	Professor, IMT, Telecom SudParis - France
Noel Crespi	Advisor	Professor, IMT, Telecom SudParis - France
Reza Farahbakhsh	Co-supervisor	Adjunct Assistant Professor, IMT, Telecom SudParis - France

Thèse de Doctorat (PhD) de Institut-Mines Télécom, Télécom SudParis et l'Institut Polytechnique de Paris (IP Paris)

Spécialité

Informatique

présentée par

Marzieh Mozafari

Détection du Discours de Haine et du Langage Offensant utilisant des Approches de Transfer Learning

Jury composé de :

Gabriella Pasi	Rapporteuse	Professeure, University of Milano-Bicocca - Italy
Ioan Marius Bilasco	Rapporteur	Maître de Conférences, Lille 1 University - France
Elena Cabrio	Examinateur	Assistant Professor, Université Côte d'Azur - France
Christophe Cerisara	Examinateur	Chargé de recherche, CNRS - France
Daqing Zhang	Examinateur	Professeur, IMT, Télécom SudParis - France
Noel Crespi	Directeur de thèse	Professeur, IMT, Télécom SudParis - France
Reza Farahbakhsh	Co-directeur de thèse	Maître de Conférences Associé, IMT, Télécom SudParis - France

Dedication

To
two of my best friends Dr. Ali Jalilvand and Dr. Ardavan Afshar who passed away during their PhD and could not defend their thesis, unfortunately...

"No one is born hating another person because of the color of his skin, or his background, or his religion. People must learn to hate, and if they can learn to hate, they can be taught to love, for love comes more naturally to the human heart than its opposite."

Nelson Mandela

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my supervisor, Prof. Noel Crespi, for his continuous support, patience, friendship, insights, as well as all the guidance and help he provided throughout my research. Thank you so much for believing in me throughout this journey and for giving me the freedom to pursue my interests and follow my curiosity. Being a part of your team, is an honor for me.

I would like to thank my co-supervisor, Dr. Reza Farahbakhsh, for his technical support, motivation, and fruitful discussions which we had. Thank you so much for being always willing and enthusiastic to assist me in any way at any time, and for providing me advices on every entangled situation.

I would like to extend my sincere thanks to my thesis reviewers, Prof. Gabriella Pasi and Dr. Ioan Marius Bilasco, who patiently read this dissertation and provided invaluable comments and suggestions. A special thanks to Dr. Elena Cabrio, Dr. Christophe Cerisara, and Prof. Daqing Zhang for being the part of my jury as examiners for my thesis defense.

During my PhD, I had a great chance to be involved in an international project, ITEA PAPUD, and I would like to address special thanks for people who were involved in and provided me an invaluable experience.

My special thanks to all the lovely team members of the Data Intelligence and Communication Engineering Lab at TSP, especially Praboda, Shanay, Samin, Yasir, Faraz, Koosha, and Hamza for wonderful times we shared. You were always there with a word of encouragement or listening ear and you provided me all the support and friendship that I needed. I also had great pleasure of working with Prof. Roberto Minerva from whom I learned new ways of thinking and how to best contribute to a project. Special thanks go to the wonderful administrative staff in TSP, Valerie Mateus and Veronique Guy, who were so helpful and friendly and always helped me in dealing with PhD administrative tasks.

My deep and sincere gratitude to my best friend, Sahar, who is a wonderful, supportive, caring, and generous friend with whom I shared my happiness and sadness. For the past three years of my PhD, she supported me a lot and had to put up with my stresses and moans. So glad that I have her.

My profound love, respect and thank goes to my family whom I owe a great deal. I deeply thank my parents, Ahmad and Fatemeh, for their unrequited love, unconditional trust, timely encouragement, and endless patience. It was their love that empowered me to break my limits and experience the life freely and fearlessly. I would like to thank my beautiful sister, Razieh, for her intelligence and passion for education which always inspired me. Although she left us very soon (during my PhD) and it is an insurmountable event for me, she will be remembered forever and I always will live with the love she left behind. I am

also grateful to my siblings, Zahra, Vahid, and Peyman who are a proof of the universe's superiority in every sense. You have been generous with your love and encouragement despite the long distance between us, and I could not have asked for anything better.

Last but never least, my biggest thanks to my husband, Kourosh, whom I am so lucky to have in my life. When I struggled during my research, he not only supported me by his technical advices but also was there to pick me up and encouraged me to keep trying. Kourosh, you are my best friend, my confidant, my support, and there are not words to express my gratitude for having you in my life. I could not be able to finish this work without your support, and this dissertation stands as a testament to your unconditional love and encouragement.

Marzieh Mozafari
28th May 2021

Abstract

The great promise of social media platforms (e.g., Twitter and Facebook) is to provide a safe place for users to communicate their opinions and share information. However, concerns are growing that they enable abusive behaviors, e.g., threatening or harassing other users, cyberbullying, hate speech, racial and sexual discrimination, as well. In this thesis, we focus on hate speech as one of the most concerning phenomena in online social media.

Given the high progression of online hate speech and its severe negative effects, institutions, social media platforms, and researchers have been trying to react as quickly as possible. The recent advancements in Natural Language Processing (NLP) and Machine Learning (ML) algorithms can be adapted to develop automatic methods for hate speech detection in this area.

The aim of this thesis is to investigate the problem of hate speech and offensive language detection in social media, where we define hate speech as any communication criticizing a person or a group based on some characteristics, e.g., gender, sexual orientation, nationality, religion, race. We propose different approaches in which we adapt advanced Transfer Learning (TL) models and NLP techniques to detect hate speech and offensive content automatically, in a monolingual and multilingual fashion.

In the first contribution, we only focus on English language. Firstly, we analyze user-generated textual content in Facebook to gain a brief insight into the type of content by introducing a new framework being able to categorize contents in terms of topical similarity based on different features, namely lexical, topical, and semantical features. Furthermore, using the Perspective API from Google, we measure and analyze the toxicity of the content. Secondly, we propose a TL approach for identification of hate speech by employing a combination of the unsupervised pre-trained model BERT (Bidirectional Encoder Representations from Transformers) and new supervised fine-tuning strategies. Finally, we investigate the effect of unintended bias in our pre-trained BERT-based model and propose a new generalization mechanism in training data by reweighting samples and then changing the fine-tuning strategies in terms of the loss function to mitigate the racial bias propagated through the model. To evaluate the proposed models, we use three publicly available datasets from Twitter.

In the second contribution, we consider a multilingual setting where we focus on low-resource languages in which there is no or few labeled data available. First, we present the first corpus of Persian offensive language consisting of 6 000 microblogs from Twitter to deal with offensive language detection in Persian as a low-resource language in this domain. After annotating the corpus, we perform extensive experiments to investigate the performance of transformer-based monolingual and multilingual pre-trained language models (e.g., ParsBERT, mBERT, XLM-RoBERTa) in the downstream task. Furthermore, we propose an ensemble model to boost the performance of our model. Then, we expand our study into a cross-lingual few-shot learning problem and we adapt a meta learning-based approach to study the problem of few-shot hate speech and offensive language detection

in low-resource languages that will allow hateful or offensive content to be predicted by only observing a few labeled data items in a specific target language. To evaluate the proposed model, we use diverse collections of different publicly available corpora, comprising 15 datasets across 8 languages for hate speech and 6 datasets across 6 languages for offensive language. To the best of the author's knowledge, there has been an insignificant number of attempts to use meta learning approaches on hate speech detection tasks.

Keywords

Hate Speech, Offensive Language, Transfer Learning, BERT, XLM-RoBERTa, Deep Learnig, Cross Lingual Text Classification, Few-shot Learning, Meta Learning, Social Media, Twitter

Résumé

Une des promesses des plateformes de réseaux sociaux (comme Twitter et Facebook) est de fournir un endroit sûr pour que les utilisateurs puissent partager leurs opinions et des informations. Cependant, l'augmentation des comportements abusifs, comme le harcèlement en ligne ou la présence de discours de haine, est bien réelle. Dans cette thèse, nous nous concentrons sur le discours de haine, l'un des phénomènes les plus préoccupants concernant les réseaux sociaux.

Compte tenu de sa forte progression et de ses graves effets négatifs, les institutions, les plateformes de réseaux sociaux et les chercheurs ont tenté de réagir le plus rapidement possible. Les progrès récents des algorithmes de traitement automatique du langage naturel (NLP) et d'apprentissage automatique (ML) peuvent être adaptés pour développer des méthodes automatiques de détection des discours de haine dans ce domaine.

Le but de cette thèse est d'étudier le problème du discours de haine et de la détection des propos injurieux dans les réseaux sociaux. Nous proposons différentes approches dans lesquelles nous adaptons des modèles avancés d'apprentissage par transfert (TL) et des techniques de NLP pour détecter automatiquement les discours de haine et les contenus injurieux, de manière monolingue et multilingue.

La première contribution concerne uniquement la langue anglaise. Tout d'abord, nous analysons le contenu textuel généré par les utilisateurs sur Facebook en introduisant un nouveau cadre capable de catégoriser le contenu en termes de similarité basée sur différentes caractéristiques, à savoir les caractéristiques lexicales, topiques et sémantiques. En outre, en utilisant l'API Perspective de Google, nous mesurons et analysons la toxicité du contenu. Ensuite, nous proposons une approche TL pour l'identification des discours de haine en utilisant une combinaison du modèle non supervisé pré-entraîné BERT (Bidirectional Encoder Representations from Transformers) et de nouvelles stratégies supervisées de réglage fin. Enfin, nous étudions l'effet du biais involontaire dans notre modèle pré-entraîné BERT et proposons un nouveau mécanisme de généralisation dans les données d'entraînement en repondérant les échantillons puis en changeant les stratégies de réglage fin en termes de fonction de perte pour atténuer le biais racial propagé par le modèle. Pour évaluer les modèles proposés, nous utilisons trois datasets publics provenant de Twitter.

Dans la deuxième contribution, nous considérons un cadre multilingue où nous nous concentrons sur les langues à faibles ressources dans lesquelles il n'y a pas ou peu de données annotées disponibles. Tout d'abord, nous présentons le premier corpus de langage injurieux en persan, composé de 6 000 messages de micro-blogs provenant de Twitter, afin d'étudier la détection du langage injurieux. Après avoir annoté le corpus, nous réalisons étudions les performances des modèles de langages pré-entraînés monolingues et multilingues basés sur des transformeurs (par exemple, ParsBERT, mBERT, XLM-RoBERTa) dans la tâche en aval. De plus, nous proposons un modèle d'ensemble pour améliorer la performance de notre modèle. Enfin, nous étendons notre étude à un problème d'apprentissage multilingue de type few-shot, où nous disposons de quelques données annotées dans la langue cible.

Nous adaptons une approche basée sur le méta-apprentissage pour étudier le problème de la détection des discours de haine et de langage offensant de type few-shot dans les langues à faibles ressources, qui permettra de prédire le contenu haineux ou offensant en n'observant que quelques éléments de données étiquetés dans une langue cible spécifique. Pour évaluer les modèles proposés, nous utilisons diverses collections de différents corpus accessibles au public, comprenant 15 datasets dans 8 langues pour le discours de haine et 6 datasets dans 6 langues pour langage offensant. Au meilleur de la connaissance de l'auteur, il y a eu un nombre insignifiant de tentatives d'utilisation des modèles de méta-apprentissage sur les tâches de détection des discours de haine.

Mots-clés

Discours de haine, Langage offensant, Apprentissage par transfert, BERT, XLM-RoBERTa, l'apprentissage en profondeur, Classification interlinguistique des textes, Few-shot learning, Meta learning, Réseaux sociaux, Twitter

Table of contents

1 Introduction **19**
1.1 Motivation . . . 20
1.2 Objectives and Contributions of the Thesis . . . 21
1.3 Publications List . . . 24
1.4 Relationship of Publications with Contributions . . . 25
1.5 Outline of the Thesis . . . 25
1.6 Ethical Considerations . . . 26

2 Background and Related Technologies **27**
2.1 Overview . . . 28
2.2 What Is Hate Speech? . . . 28
2.3 Major Advancements in NLP . . . 30
2.3.1 Language Modeling . . . 30
2.3.2 Text Representations in NLP . . . 31
2.3.3 Sequence to Sequence Models . . . 32
2.3.4 Attentions . . . 33
2.3.5 Transformers . . . 33
2.4 Transfer Learning in NLP . . . 33
2.4.1 Generalized Language Models . . . 34
2.4.2 Multilingual Language Models . . . 36
2.4.3 Meta Learning . . . 37
2.4.4 Few-Shot Learning . . . 38
2.5 Automatic Detection of Hate Speech . . . 38
2.5.1 Machine Learning Approach . . . 39
2.5.2 Deep Learning Approach . . . 40
2.6 Summary and Conclusion . . . 40

3 Social Media Content Analysis **43**
3.1 Introduction . . . 44
3.2 Related Work . . . 46
3.3 Methodology and Framework . . . 47

3.3.1 Features Description . . . 48
3.4 Experiments and Results . . . 52
3.4.1 Dataset Description . . . 53
3.4.2 Gold Standard Annotation . . . 53
3.4.3 Results . . . 55
3.4.4 Case Study . . . 58
3.5 Conclusion . . . 65

4 Monolingual Hate Speech Detection 67
4.1 Overview . . . 68
4.2 A BERT-Based Transfer Learning Approach for Hate Speech Detection . . . 68
4.2.1 Introduction . . . 68
4.2.2 Related Work . . . 70
4.2.3 BERT-Based Hate Speech Detection Module . . . 72
4.2.3.1 BERT . . . 72
4.2.3.2 Fine-Tuning Strategies . . . 73
4.2.4 Experiments and Results . . . 75
4.2.4.1 Dataset Description . . . 75
4.2.4.2 Implementation . . . 77
4.2.4.3 Results . . . 77
4.2.4.4 Performance Evaluation with a Limited Amount of Training Data . . . 79
4.2.4.5 BERT Embeddings Analysis . . . 81
4.2.4.6 Error analysis . . . 85
4.2.5 Conclusion . . . 87
4.3 Racial Bias Mitigation in Social Media based on BERT Model . . . 87
4.3.1 Introduction . . . 87
4.3.2 Related Work . . . 88
4.3.3 Bias Mitigation Module . . . 90
4.3.3.1 Towards Unbiased Training Data . . . 90
4.3.3.2 Re-weighting Mechanism . . . 92
4.3.3.3 Scrutinizing Bias Mitigation Mechanism . . . 93
4.3.4 Discussion and Challenges . . . 97
4.3.5 Conclusion . . . 99
4.4 Summary and Discussion . . . 100

5 Multilingual Hate Speech Detection 101
5.1 Overview . . . 102
5.2 Offensive Language Detection in Low Resource Languages: a use case of Persian language . . . 102
5.2.1 Introduction . . . 102
5.2.2 Related Work . . . 104
5.2.3 Dataset Description . . . 106
5.2.3.1 Data Collection . . . 106

5.2.3.2 The Ethical Consideration . . . 107
5.2.3.3 Data Annotation Schema . . . 108
5.2.3.4 Annotation Process . . . 109
5.2.4 Methodology . . . 109
5.2.4.1 Classical Machine Learning Models . . . 111
5.2.4.2 Deep Learning Models . . . 112
5.2.4.3 Monolingual and Multilingual Transformer-Based Networks 113
5.2.4.4 Stacking Ensemble Model . . . 115
5.2.5 Experiments and Results . . . 116
5.2.5.1 Training Procedure . . . 117
5.2.5.2 Single Models Results . . . 118
5.2.5.3 Ensemble Model Results . . . 121
5.2.6 Conclusion . . . 123
5.3 Cross-Lingual Hate Speech Detection using Meta Learning . . . 124
5.3.1 Introduction . . . 124
5.3.2 Related work . . . 125
5.3.3 Methodology . . . 128
5.3.3.1 Meta Learning . . . 129
5.3.3.2 Model-Agnostic Meta-Learning . . . 130
5.3.3.3 Proto-MAML . . . 132
5.3.3.4 Base-Learner Model . . . 133
5.3.4 Dataset Description . . . 133
5.3.4.1 Hate Speech Data . . . 133
5.3.4.2 Offensive Language Data . . . 134
5.3.5 Experiments and Results . . . 134
5.3.5.1 Training Models . . . 134
5.3.5.2 Training Setup and Implementation . . . 136
5.3.5.3 Results and discussions . . . 137
5.3.6 Conclusion . . . 142
5.4 Summary and Discussion . . . 143

6 Conclusion and Future Work 145
6.1 Conclusion . . . 146
6.1.1 Summary and Insights of Contributions . . . 146
6.2 Future Work and Challenges . . . 149

References 150

List of figures 163

List of tables 166

A Appendix **169**
A.1 Hate Speech Datasets 169
A.2 Offensive Language Datasets 172

Chapter 1

Introduction

Contents

1.1 Motivation 20
1.2 Objectives and Contributions of the Thesis 21
1.3 Publications List 24
1.4 Relationship of Publications with Contributions 25
1.5 Outline of the Thesis 25
1.6 Ethical Considerations 26

1.1 Motivation

Nowadays, most of the people around the world are increasingly using social networking platforms such as Twitter, Facebook, YouTube, etc. to communicate their opinions and share information. Although online interactions among users can enable constructive and insightful conversations, they may potentially be a place for disseminating verbal abuse as well; witch triggers some negative outcomes including hate speech, cyberbullying, harassment, offensive language, etc. In addition, due to the user anonymity in online platforms, people have more tendency to dispose their racist, misogynist, homophobic attitudes toward minority groups, e.g. immigrants, LGBT, Muslims. The more people use online platforms the more hatred, abusive and toxic content might be inclined toward them and thus having a mechanism for detecting and mitigating such content is crucial.

Online hate speech that is defined as "a speech advocating for incitement to harm, discrimination, hostility, or violence directed toward individuals or groups based on their race, religion, sexual orientation or gender" is linked to the different types of violence in the real word recently [1]. Since 2015, social scientists have been actively investigating the impact of online hatred and offensive content on real world acts of violence. For example, there was a correlation between anti-refugee Facebook posts written by the far-right Alternative for Germany party and attacks on refugees in 2016[1], or a significant increase in hate speech towards Muslim communities following the Manchester Arena bombing and the London Bridge attack in 2017[2]. Very recently on 16 September 2020, online users launched a movement against the hateful contents in Facebook and Instagram by pausing social posts on Stop Hate For Profit day[3] that highlights the important role of online platforms in this issue. Apart from that, there is different periodical survey and reporting activities from anti-hate organizations such as Anti-Defamation League (ADL)[4] which works to stop discrimination against Jewish people or the National Association for the Advancement of Colored People (NAACP)[5] which works to fight racial discrimination in the United State. On the other hand, the online platforms such as Facebook, Instagram[6], and Twitter[7] have a Community Standards Enforcement Report to report their statistics related to the volume of hate speech they detected and the actions they performed.

These examples among many others declare the proliferation of hate speech among so-

[1] https://www.cfr.org/backgrounder/hate-speech-social-media-global-comparisons
[2] https://www.theguardian.com/society/2017/jun/20/anti-muslim-hate-surges-after-manchester-and-london-bridge-
[3] https://www.bbc.co.uk/newsround/54174625
[4] https://www.adl.org/onlineharassment
[5] https://www.naacp.org/campaigns/no-hate/
[6] https://transparency.facebook.com/community-standards-enforcement\#hate-speech
[7] https://blog.twitter.com/en_us/topics/company/2019/twitter-transparency-report-2019.html

cial media platforms and the necessity of tackling it all. Most of the popular social media platforms (e.g., Facebook and Twitter) are employing several techniques to identify such content by: i) leveraging collective knowledge and asking users to report what they recognize as hate through the platform; ii) using basic approaches like searching for special keywords; iii) manual inspection by experts; and iv) using Artificial Intelligence-based models. However, due to the diverse, subtle, and complex nature of this problem it is still far from being solved.

In the latest years, due to the advancements in Natural Language Processing (NLP) techniques, there has been a huge interest in adapting NLP techniques together with Machine Learning (ML) and Deep Learning (DL) approaches to address the problem of hate speech detection at scale by developing automatic detection models. Early attempts have used classical ML models with different feature engineering techniques [2,3]. However, these features were generally task-specific and they required domain expertises. Later on, deep neural networks have become an interesting model of choice for identification of hate speech and offensive language [4, 5], however, they need a huge number of labeled data in a specific task. Due to the lack of enough labeled data related to hate speech detection tasks, especially in low-resource languages, the usage of deep neural networks lonely is restricted. Transfer Learning approaches pledge to improve aforementioned challenges by transferring knowledge gained from different domains and tasks to a downstream task.

The main goal of this thesis is to provide some approaches to detect hate speech and offensive language in social media using recent advanced NLP methods such as transfer learning and meta learning. To this end, we consider the problem in a monolingual and multilingual setting in which we are dealing with English and low-resource languages respectively, to provide some methods for identification of hate speech and offensive language in social media.

1.2 Objectives and Contributions of the Thesis

In this section, we outline the main objectives of this thesis in which each objective is represented as one contribution. This thesis aims to analyze how Transfer Learning approaches can be leveraged to address the identification of hate speech and offensive language in monolingual and multilingual perspectives. The main objectives to achieve this aim are as follows:

- To analyze user-generated content in social media and its relation with hate speech.
- To propose an automatic model based on Transfer Learning approach for detecting hate speech and tackling the problem of unintended data-driven bias.

- To expand the hate speech and offensive language detection tasks into low-resource and multilingual settings with leveraging Transfer Learning techniques.

Our approach to achieve the above research objectives is organized into three parts as three contributions. We discuss each contribution as follows:

C1: The first contribution is on analyzing user-generated textual content on social media. This contribution proposes an effective framework to measure the similarity between the posts and following comments of a news agency page on Facebook and distinguish the related and unrelated written comments to the actual post in terms of the topics discussed. The proposed framework introduces a novel feature engineering by combining a lexical, topical, and semantical set of features and leveraging word embeddings approach. The dataset used in this contribution is collected from Facebook, and is annotated based on a set of defined rules to determine the most relevant comments to the posts' topics. Then, a supervised machine learning model is trained on a portion of data to predict related and unrelated comments of each post, and the learned classifier is applied on the rest of dataset to investigate user-generated content in a large scale, as a case study. Furthermore, a publicly available tool is used to measure and analysis the toxicity of the comments.

C2: The second contribution is about detecting hate speech in a monolingual setting where only English labeled data is used. This contribution aims to design and develop an automatic hate speech detection model based on Transfer Learning techniques to improve performance results and deal with the lack of enough labeled data in the downstream task. Furthermore, it studies the problem of unintended data-driven bias in our automatic hate speech detection model and introduces an approach to alleviate its effects. The general term bias is used to describe problems related to the collecting or annotating of data that might result in prejudiced decisions on the bases of demographic features such as race, sex, religion, etc. For example, a classifier trained on a dataset with a systematic racial bias tends to predict content written in a specific dialect as hateful or offensive at substantially higher rates. More specifically, this contribution provides two sub-contributions as follows:

C2.1: Firstly, it proposes a novel Transfer Learning approach based on an existing pre-trained language model called BERT (Bidirectional Encoder Representations from Transformers) [6]. More specifically, it investigates the ability of BERT at capturing hateful context within social media content by using new fine-tuning methods. To evaluate our proposed approach, we use two publicly available datasets that have been annotated for racism, sexism, hate, or offensive content

on Twitter. Furthermore, it conducts an experiment to inspect the impact of the proposed Transfer Learning approach in a shortage of labeled data and in capturing syntactical and contextual information of BERT embeddings.

C2.2: Secondly, it introduces a bias alleviation mechanism to mitigate the effect of bias in training set, which may be the result of the collection or annotation process of training data, during the fine-tuning of the pre-trained BERT-based model for hate speech detection. Toward that end, it uses a regularization method to reweight input samples, thereby decreasing the effects of high correlated training set' s n-grams with class labels, and then fine-tunes the pre-trained BERT-based model with the new re-weighted samples. To evaluate the proposed bias alleviation mechanism, it employs a cross-domain approach in which it uses the trained classifiers on the aforementioned datasets to predict the labels of two new datasets from Twitter, AAE-aligned and White-aligned groups, which indicate tweets written in African-American English (AAE) and Standard American English (SAE), respectively. This contribution could institute the first step towards debiasing hate speech and abusive language detection systems.

C3: The third contribution is on tackling the hate speech and offensive language in a multilingual and low-resource setting in which there is no or few available labeled data including hateful or offensive content in a specific language. This contribution aims to investigate the ability of monolingual and multilingual pre-trained language models in Persian offensive language detection task; where it provides the first Persian offensive language dataset to the research community. Furthermore, it introduces a meta learning approach for cross-lingual hate speech detection, where there is only k labeled data available in a target language. More specifically, this contribution provides two sub-contributions as follows:

C3.1: Firstly, it focuses on the problem of offensive language in Persian as a low-resource language, where there is not a labeled data available for hate speech or offensive language. Hence, it presents the first corpus of Persian offensive language consisting of $6k$ out of $520k$ randomly sampled micro-blog posts from Twitter to deal with offensive language detection in Persian as a low-resource language in this area. It introduces a method for creating the corpus and annotating it according to the annotation practices of recent efforts for some benchmark datasets in other languages which results in categorizing offensive language and the target of offense as well. Furthermore, it performs extensive experiments with three classifiers in different levels of annotation (offensive vs non-offensive, targeted vs untargeted, and the target of offense as individual, group, or other)

with a number of classical Machine Learning (ML), Deep learning (DL), and transformer-based neural networks including monolingual and multilingual pre-trained language models. At the end, it proposes an ensemble model integrating the aforementioned models to boost the performance of offensive language detection models.

C3.2: Secondly, it focuses on a cross-lingual few-shot learning problem where there is a sufficient amount of labeled data in source languages and a few labeled data in a target language. It employs a meta learning approach based on Model-Agnostic Meta-Learning (MAML) [7] method to use trained knowledge from different source languages for adapting to the target language with few labeled data. It considers hate speech and offensive language detection as two separated tasks where each task has a specific dataset containing different languages with a similar definition of hateful or offensive content.

1.3 Publications List

Journal Papers

- Marzieh Mozafari, Reza Farahbakhsh, and Noel Crespi, "Hate speech detection and racial bias mitigation in social media based on BERT model", PLoS ONE 15(8): e0237861, 2020.

Conference Papers

- Marzieh Mozafari, Reza Farahbakhsh, and Noel Crespi, "A BERT-Based Transfer Learning Approach for Hate Speech Detection in Online Social Media", Complex Networks 2019: 8th International Conference on Complex Networks and their Applications, Springer International Publishing, Cham, Dec 2019, Lisbon, Portugal, pp. 928-940.

- Marzieh Mozafari, Reza Farahbakhsh, and Noel Crespi, "Content Similarity Analysis of Written Comments under Posts in Social Media", 2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS), Oct 2019, Granada, Spain, pp. 158-165.

Under Review

- Marzieh Mozafari, Reza Farahbakhsh, and Noel Crespi, "Offensive Language Detection in Low Resource Languages: a use case of Persian language", Transactions on Asian and Low-Resource Language Information Processing, 2020.

- Marzieh Mozafari, Reza Farahbakhsh, and Noel Crespi, "Few-Shot Cross-Lingual Hate Speech and Offensive Language Detection using Meta Learning", Information Processing and Management, 2021.

1.4 Relationship of Publications with Contributions

In this section, we provide the relationships of publications with contributions.

- The publication 'Content Similarity Analysis of Written Comments under Posts in Social Media' corresponds to Contribution C1 in Chapter 3.
- The publications 'A BERT-Based Transfer Learning Approach for Hate Speech Detection in Online Social Media' and 'Hate speech detection and racial bias mitigation in social media based on BERT model' correspond to Contributions C2.1 and C2.2 in Chapter 4.
- The submitted papers 'Offensive Language Detection in Low Resource Languages: a use case of Persian language' and 'Few-Shot Cross-Lingual Hate Speech and Offensive Language Detection using Meta Learning' correspond to Contributions C3.1 and C3.2 in Chapter 5.

1.5 Outline of the Thesis

The thesis is structured into six chapters.

- Chapter 1 describes the background of research topics, motivation, contributions of this thesis, summary of each chapter and the outline of the thesis.
- Chapter 2 presents an overview of background information that is relevant in order to understand the contents of this thesis, i.e., hate speech definition and detection, natural language processing techniques, transfer learning, meta learning, and few-shot learning.
- Chapter 3 presents an experiment to analyze user-generated content in social media and give a brief view of its relation with toxicity and hate speech.
- Chapter 4 presents the hate speech detection methodology in a monolingual setting, which is divided into two parts: i) automatic detection of hate speech based on Transfer Learning approach and ii) mitigating unintended data-driven and algorithm-driven bias in the proposed method.

- Chapter 5 presents the hate speech and offensive language detection in a multilingual setting to address the problem of low-resource languages in hate speech detection task. This chapter is divided into two parts: i) offensive language detection in Persian and ii) cross-lingual hate speech and offensive language detection using a meta learning approach.
- Chapter 6 summarizes the thesis and provides an outlook into the future.

1.6 Ethical Considerations

Regarding language concerns, it is important to point out that this thesis uses words or language that is considered profane, vulgar, or offensive by some readers. Owing to the topic studied in this thesis, quoting offensive language is academically justified but neither we nor any of publication venues, in which we published the outputs of this thesis, in any way endorse the use of these words or the content of the quotes. Likewise, the quotes do not represent our opinions, and we condemn online harassment and offensive language.

Regarding General Data Protection Regulation (GDPR) compliance, to respect privacy and ethical aspects of users on social media, we did not collect any sensitive and personal information of users. We only collected publicly available data from Twitter and Facebook and enforced a few steps to protect user privacy by eliminating contact information of users and anonymizing it.

Chapter 2

Background and Related Technologies

Contents

2.1 Overview **28**
2.2 What Is Hate Speech? **28**
2.3 Major Advancements in NLP **30**
2.3.1 Language Modeling 30
2.3.2 Text Representations in NLP 31
2.3.3 Sequence to Sequence Models 32
2.3.4 Attentions 33
2.3.5 Transformers 33
2.4 Transfer Learning in NLP **33**
2.4.1 Generalized Language Models 34
2.4.2 Multilingual Language Models 36
2.4.3 Meta Learning 37
2.4.4 Few-Shot Learning 38
2.5 Automatic Detection of Hate Speech **38**
2.5.1 Machine Learning Approach 39
2.5.2 Deep Learning Approach 40
2.6 Summary and Conclusion **40**

2.1 Overview

The background and related technologies presented in this chapter give a general overview relevant to the main topics of the thesis and set the stage for the subsequent chapters. Later on, a separate and detailed overview of the related work will be discussed for each study in this thesis.

2.2 What Is Hate Speech?

There is enormous variation in the definition of hate speech and a legal definition of hate speech varies from country to country. In the international legislation, any kind of communication in speech, writing, or behavior that advocates incitement to harm, discrimination, hostility, or violence on the base of social or demographic identities of an individual or minority rights is regularized as hate speech [8]. Governments are responsible to restrict hate speech by drafting a law against various types of hate speech especially in online platforms. In general, any type of online content that targets a person or group based on their race, religion, ethnic origin, sexual orientation, disability, or gender is established as online hate speech [9]. Organizations that mediate online communications such as Facebook, Twitter, Google, etc., known as internet intermediaries, have tried to bind users to a set of rules and allow companies to certain forms of expression by developing their own definitions of hate speech. Table 2.1 shows different definitions of hate speech from online platforms, academic community, European Union (EU) institution, and United Nations (UN).

According to Table 2.1, there is a discursive standard criteria between different definitions, however, in practice it is more marginal due to the subjective and implicit nature of this phenomenon in online environments. Indeed, there are different concepts partially overlapped with hate speech such as offensive language [10–12], cyberbullying [13, 14], aggression [14–17], and toxicity [18], that make it difficult to have a distinctive definition for hate speech. Therefore, in the scope of this thesis, we define hate speech as:

"Any textual content in online social media attacking, diminishing, disparaging, or dehumanizing a person or a group based on social identity and characteristics such as gender, sexual orientation, nationality, religion, disability, race, color, or other characteristic. This content may incite violence or hate against the mentioned individuals or groups in real world. Furthermore, it can be in the form of explicit or implicit, where there is unambiguous or ambiguous derogatory or insulting terms implying hatred, respectively."

It should be noted that, in some sections of this thesis, we may consider an alternative concept such as offensive language, instead of hate speech, that we define it as:

"A language to refer to a content comprising any form of non-acceptable language, profane language or swear words, or a targeted offense including insults or threats."

Table 2.1 – Hate speech definition from different sources.

Platforms		Definition
Online platforms	Facebook[1]	Facebook defines hate speech as a direct attack against people on the basis of what called protected characteristics such as race, ethnicity, national origin, disability, religious affiliation, caste, sexual orientation, sex, gender identity and serious disease. We define attacks as violent or dehumanizing speech, harmful stereotypes, statements of inferiority, expressions of contempt, disgust or dismissal, cursing, and calls for exclusion or segregation.
	Twitter[2]	Twitter prohibits the promotion of hateful content against a protected group, individual, or organization based on, but not limited to, Race, Ethnicity, Color, National origin, Sexual orientation, Sex, Gender identity, Religious affiliation, Age, Disability, Medical or genetic condition, Status as a veteran, Status as a refugee, Status as an immigrant. Furthermore, it condemns any kind of degrading, mocking, or harassing references to events or practices that negatively affected a protected group.
	YouTube[3]	YouTube removes content promoting violence or hatred against individuals or groups based on any of the following attributes: age, caste, disability, ethnicity, gender identity, nationality, race, immigration status, religion, sex/gender, sexual orientation, victims of a major violent event and their family members, and veteran status.
Academic Communities	Nobata [19]	They defined hate speech as language that attacks or demeans a group based on race, ethnic origin, religion, disability, gender, age, disability, or sexual orientation/-gender identity.
	Waseem [2]	They defined hate speech as any kind of content that is contained sexist or racial slurs, attacking a minority, seeking to silence a minority, criticizing a minority, defending xenophobia or sexism; by applying a list of criteria based in critical race theory.
Council of the European Union[4]		The EU combats all conduct publicly inciting to violence or hatred directed against a group of persons or a member of such a group defined by reference to race, color, religion, descent or national or ethnic origin.
United Nations [8]		The term hate speech is understood as any kind of communication in speech, writing or behaviour, that attacks or uses pejorative or discriminatory language with reference to a person or a group on the basis of who they are, in other words, based on their religion, ethnicity, nationality, race, colour, descent, gender or other identity factor.

[1] https://www.facebook.com/communitystandards/hate_speech
[2] https://help.twitter.com/en/rules-and-policies/hateful-conduct-policy
[3] https://support.google.com/youtube/answer/2801939?hl=en
[4] https://ec.europa.eu/info/policies/justice-and-fundamental-rights/combatting-discrimination/racism-and-xenophobia/eu-code-conduct-countering-illegal-hate-speech-online_en

Hate speech is not limited to a computer science and engineering point of view, and it is a multidisciplinary field in conjunction with the social sciences, law, and political studies as well. However, in the scope of this thesis, we focus on a computer science perspective. In each section, according to the datasets that we use and their annotation process, we will specify the meaning and the type of hateful and offensive content in social media.

2.3 Major Advancements in NLP

Natural Language Processing (NLP), also known as computational linguistics, is a subfield of Artificial Intelligence (AI) that concerns with the interactions between computers and human language and deals with helping computers to understand, interpret, and manipulate human language [20]. The current advanced neural network-based NLP models is increasingly applied to different tasks, domains, and languages such as speech recognition, natural-language understanding (NLU), natural-language generation (NLG), machine translation (MT), etc.

2.3.1 Language Modeling

From computational linguistic perspective, the first attempt in NLP was made by Turing [21], which outlines concept of Turing test[5], a test for machine intelligence, to answer this question: *"Can machines think?"*. On the other hand, from theories of linguistics perspective, Chomsky [22] introduced a new style of grammar, called Phase-Structure Grammar, in which the structure of a sentence is changed to be understandable by computers. These efforts promoted new researches in NLP domain including symbolic, statistical, and neural NLP approaches. Up to the 1980s, different sets of hand-written rules were used by computers to emulate natural language understanding. With the emergence of machine learning algorithms for language processing in the late 80s, data-driven computations including statistics and probability were used to automatically learn the hand- written rules through the analysis of large corpora. Using statistical methods in NLP resulted a huge development in different NLP tasks especially in language modeling and machine translation. However, the availability of large corpora through internet along with the rapid advances in deep neural network-based models and computational powers caused to a high usage of neural network methods with state-of-the-art results in many natural language tasks [23,24].

One of the important language processing tasks is language modeling that is the task of determining the probability of a given word or sequence of words occurring in a sentence by using various statistical and probabilistic techniques. Bengio et al. [25] proposed the

[5]The Turing test is a criterion of intelligence in computing systems based upon the system's ability to impersonate human communication.

first neural language model that used a feed-forward neural network of three layers to learn the parameters of conditional probability distribution of the next word, given the previous $n-1$ words. Collobert et al. [26] demonstrated the advantage of pre-trained word embeddings for different downstream NLP tasks. To overcome the fixed length context in feed-forward neural network-based language models and model long-term dependencies common in natural language, recurrent neural networks [27] and bi-directional long short-term memory networks [28] were used. Kalchbrenner et al. [29] adapted a convolutional neural network architecture in NLP by proposing a model for the semantic modeling of sentences in which the local context of the text is captured.

Do note that, language modeling is the core of later advances in NLP such as word embeddings, sequence-to-sequence models, and pre-trained language models that we will explain in the following.

2.3.2 Text Representations in NLP

There is a long history behind transforming a text corpus into a numeric representation of words and documents for a machine learning model. Here, we explain the most popular text representations that we have used in this thesis as well.

Traditional Context-Free Representations Bag of Words (BoW) was one of the first attempts to vectorize a text in which each element in the vector corresponds to a unique word or n-gram, contiguous sequence of n words, in the corpus vocabulary and shows the number of occurrences of a specific word or n-gram in the vocabulary in the text. To reflect how important a word is to a document in a collection of documents, called corpus, the frequency of words is weighted by term frequency-inverse document frequency (TF-IDF) measure. Both BoW and TF-IDF were unable to capture the meaning and semantic of the words in the corpus and they suffered from the sparse representations. Therefore, neural word embeddings were proposed to address these limitations.

Word Embeddings A word embedding is a dense vector representation for a word, a real-valued vector, learned by using a shallow neural network trained on large amount of unlabeled data. The more two words are semantically related, the more their vector representations are close. Although word embeddings, was introduced for the first time by Bengio et al. [25], Mikolov et al. proposed a more efficient way to learn word embeddings, called Word2Vec [30]. The Word2Vec has two variations Continuous Bag of Words (CBOW) and Skip-Gram where the word embeddings are learned by predicting the current word based on its context or by predicting the surrounding words (context) given a current word, respectively [31]. In addition, a toolkit[6] was created that can be used to train a model from scratch on a desired task with a large corpus or to leverage the pre-trained embeddings as

[6] `https://code.google.com/archive/p/word2vec/`

initial vector representation of words. Leveraging a very large unlabeled corpus enabled the word2vec model to capture certain linear relations between words investigated deeply in [32,33], and authors in [34] indicated that these learned relations are not without bias. Afterward, some attempts have been done to extend word level embeddings to sentences and documents embeddings [35]. Pennington et al. [36] proposed the GloVe model, a short for Global Vectors, to learn from the global corpus statistics unlike context window-based methods. Word embeddings are not context-specific, which means there is one vector for a word in different contexts with different meanings.

Contextualized Word Embeddings Word embeddings have been widely conducted in different NLP tasks, however, they are context-agnostic and only can be used to initialize the first layer of a neural network and then a model is trained on data of a downstream task. There is a huge trending history behind adding more context to the word vectors using advanced neural language modeling techniques. Language modeling is a self-supervised technique in which the requirement of human annotations is eliminated, and it can learn both word and sentence representations with a verity of objective functions such as autoregressive language modeling, masked language modeling, skip-thought, cross-view training, etc. Language modeling has been using in many currently successful pretraining approaches such ELMO [37], ULMFiT [38], GPT [39], BERT [6], etc. by learning hierarchical representations. The pre-trained language models make able the models, for supervised NLP tasks, to be pre-trained on language modeling and then fine-tuned with labeled data of a downstream task. Therefore, many features of language relevant for downstream tasks are captured by language modeling, such as long-term dependencies, hierarchical relations, sentiment, etc. Since pre-trained language models are very closed concepts to the transfer learning, we will detail these pre-trained language models in Section 2.4.

2.3.3 Sequence to Sequence Models

One of the milestones in NLP was the emergence of sequence to sequence models, abbreviated to Seq2Seq, in 2014 by Google to solve complex NLP tasks like machine translation. Seq2Seq model, proposed by Sutskever et al. [40], is a special class of RNN architectures that map one sequence to another sequence based on an Encoder-Decoder architecture. All the information of input sequence are encapsulated by context vectors in an encoder neural network and then the final hidden states of the encoder, which represents the context of the entire sequence, is used to initialize the decoder neural network for generating the output sequence. A verity of neural network architectures such as LSTM [41], convolutional encoders [42], and attentions [41] have been used in both the encoder and decoder to provide information in several hierarchical layers rather than a multitude of recurrent steps to address the strict sequential processing problem.

2.3.4 Attentions

To alleviate the problem of weakened context in seq2seq models, in which the context becomes weak with longer sentences, attention mechanism is proposed [43]. Attention is a key innovation in neural machine translation models that helps to grasp individual parts of the input sequence which are most important at that particular instance. Unlike seq2seq models, attention uses all encoder hidden states with different weights to provide the attention based context vector to the decoder. Different variation of attention models have been proposed to tackle different NLP tasks at which making decisions based on certain parts of the input [44] or obtaining more contextually sensitive word representations by looking at the surrounding words in a sentence or document, known as self-attention [45] is desired.

2.3.5 Transformers

Transformer, proposed by Vaswani et al. [46], is an architecture for transforming one sequence into another one relying on an encoder and decoder components, which is the same as seq2seq mechanism. However, it does not employ any recurrent or convolutional neural networks directly and it solely utilizes attention mechanisms. The capability of being more parallelizable and efficient by reducing the training time significantly has made Transformer as a widely used architecture in different NLP tasks such as pre-trained language models [6,39] which have been trained with huge general language datasets.

2.4 Transfer Learning in NLP

As explained in previous section, the advancement in deep learning leads the RNN, LSTM, and CNN based neural networks to achieve a reasonable improvement in different NLP tasks [47]. However, deep learning is a training data intensive approach, which means that the requirement of a large amount of training data is vital for achieving such promising results in different NLP tasks such as text classification, question answering, name entity recognition, language modeling, etc. Furthermore, some deep neural networks suffer from a huge computation time or overfitting due to the lack of enough labeled data along with their huge number of parameters during the training phase. Hence, transfer learning emerges as an adaptation for these limitations.

Transfer learning is an approach that uses a knowledge gained from one task (source task) to bootstrap a different but similar task (target task), in which need for significant additional training data and computational time and cost are reduced [48]. Such a deep learning model, called pre-trained model, prevents to build a model from scratch in different applications. Figure 2.1 depicts the learning process in traditional machine learning and

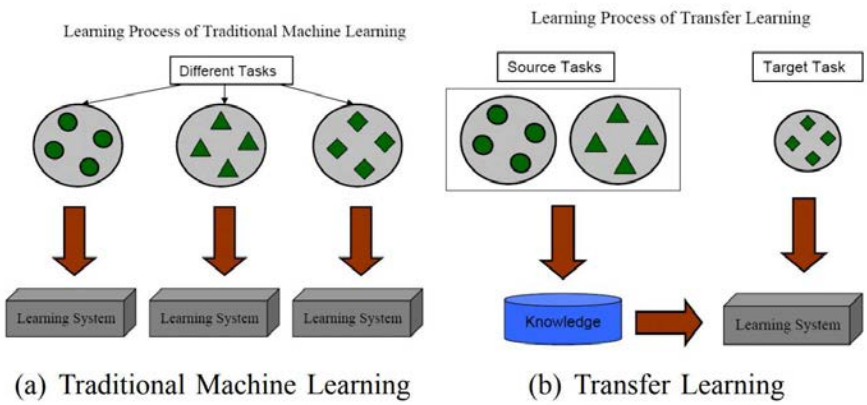


Figure 2.1 – Different learning processes between traditional machine learning and transfer learning [48].

transfer learning approaches. The traditional way used to train a machine learning model is depicted in Figure 2.1(a). Having a labeled data on a specific task, we initialize a model from scratch with random weights and train the model on that task. Therefore, when we tackle another task, we have to train another model from scratch again containing gathering labeled data, initializing random weights, and training the model. However in transfer learning, as shown in Figure 2.1(b) , we have some source tasks that we have learned a model on that, and then we use this knowledge for tackling a new target task.

Although transfer learning has been comprehensively studied in computer vision because of the availability of vast amounts of training data [49], it has raised major attentions in NLP, recently, and resulted in state of the art for many supervised NLP tasks. The main reasons behind a surge in interest in using TL in NLP tasks are: i) many NLP tasks share common knowledge about languages such as linguistic representations, syntax, structural similarities, etc.; ii) some tasks have common semantics that can be capitalized by transfer learning, for example, a question-answering model trained in English can be used in the same task from another language like German; and iii) collecting and annotating data for different NLP tasks are labor and time consuming. On the other hand, there is a huge corpus of unlabeled textual data that can be used for training pre-trained models using TL.

2.4.1 Generalized Language Models

McCann et al. [50] proposed the first contextualized word embeddings, known as Contextual Word Vectors (CoVe), that leverages both encoder-decoder architecture and attention mechanism. The encoder is based on a two-layer bidirectional LSTM and the attentional decoder is based on a two-layer unidirectional LSTM. Unlike traditional word embeddings , word representations from this model are functions of the entire input sentence, however,

their contributions to the final performance was limited by the model architecture of the target task. Since CoVe model was a supervised learning model bounded by available datasets on the translation task, Peters et al. [37] proposed an unsupervised two-layer bidirectional language model named ELMO, short for Embeddings from Language Model. Different layers of ELMO represent syntactic and semantic information that can be capitalized by task-specific models. Since language model embeddings from ELMO could be used only as features in a target model, for the first time, Howard et al. [51] brought the idea of using generative pre-trained language modeling and task-specific fine-tuning in different NLP tasks by concatenating two independently trained left-to-right and right-to-left multi-layer LSTMs on the WikiText-103 corpus.

One of the firmly established state-of-the-art neural network models is Transformer, explained in the previous section. Radford et al. [39] proposed a variant of the Transformer, named Generative Pre-training Transformer (GPT), to train an unsupervised language model on a diverse corpus of unlabeled text using a multi-layer transformer decoder. GPT uses a 12-layer decoder-only stack containing a multi-headed self-attention layer operating on the embeddings of input sequences and a point-wise feed-forward layer for producing an output distribution over target tokens. The main advance of GPT model was that it could be fine-tuned for all downstream tasks without a customized task-specific model, however, it considered only left-to-right context in the input sequences. To overcome the unidirectional nature of GPT, Devlin et al. [6] proposed Bidirectional Encoder Representations from Transformers (BERT) model to predict based on both left-to-right and right-to-left context. BERT adapted the Transformer architecture with a multi-layer bidirectional Transformer encoder to generate a language representation model. Then, this pre-trained model can be fine-tuned with one additional output layer on downstream tasks. BERT performs two specific training strategies Mask Language Model (MLM) and Next sentence prediction. During pre-training in the MLM task, some tokens from the input are masked randomly and the model tries to predict the masked word based on its context. During pre-training in the next sentence prediction task, model tries to jointly learn text-pair representations. During the fine-tuning, the BERT model uses its pre-trained parameters for the first initialization and then fine-tunes all parameters using labeled data from the downstream tasks. BERT outperforms many task-specific architectures and achieves state-of-the-art results in a wide range of sentence-level and token-level tasks.

By inspiring the BERT model, a variation of language representation models have been developed. Lan et al. [52] proposed a light-weighted version of BERT model, named ALBERT. Using the same architecture of BERT model, ALBERT performs 1.7x faster with 18x fewer parameters compared to the BERT model. Liu et al. [53] presented a modification of the BERT model named RoBERTa, stands for Robustly optimized BERT approach, in

which a variation of modifications were applied in training step of the BERT model; including removing the next sentence prediction task, making the masking pattern dynamically, using longer sequences patterns of input data in training, etc. Raffel et al. [54] built a unified framework based on encoder-decoder architecture that converts all text-based language problems into a text-to-text format, named Text-to-Text Transfer Transformer (T5).

Following the GPT model, Radford et al. [55] proposed OpenAI GPT-2 language model pre-trained on a large corpus containing 8 million Web pages, with 1.5 billion parameters, that achieved promising results in a zero-shot transfer setting without any task-specific fine-tuning. Very recently, in 2020, Brown et al. [56] introduced GPT-3 that mainly focuses on a few-shot setting in which the model is given a few demonstrations of the task at inference time without any weight updating. In comparison with GPT-2, GPT-3 has 175B parameters and is 10x larger than GPT-2 and it requires no or very little fine-tuning after pre-training. Due to the concerns about malicious applications of the technology, providers of GPT-2 and GPT-3 have not made their source code public.

2.4.2 Multilingual Language Models

Apart from monolingual word embeddings and generalized language models, multilingual word embeddings and multilingual pre-trained language models have raised a lot of attention recently. Multilingual word embeddings, also called cross-lingual word embeddings, have a shared embedding space between two languages in which semantically similar words in two languages are close together. Having such a vector space makes machine learning algorithms able to train on data in any languages.

A variety of models has been proposed to train cross-lingual word embeddings that we refer to a comprehensive survey done by Ruder et al. [57] for more details. A common strategy between most of the models is that they rely on additional cross-lingual resources such as bi-lingual dictionaries or parallel corpora to train cross-lingual word representations. However, very recently, some attempts have been dedicated to creating unsupervised cross-lingual word embeddings in which models only rely on monolingual text corpora [58–60]. Apart from cross-lingual word embeddings, Artetxe et al. [61] presented universal language agnostic sentence embeddings model LASER, short for Language-Agnostic SEntence Representations, by pre-training a single sentence encoder on a comprehensive dataset consisting of sentences and their translation in 93 languages.

Most of the research efforts have focused on developing pre-trained language models on English data, however, transformer architectures are adapting to the multilingual language models in which a single language model is trained with data from multiple languages jointly. After proposing BERT by Google, Devlin et al. [6] proposed multilingual BERT model, called mBERT, to feed the BERT model with texts from multiple languages without

any cross-lingual supervision. The representations of mBERT are partially language independent and it is able to work with 104 languages. However, Lample et al. [62] proposed a cross-lingual language model, named XLM, to establish a connection between different languages by using a cross-lingual supervision. The XLM model used a translation language modeling (TLM) objective to leverage parallel data when it is available. Conneau et al. [63] proposed an unsupervised cross-lingual language model, XLM-RoBERTa, by leveraging a significantly large amount of training data in 100 languages. The proposed model employed a transformer-based multilingual masked language model objective to achieve state-of-the-art results on a wide range of cross-lingual transfer tasks including sequence labeling and question answering. These multilingual language models can be used in either extracting multilingual hidden representations or fine-tuning the pre-trained language model on the training data of a downstream task .

2.4.3 Meta Learning

Meta Learning, known as learning to learn, is conceptually related to transfer learning because both of them deal with incorporating additional knowledge from different source of data that is not quite from a target task but will help the target task to be solved more efficiently. The main difference between transfer learning and meta learning is that the former transfers the useful prior knowledge from a source task to a new downstream task while the latter improves the learning ability of a model by leveraging the different but related training data from a distribution of tasks. Meta learning can be used in situations in which we do not have a large dataset for every possible task and want to quickly adapt or generalize to new tasks based on our previous experiences.

Meta Learning Formulation: Conventionally, in supervised machine learning a task-specific model is trained on a task-specific labeled data. Given a classification task T with training data $D = (x_i, y_i)$, such that x_i is the feature vector of the ith sample and y_i is its label from a label space $\mathcal{Y}$, a classifier f parameterized by θ outputs the probability of a sample belonging to the class y_i as learning following:

$$P_\theta\left(y \mid x\right) \tag{2.1}$$

where our objective is to find:

$$\theta^* = arg\,\underset{\theta}{max} \underset{(x,y)\in D}{\mathbb{E}} \left[P_\theta\left(y|x\right)\right] \tag{2.2}$$

However in meta learning, instead of having a single task-specific data, we have a distribution over tasks $p\left(\mathcal{T}\right)$ where training tasks and test tasks are exclusive. Considering M training and N test tasks, we have a meta-training set $\mathcal{D}_{meta-train} = \left\{\left(D^i_{train}, D^i_{test}\right)_{i=1}^{M}\right\}$

and a meta-test set $\mathcal{D}_{meta-test} = \left\{ \left(D^j_{train}, D^j_{test} \right)_{j=1}^{N} \right\}$ in which each entry is a bunch of training samples and test samples of a task T_i from a distribution over tasks $p(\mathcal{T})$, denoted as D^i or D^j. We sample a subset of the $Y \in \mathcal{Y}$ labels, such that $\mathcal{D}^Y_{meta-train}, \mathcal{D}^Y_{meta-test} \in \mathcal{D}$, and train the model on the meta-train set and test it on the meta-test set in an episodic fashion. Therefore, the model learns to learn from smaller datasets gradually by optimizing the loss and updating the parameters through backpropagation; where our objective is:

$$\theta = \underset{\theta}{arg\,max} \mathbb{E}_{Y \in \mathcal{Y}} \left[\mathbb{E}_{D^Y_{meta-train} \subset \mathcal{D}, D^Y_{meta-test} \subset \mathcal{D}} \left[\sum_{(x,y) \in \mathcal{D}^Y_{meta-test}} P_\theta \left(x, y, \mathcal{D}^Y_{meta-train} \right) \right] \right] \tag{2.3}$$

Three types of approaches are proposed to meta learning: i) metric-based; ii) model-based; and iii) optimization-based. In this thesis, we use an optimization-based meta learning approach for solving the problem of hate speech and offensive language detection in a cross-lingual few-shot setting; which is detailed in Chapter 5.

2.4.4 Few-Shot Learning

One of successful applications of meta learning is few-shot learning. Few-shot learning, also known as k-shot learning, is the problem of learning using a few number of labeled data (k number of samples per class) in a downstream task. Suppose we have a hate speech classification problem with 3 classes as "Racism", "Sexism", "Neither". If each class has 15 samples, then it is defined as 15-shot learning. If the number of samples per class is 1 then it is defined as one-shot learning while if there is not any sample per class it is referred to as zero-shot learning.

2.5 Automatic Detection of Hate Speech

A great deal of research has been conducted to demonstrate different aspects of hate speech, including, but not limited to: (1) its definition [2,3] and typology [64]; (2) the data collection and annotation process [65–67]; (3) investigation of automatic machine learning and deep learning classification models [4, 68–70] and their generalizations [71]; (4) investigation of the most effective features of hate speech classification [2, 19]; (5) unintended bias(es) in datasets or classification models [71–75]; and (6) some of the relevant ethical principles [76].

Abusive language is an unwelcome online conduct that is based on using different remarks intended to be demeaning, humiliating, intimidation, mocking, ridicule, insulting, or belittling. These remarks may or may not be based on an individual's protected status or protected characteristics such as race, color, religion, sex, national origin, sexual

orientation, or gender identity of an individual [2]. By considering abusive language as an umbrella term, that covers different types of online abuse, extensive studies have been done to address hate speech [2, 3, 77, 78], offensive language [10–12], cyberbullying [13, 14], aggression [14–17], and toxicity [18] detection in social media. A wide range of studies has therefore been dedicated to developing automatic methods to detect these types of content in social media by proposing different models based on machine learning and deep learning approaches.

2.5.1 Machine Learning Approach

To detect hateful and abusive contents automatically, different machine learning approaches utilizing distinguishable feature engineering techniques have been employed in the literature [2, 19, 79] and it is asserted that surface-level features such as bag of words, word-level and character-level n-grams, etc. are the most predictive features in this task. Regarding classification perspective, different algorithms such as Naïve Bayes [80], Logistic Regression [2, 3], Support Vector Machines [68], multi-view tacked Support Vector Machine (mSVM) [69], etc. have been used to train a classifier for predicting the hateful contents.

As a baseline, Waseem et al. [2] provided a test with a list of criteria based on a work in gender studies and critical race theory to annotate a corpus of more than $16k$ tweets as racism, sexism, or neither. To classify tweets, they used a logistic regression model with different sets of features, such as word and character n-grams up to 4, gender, length, and location. They found that their best model produces character n-gram as the most indicative features, and using location or length is detrimental. Davidson et al. [3] collected a $24k$ corpus of tweets containing hate speech keywords and labeled the corpus as hate speech, offensive language, or neither by using crowd-sourcing and extracted different features such as n-grams, some tweet-level metadata such as the number of hashtags, users' mentions, retweets, and URLs, Part Of Speech (POS) tagging, etc. Their experiments on different multi-class classifiers showed that the Logistic Regression with L2 regularization performs the best at this task. Malmasi et al. [68] proposed an ensemble-based system that used some linear SVM classifiers in parallel to distinguish hate speech from general profanity in social media. Recently, MacAvaney et al. [69] discussed different aspects of an automatic hate speech system. They mainly addressed challenges pertaining to the definition of hate speech, dataset collecting and annotation process and its availability, and the characteristics of existing approaches. Furthermore, they proposed a multi-view tacked Support Vector Machine (mSVM) based approach that achieved near state-of-the-art performance; using word and character n-grams up to 5 as feature vectors.

2.5.2 Deep Learning Approach

Due to the advances in deep neural network models and the volume of available labeled data in this domain, mainly for English, different neural networks based approaches such as Recurrent Neural Network (RNN) [81], Long Short-Term Memory (LSTM) [70], Convolutional Neural Network (CNN) [82], bidirectional LSTMs [83], Gated Recurrent Units (GRUs) [4] have been employed in identification of hate speech content, which outperformed traditional machine learning models.

With regard to the word representation as a dense vector pre-trained on a large amount of data, some basic deep learning approaches proposed to tackle the problem of hate speech [4, 84]. The most frequently used word embeddings approaches are Word2Vec [31], Glove [36], and FastText [85]. As one of the first attempts in neural network models, Djuric et al. [86] proposed a two-step method including a continuous bag of words model to extract paragraph2vec embeddings and a binary classifier trained along with the embeddings to distinguish between hate speech and clean content. Badjatiya et al. [70] investigated three deep learning architectures, FastText, CNN, and LSTM, in which they initialized the word embeddings with either random or GloVe embeddings. Gambäck et al. [84] proposed a hate speech classifier based on CNN model trained on different feature embeddings such as word embeddings and character n-grams. Zhang et al. [4] used a CNN+GRU neural network model initialized with pre-trained word2vec embeddings to capture both word/character combinations (e. g., n-grams, phrases) and word/character dependencies (order information). Founta et al. [81] built a unified classification model that can efficiently handle different types of abusive language such as cyberbullying, hate, sarcasm, etc. using raw text and a set of metadata from Twitter including tweet-based, user-based, and network-based features.

Furthermore, researchers have recently focused on the bias derived from the hate speech training datasets [64, 87, 88]. Davidson et al. [87] showed that there were systematic and substantial racial biases in five benchmark Twitter datasets annotated for offensive language detection. Wiegand et al. [88] also found that classifiers trained on datasets containing more implicit abuse (tweets with some abusive words) are more affected by biases rather than once trained on datasets with a high proportion of explicit abuse samples (tweets containing sarcasm, jokes, etc.).

2.6 Summary and Conclusion

This chapter presented a general overview of the major topics relevant to this thesis. To summarize, it covered three major areas in different parts. In the first part, it discussed hate speech in social media and provided its definition and the major ideas behind the solutions

proposed, so far, to tackle this problem. In the second part, it discussed the major advances in NLP and showed our intuition behind using different NLP techniques in this thesis. In the third part, it discussed Transfer Learning and its progress in NLP. In addition to this chapter, for each study in this thesis, a separate related work will be discussed focusing on the main relevant works to the specific study.

Based on our literature review, different machine learning and deep learning models have been proposed by the research community, however, there exist some challenges in the automatic detection of hate speech, which have made this problem far from being solved at scale. The first one is the multilinguality of social media where these platforms foster their users to interact in their primary languages. Hence, it is essential to have automatic detection tools to protect users with different languages, other than English, against hateful and abusive content. However, the majority of proposed ML and DL models are reliant on large volumes of labeled data and their performance relies heavily on the size of training data available. To address this challenge, we will investigate the problem of the limited availability of labeled training datasets for hate speech detection by utilizing transfer learning approach; which has not yet been thoroughly explored in this domain. We will detail some pre-trained language models used in this thesis such as BERT, mBERT, ALBERT, and XLM-RoBERTa and explain the way we fine-tune them on hate speech detection task in Chapters 4 and 5.

The second challenge is the lack of sufficient annotated data containing hatred, offense, and abuse for low-resource languages, because collecting and labeling data is a labor- and time-consuming work. In addition, the complex, subjective, and implicit nature of hate speech makes the annotation process more difficult. To address this challenge, we will introduce a labeled dataset for Persian language as a use case for low-resource language, and will investigate the potential of applying meta learning approach to address the problem of few-shot learning in cross-lingual hate speech and offensive language detection tasks; which has not yet been thoroughly explored in this domain.

Chapter 3

Social Media Content Analysis

Contents

3.1 Introduction **44**
3.2 Related Work **46**
3.3 Methodology and Framework **47**
3.3.1 Features Description 48
3.4 Experiments and Results **52**
3.4.1 Dataset Description 53
3.4.2 Gold Standard Annotation 53
3.4.3 Results 55
3.4.4 Case Study 58
3.5 Conclusion **65**

3.1 Introduction

User-generated contents in online social media including text, images, videos, etc. are one of the significant sources of knowledge in a variety of topics. However, they may broaden the potential for harm as well. For example, comments written by users in Facebook can be informative, truthful, and related to a post's content, or they can be completely or partially unrelated and contain hateful or toxic messages. Therefore, analyzing use-generated content not only reveals the way users are communicating, but also brings more insight into what are hateful, offensive, or toxic content and how they are generating and propagating in online platforms. In this chapter, we analyze the textual content generated by users in Facebook to filter related and unrelated written comments to an actual post. Furthermore, we look into the problem of hatred and toxicity in the comments using a publicly available toxicity detection tool.

Written comments to the posts on social media are an important metric to measure the followers' feedback to the content of the posts. However, the huge presence of unrelated comments following each post can affect many parts of people engagement as well as the visibility of the actual post. Related comments to a post's topic usually provide readers more insight into the post content and can attract their attention. On the other hand, unrelated and toxic comments distract them from the original topic of the post or disturb them by worthless content and can mislead their opinion or even caused them to leave a conversation.

News agencies are disseminating the news through social media such as Facebook to a large community of people; meanwhile, people are more interested in following the breaking news and stories from this platform rather than the main news agencies' websites[1]. Comments generated by users are one of the significant sources of information following the posts of news agencies' pages in Facebook which can be truthful and related to a post's content, or they can be completely or partially untrue and unrelated. Some popular news agencies' pages in Facebook, such as the BBC News, have millions of readers per day and so generating the unrelated comments by users can have a negative effect on their visiting traffic and reader's satisfaction [89]. Since readers consider comments as a valid source of supplemental information, they prefer to see comments that are more meaningful and discuss a post's topics rather than unrelated concepts such as personal opinions, advertisements, bot-generated content, etc. Therefore, identifying such unrelated comments following a post is a big challenge in social media content analysis [90–92].

A growing body of research has focused on analyzing social media content generated

[1]News use across social media platforms (2016) by Gottfried: `https://www.journalism.org/2016/05/26/news-use-across-social-media-platforms-2016`

by users [93–96]. Many approaches have been suggested, including lexical or syntactic matching, semantic knowledge, latent topic models such as Latent Semantic Analysis (LSA) and Latent Dirichlet allocation (LDA) [97], and word embeddings [30], to identify similarities between short texts. These efforts have used external corpora such as Wikipedia or webpages related to the post content to enrich their corpus. Other studies have tried to identify unrelated comments that are generated to distribute spontaneous spam, influence public opinions, advertise products and events, etc. by leveraging text contents or temporal and spatial user behavior in social media [90, 92, 98]. In some content analysis applications, where we are dealing with posts and following comments as short texts in social media, we may not have access to a post's complete story or to some external corpora such as Wikipedia or Google web pages related to the post content to enhance existing short texts. On the other hand, in real-time content analysis, using these sorts of external corpora can be time-consuming and thus may have a negative effect on the efficiency of a real-time application.

To address these issues, we propose a combination of lexical, topical, and semantical features by taking advantage of word embeddings approach to identify related and unrelated comments following the posts of a news agency page in Facebook without referring to a post's entire article. By applying word embeddings technique and extracting abstract semantic concepts in numerical form, vector form from both pre-trained word embeddings models and our existing dataset, we can improve topical and semantical features to identify related and unrelated contents.

The primary contributions of this study are:

- Proposing an effective framework to extract three categories of features: lexical, topical, and semantic from the posts and following comments of a news agency page on Facebook. These features then are used to identify related and unrelated contents.

- Using word embeddings approach within both topical and semantical features to enhance similarity detection without having access to the entire story of a post or external corpora related to each post content.

- Analyzing toxicity in the comments associated with the posts to get an insight into the types of hateful content.

- The experiment results show that by using a combination of lexical, topical, and semantic features along with word embeddings, our model can outperform approaches that just use topical modeling methods to identify related/unrelated contents in terms of accuracy, precision, recall, and F1-measure.

3.2 Related Work

A major group of studies has focused on user-generated content (e.g., posts, comments, and reviews) analysis in social media by considering textual contents or temporal and spatial user behaviors [99–101]. Spam content is a specific concept throughout the emails, webpage, blog posts, and comments. Short text type spam such as spam comments following posts in blogs and social networks has attracted further attention [102, 103]. Mishne et al. [104] followed a language-based model to create a statistical model for text generation to identify spam comments in blogs. Bhattarai et al. [105] investigated the characteristics of spam comments in the blogosphere based on their content, with an effort to extract the features of the blog spam comments and classify them by applying a semi-supervised and supervised learning method. Wang et al. [102] aimed to identify diversionary comments as comments designed to deliberately divert readers' attention to another topic on political blog posts. They applied a combination of co-reference resolution and Wikipedia embedding to replace pronouns with corresponding nouns and used the topic modeling method LDA to group related terms in the same topics. A context-aware approach to detect irrelevant comments following posts was proposed by Xie et al. in [90]. Their approach assumed that the context-aware semantics of a comment are determined by the semantic environment where the comment is located. They also focused on facilitating the early detection of irrelevant comments by constructing a corpus of the most similar previous comments to the current posts in the same topic.

As a common approach for topical similarity of texts, topic modeling is used to find hidden topical patterns of words in similar texts [106]. Latent Semantic Analysis (LSA) [107] is the foundational model for the development of a topic model. Since it is not a probabilistic model and thus cannot handle polysemy, other topic models such as probabilistic Latent Semantic Analysis (pLSA) and LDA have been proposed based on LSA [97]. In a corpus, LDA tries to discover a topic distribution over each document and a word distribution over each topic. Both pLSA and LDA need the number of topics and they do not capture the relationship among topics. While topic models can discover latent topics in a large corpus, Dat et al. [108] proposed a new approach to make a combination between Dirichlet multinomial topic models such as LDA and latent feature (LF) vectors of words called word embeddings to improve word-topic mapping learned on a smaller corpus. They showed that in the case of datasets with few or short texts, the LF-LDA model outperforms LDA, significantly improving topic coherence and document clustering tasks. Here for the first time, we use LF-LDA as a feature to determine topical similarity in related/unrelated short text identification task. We describe this model in detail in Section 3.3.1.

Regarding short text mining, a number of recent efforts focus on using topic modeling

methods such as LSA, and LDA [97] to find similarities between short texts in social media. For the first time, Hieu et al. [109] used LDA to enhance the bag-of-word approach and thereby deal with short and sparse texts by finding most of the hidden topics similar to them from large scale data collections. Xie et al. [90] proposed a framework to identify relevant and irrelevant texts by capturing the semantic of short texts in a context-aware approach. Their work considered topic similarity in short texts to capture their relevancy to each other.

Considering all the previous mentioned studies in identifying related/unrelated comments following a post, we believe that it is the first attempt in using a combination of lexical, topic, and semantic-based features to find similarity between short texts. Our model does not rely on the entire story of a post or external webpages content related to the post in comparison with previous studies [90, 102], and we leverage word embeddings approach to enrich the short text corpus. Therefore, it can be applied in different social media applications in which we are just dealing with short texts to categorize them as related/unrelated contents.

3.3 Methodology and Framework

Figure 3.1 depicts our proposed framework. To categorize comments as related/unrelated to a post, the framework takes each post P_i and all comments C_{ij}, where ($j = 1, 2, \ldots, numberofcomments$), following it as input, and returns the predicted label as related/unrelated for each comment as output. As a classification problem, the framework has two main parts: Training and Prediction. The Training part has three main components: Pre-processing, Similarity-based feature extraction, and Supervised algorithm. Pre-processing is where we clean the input data by applying some pre-processing methods such as stop word and punctuation removal, tokenization, and lemmatization. The Similarity-based feature extraction component is the most important part of our framework, as it is where features are defined to capture the degree of similarity between post and comments more effectively. It contains three different feature categories: lexical, topical, and semantical. We try to capture not only the lexical and topical features of texts but also the context of a word, its relation with other words, the context-dependent semantic similarity, etc. by applying the word embeddings approach in topical and semantical categories. To the best of our knowledge, this is the first time that this combination of lexical and topical features is being linked with a word embeddings approach to solve the problem of related/unrelated comments to news agencies' posts in social media. We use both pre-trained Word2Vec models on Google News corpus [30] and Wikipedia [36] and word embeddings learned from our collected corpus. After extracting the features, we apply Support Vector Machines (SVM)

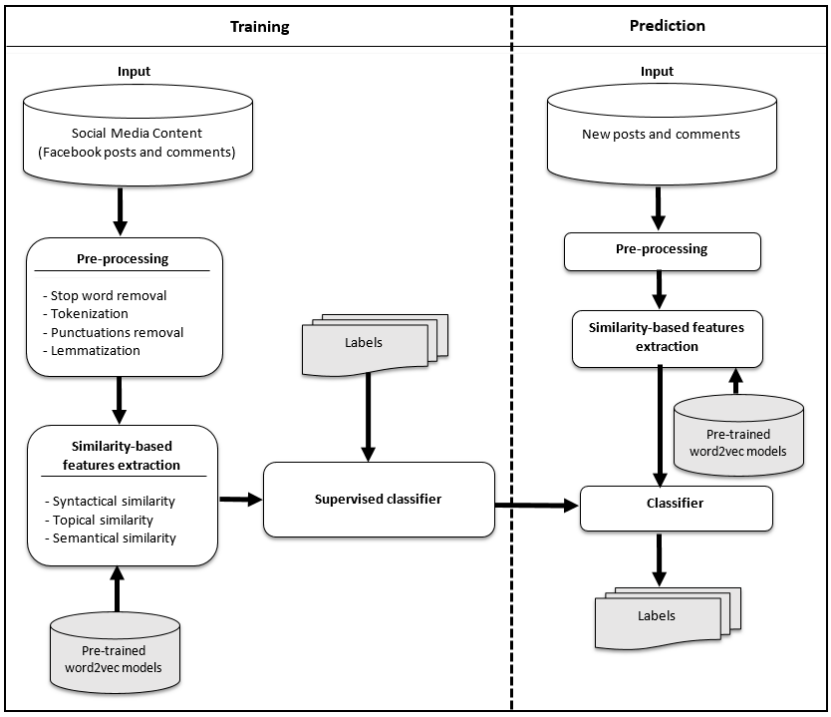


Figure 3.1 – Schema of the proposed framework.

to evaluate the performance of the model in identifying related/unrelated comments. After training our model, our framework will be able to predict the label of each new comment through this classification process.

3.3.1 Features Description

The three categories of features used in this study are shown in Table 3.1. To the best of our knowledge, this is the first time that we are using the combination of both bag of words-based and word embedding-based similarity measures to estimate similarity between post and comments as short texts without including the entire story of the post or external corpus related to the post itself. Among all features shown in Table 3.1, three word embedding-based features: Word2Vec, GloVe, and Native context-word2vec are proposed as new features based on post and comments corpus. We examine these features to determine the similarity between a post and comments following it. Here, we use the native context of a post [90] as a set of one post and all comments following the post, and try to consider not only the pair of post and comment but also to pair a comment and all comments following a post, since these comments are more likely to be similar to each other in terms of language and topics. We also consider the native context of all posts as a corpus and employ some

models like the LF-LDA and word embeddings to capture the context-dependent semantics from short comments. According to Table 3.1, these different similarity measures are described next.

Table 3.1 – Feature sets of the proposed framework

Lexical	**Topical**	**Semantical**	
Cosine	Latent Feature-Latent Dirichlet Allocation (LF-LDA)	**String-based**	**Word Embeddings**
Native context		WordNet	Word2Vec GloVe Native context-word2vec

Lexical similarity: The lexical similarity is a measure of the degree to which the word sets of two given sentences are similar. As posts and comments in social media are generally short, considering the lexical similarity among them can be a way to identify them as related or unrelated. Commentators discuss a post in the comment section, and their comments can be lexically similar to the post or similar to other comments following the post. To capture these kinds of similarities we use Cosine and Native context similarities as follows:

- *Cosine similarity*: by considering each pair of a post and following comment as P_i and C_{ij}, Cosine similarity calculates the similarity between P_i and C_{ij} by measuring the cosine of angle between the term frequency-inverse document frequency (tf-idf) vectors of P_i and C_{ij} determined according to the bag of words approach.

- *Native context*: by defining all comments following a post and the post itself as NC_i (native context), the similarity between each comment C_{ij} and post P_i or other comments following the post is formulated as:

$$similarity(C_{ij}) = cos(m(NC_i), C_{ij}) = \frac{m(NC_i) \cdot C_{ij}}{\|m(NC_i)\| \, \|C_{ij}\|} \tag{3.1}$$

According to Equation 3.1, a tf-idf matrix of the post and all following comments is created. Then for each comment, the cosine similarity between its vector and the mean of other native context vectors is calculated to capture the comments similar to the native context.

If each of the above lexical similarity functions is applied to two semantically related sentences with different lexical terms, the similarity score will be zero because they cannot capture the semantics in the sentences. Therefore, we consider topical and semantical approaches based on word embeddings to include semantic in our model.

Topical similarity: Comments can be related to posts in terms of different topics, which are common between the posts and following comments and what commentators discuss. One of the most frequently used methods to investigate how short texts are similar

in terms of topics is LDA. The LDA models each document as a probability distribution over topics, and each topic as a probability distribution over words based on the co-occurrence of words within documents via tf-idf matrix. Thus, for short documents in a small corpus, LDA results might be based on little evidence and so external corpora such as Webpages or Wikipedia content must be used to improve the topic representations [102, 110]. To deal with this challenge in our study, we use LF-LDA [108] to make topical similarity detection more efficient by leveraging both a latent feature trained on a large corpus and the topic modeling method. In the following, we describe both the LDA and LF-LDA models and explain how we adapt them to identifying related/unrelated content.

- *Latent Dirichlet Allocation (LDA)*: for each post P_i, we apply the topic model LDA to learn the topics from all the comments in native context C_i. LDA assumes that each document has a probabilistic multinomial distribution θ over latent topics, where each topic is characterized by a probabilistic multinomial distribution φ over the words. Both the topic distribution in all documents and the word distribution of topics share a common Dirichlet prior [97, 111]. By assuming α as the parameter of the Dirichlet prior on the per-document topic distribution (θ) and β as the parameter of the Dirichlet prior on the per-topic word distribution (φ), two distributions θ and φ can be given by:

$$\theta = \frac{C_{dj}^{DT} + \alpha}{\sum_{k=1}^{T} C_{dk}^{DT} + T\alpha} \tag{3.2}$$

Where D and T stand for documents and the number of topics, respectively. C_{dj}^{DT} is the number of occurrences of terms in document d that have been assigned to topic j, and:

$$\varphi = \frac{C_{ij}^{WT} + \beta}{\sum_{k=1}^{W} C_{kj}^{WT} + W\beta} \tag{3.3}$$

Where W and T stand for the number of terms and topics, respectively. C_{ij}^{WT} is the number of times that term i has been assigned to topic j. To estimate LDA parameters more accurately, we use the Gibbs sampling approximation method proposed by Heinrich [112].

Here we consider each post and all the following comments as document D and find k topics that are discussed in the comments. After training the LDA on native context C_i, we estimate the topical similarity between each post P_i and its following comment C_{ij} by applying the Jensen-Shannon divergence metric based on the Kullback-Leibler (KL) divergence. Since Jensen-Shannon is a measure of the distance between two probability distributions, we consider the topic distribution over each post and comment as P and Q and calculate their similarity using the following function:

$$JSD\left(P, Q\right) = \frac{1}{2}\left(D_{kl}\left(P, M\right) + D_{kl}\left(Q, M\right)\right) \tag{3.4}$$

$$D_{kl}\left(P,M\right)=\sum P\left(i\right)log\frac{P\left(i\right)}{M\left(i\right)} \tag{3.5}$$

Where $M = 1/2(P + Q)$.

- *Latent Feature-Latent Dirichlet Allocation (LF-LDA)*: is a probabilistic topic model that combines a latent feature model with an LDA model. The LDA extracts topics by relying on the co-occurrence counts from the bag of words approach, where relation between topics cannot be captured [113]. Recently, neural network methods have been used to learn and represent words as vectors in real numbers, known as word embeddings. These vectors have latent features that capture the context of a word in a document, its semantics, and relation with other words [113]. The Word2Vec model is one of the most famous word embeddings models [30]. Based on this vector representation, Dat et al. [108] proposed the LF-LDA model to go beyond LDA for topic modeling. In LF-LDA, the Dirichlet multinomial distribution for topic-to-words has two components: a topic-to-word Dirichlet multinomial component and a latent feature component. This model can perform well on corpora with few or short documents compared to the LDA's requirements. Here we use a pre-trained word embeddings model named Word2Vec, which is trained on a 100 billion word subset of the Google News corpus [30]. For each post P_i, we apply the topic model LF-LDA to learn topics from the post and all its comments. We eliminate each word from the post and comments that is not in the pre-trained Word2Vec models. To estimate the topical similarity between each post and comment through the LF-LDA learned topic-to-word distributions, we use Jensen-Shannon divergence defined in equation 3.4.

Semantical similarity:

- *WordNet-based*: WordNet is one of the earliest methods for extracting semantic similarity or relatedness between a pair of concepts or word senses [114]. It is a large lexical database of English words including nouns, verbs, adjectives, adverbs, etc., and their sets of cognitive synonyms. Since WordNet contains information on nouns, verbs, adjectives, and adverbs, we use Part Of Speech (POS) tagging on each post and comment pair and then find semantic similarity between them by WordNet using NLTK package in Python.

- *Word Embeddings-based*: We use three word embeddings methods to capture the semantic similarity between a post and its comments. We use a combination of pre-trained models Google Word2Vec [30] and Stanford GloVe [36] and we also train a model based on all comments in our dataset. A brief comparison between the effect of these vector-based word representation methods will be presented in Section 3.4.3.

1. *Word2Vec*: For each word in a post, its vector representation with 300 dimensions is extracted from the Google News corpus pre-trained model [30]. The average value is then calculated among all vectors as a 1*300 dimension vector. This process is

repeated for each comment. Finally, the cosine similarity between the post and comment vectors is calculated as a word embedding similarity measure between them. For two documents d_1 and d_2 as post P_i and comment C_{ij}, word embeddings similarity (WESim) between post and comment is defined as follows:

$$WESim\,(d_1, d_2) = Cosine\left(\frac{\sum_{j=1}^{|W_{d_1}|} v_j(d_1)}{|W_{d_1}|}, \frac{\sum_{k=1}^{|W_{d_2}|} v_k(d_2)}{|W_{d_2}|}\right) \tag{3.6}$$

Where $v_j\,(d_1)$ and $v_k\,(d_2)$ are vector representations of jth and kth word in document d_1 and d_2, respectively. $|W_{d_1}|$ and $|W_{d_2}|$ are the number of words in d_1 and d_2, respectively. Here we remove the words in post and comments that do not exist in the pre-trained model.

2. *GloVe*: In word embedding based models, the corpus used for training vectors is an important issue, as the meaning of the vector representation of words will be different depending on the context and the semantics of the corpus in which words are represented. Therefore, we include the GloVe word embeddings pre-trained model in addition to the Google Word2Vec to see how a corpus can be effective in applying word embeddings similarity measures to identify related/unrelated content. The GloVe vectors were trained from 840 billion tokens of Common Crawl web data and have 300 dimensions [36]. This feature is extracted similar to the Word2Vec similarity by using equation 3.6 for each post and comment pair.

3. *Native context-word2vec*: We considered all the posts and following comments in our filtered BBC News dataset to train a word embeddings model using Word2Vec model, named Native context-word2vec. To extract word embeddings, we trained a neural network with a single hidden layer in our corpus, so that the weights of the hidden layer will be vector representation of words according to the Word2Vec approach in [30]. We used the Gensim library in Python to train our model with the Google Word2Vec toolkit [115]. The word embeddings similarity between each post and comment pair can then be estimated with equation 3.6.

3.4 Experiments and Results

In this section, we explain the collection and annotation process of the data used in this study. Then, we perform different experiments to investigate the performance of the proposed framework. At the end, we study the problem of toxicity in user-generated content on Facebook.

3.4.1 Dataset Description

We focused on Facebook news posts originated by news media pages. As a use-case, we identified one popular news agency on Facebook, BBC News because it is the world's largest broadcast news organization and it has global audiences around the world. The news posts and comments were collected using Python scraper for a two-month interval: 10th Dec 2017-20th Feb 2018, and Facebook Graph API Explorer was used to access the token and page id of the BBC News on Facebook. We gathered a total of 362 news posts and 398476 comments. Since the dataset is noisy, we filtered out some comments: those not in English, posts or comments that contain only pictures or videos, and comments with a length of fewer than 2 words. The filtered data, used in this study, contains 362 posts and 312291 comments. Our dataset is a bit large compared to those of previous studies on identifying related/unrelated content in social media [90, 102].

We should note that, at the time of this research, we could not find any study on Facebook focusing on related/unrelated content. There were a few existing works mainly focused on news webpages [102] or political blogs [90] in which the user behavior is deferent from Facebook as a social media platform, and they had different definitions of related/unrelated content (e. g., spam, comments aiming to divert the reader' s attention or point of views, etc.). In addition, there was not a publicly available dataset with a clear definition of unrelated content and data collection and annotation process to be aligned to our study. Therefore, we decided to collect and annotate a new dataset with a precise and clear definition of related/unrelated comments.

3.4.2 Gold Standard Annotation

Since labeling a comment as related/unrelated requires reading and comparing all posts and its following comments, we sampled 10% of posts (30 posts out of 300k sampled data) with all their comments using Stratified random sampling [116] that branches off the entire dataset into multiple non-overlapping homogeneous subgroups, and randomly chooses final members from the various subgroups as train dataset. In accordance with the distribution of comments (max and min number of comments, mean of all comments, and standard deviation) following all the posts, we observe that 5% of the posts have fewer than 164 comments and 5% of them have more than 2766. Therefore, we chose the fifth and ninety-fifth percentiles as criteria to create three subgroups. Table 3.2 lists the breakdown of the sampled posts. There are 2 posts in the sample dataset that have fewer than 164 comments, 26 posts that have number of comments between 164 and 2766 and 2 posts with more than 2766 comments. In this way, our sampling data is not biased to a specific kind of posts.

The sampling method produced 33,921 pairs of post and comments. We define comments

Table 3.2 – Data sampling

	Posts	#Sample
Subgroup 1	#comments<164	2
Subgroup 2	164<#comments<2766	26
Subgroup 3	#comments>2766	2

in which commentators are discussing the topic of a post or the topic of other comments following that post which are similar to the post's topic as related comments. These types of comments offer arguments and are similar to the post's content and therefore give readers some potentially good information. On the other hand, comments that contain contents merely to attract a reader's attention and do not have useful information are considered as unrelated comments. We have defined some main clues to select unrelated comments as follows:

1) Comments with advertising contents referring to websites, companies, or to a product advertising mechanism in social media. For example, using commercial URLs without any textual data or with texts that are unrelated to a post's content.

2) Comments with very little contents, that are very brief and without words in common with a post's content. This category includes comments that just show a commentator's sentiment in reaction to a post, such as "I love this" or "I hate that" and do not give readers any additional information related to the post content.

3) Comments in which commentators are only arguing with each other without discussing the topic of a post. These kinds of comments usually do not have a common context with the post.

4) Comments in which commentators are giving their opinion about a news agency page on Facebook and not about a post's content. Due to the high diversity of contents in Facebook [102], we consider these kinds of contents as unrelated and defined these clues to have a unique definition for labeling the train data.

The proposed model does not use the clues as specific features. However, some of these clues are reflected in the features that we defined in Section 3.1. For example, options 1 and 2 can be captured with cosine and native context features and if a comment has a few content with/without URLs which are not similar to the posts content then cosine and native similarities will be zero. On the other hand in options 3 and 4, semantic and topical similarities scores will be small.

The corpus is annotated by five graduate students as follows: First, two annotators conducted a labeling process of two separated sets of 15 posts (among 30 sampled posts) and all their following comments. Next, 3000 pairs of posts and comments, which were annotated before, were randomly sampled and given to three other annotators to annotate again.

Finally, the accuracy of the labels annotated by the first two annotators was estimated based on the three other labels. We selected a label for each sample (3000 pairs of posts and comments) using the majority vote among the three annotators' labels and then compared that label with the first two annotators' labels. This comparison results in a 6.2% error rate, which shows the annotation process achieved a high level of trustworthiness. Therefore, we considered the first two annotators' labels as gold standard labels of the training corpus in the rest of the study.

Pre-Processing Before extracting features from posts and comments, they must be pre-processed. We eliminate comments with fewer than two words and all non-English texts. Since stop words such as a, the, etc., do not have much meaning in our application, we remove them from all post and comments. All post and comment sentences are tokenized to words, and then the lemma for each word is derived by using the NLTK package in Python [117].

3.4.3 Results

After extracting the features, mentioned in Section 3.3.1, they are taken to the Support Vector Machine (SVM) algorithm for learning a binary classifier on the train dataset. The average accuracy, precision, recall, and F1-measure are calculated based on k-fold cross-validation ($k = 10$) to evaluate the performance of the classifier. As our model consists of several features, first we conduct experiments by comparing our model to baselines that only utilize one feature or combine fewer features to investigate the impact of the combination of features in the performance of the model. We also compare our model with LDA as a most frequently used method for topic detection in previous studies to investigate the effect of using LF-LDA in comparison with LDA. Finally, to evaluate the performance of our model in comparison with previous studies, we use a proposed model by Xie et al. [90].

The performance metrics evaluation is reported in Table 3.3, in which it is shown that the proposed model with a combination of all features obtains 86% accuracy on average and it outperforms all other combination of features. We analyze classification results by eliminating each category of features and it indicates that eliminating the lexical category has a small effect on reducing the accuracy of the model (W/O Lexical column in Table 3.3). The accuracy of the model without lexical features is 85% because these features can not capture related words with different lexical context and semantics of context in which words are represented. On the other hand, eliminating the semantical category (W/O Semantical column in Table 3.3) has the most effect on the accuracy of the model. The accuracy of the model without the semantical category will be 74% because these features play the main role in including context-based semantics to the model especially by using the word embeddings method. Eliminating topical category has also effect on the efficiency of the

Table 3.3 – Performance of different feature combinations.

	Accuracy	**Precision**	**Recall**	**F1-Measure**
All features	86.1	85.5	84.4	84.9
W/O Lexical	85.3	85.4	83.5	84.4
W/O Topical	84.3	85.7	84.5	85.0
W/O Semantical	73.9	65.6	75.1	70.0
Just Lexical	60.3	64.3	74.8	69.1
Just Topical	64.6	54.3	64.0	58.7
Just Semantical	82.4	85.4	83.4	84.3

W/O = exclude one kind each time; Just = include one kind each time

model since the accuracy reduces to 84% when the topical feature is eliminated.

To show the necessity of combining three categories, we examine the effect of each category alone in identifying related/unrelated comments following a post too. From Table 3.3 it is obvious that using lexical features only is not efficient in this problem because cosine and native-context similarities are incapable of matching a post with a comment if they have related meanings but different terms. Even applying only topical feature results in low accuracy. Among three categories, just semantical features give the high accuracy of 82.4% in identifying correct labels for each comment whereas it is still capable to be increased by involving other categories (all features).

LDA vs LF-LDA: To the best of our knowledge, we are the first to propose a combination of topical and word embedding-based approaches in identifying related/unrelated comments following a post on social media. Therefore, we examine the efficiency of our model with the LDA [97] as a baseline, which has been used in previous studies to find topical similarity between texts, and LF-LDA along with semantic-based features. According to our experiments, we set hyper-parameters α and β in both LDA and LF-LDA to 0.1 and 0.01 and the number of topics to 8. To set the number of topics, we tried out different values of k (4 to 15 due to the size of corpus), and selected the one that minimized the perplexity measure (based on likelihoods). For Native context word2vec the window size and embedding vector dimension are set to 5 and 100, respectively, and words with a frequency of less than 2 are eliminated. Table 3.4 shows the classification results using LDA or LF-LDA with the semantical category. Although lexical features make a little bit of change in the accuracy of the proposed model, based on Table 3.3, we do not consider it in the rest of the analysis.

The results show that LF-LDA can outperform LDA in combination with semantical features. The accuracy results from LDA along with semantical features is 82.7% whereas this value is 85.3% for LF-LDA among with semantical features. Because LF-LDA uses latent features resulting from Word2Vec pre-trained model to provide more sufficient information for topic distribution modeling. Therefore in LF-LDA, the coherence between topics is more

Table 3.4 – Impact of combining a topical approach with word embeddings on identifying related/unrelated contents

	LDA + Semantical	LF-LDA + Semantical
Accuracy(%)	82.7	85.3
Precision(%)	81.1	84.4
Recall(%)	84.0	83.5
F1-Measure(%)	82.5	84.4

Lexical features are not considered.

Table 3.5 – Impact of pre-trained word embeddings models on identifying related/unrelated contents

	Accuracy (%)
W All word embedding methods	86.1
W/O Word2Vec	69.2
W/O GloVe	74.3
W/O Native context-word2vec	80.1

W: include all word embeddings; W/O: exclude one kind each time

than LDA and more context-based semantic is included in the model through latent feature vector of words. Considering that we do not have access to the entire story of a post and any external web pages related to the post content specifically, LDA trains topic distributions based on our existing corpus. Whereas, LF-LDA uses a pre-trained model (Word2Vec) to leverage the latent feature vector of words for improving the topic distributions learned from our existing corpus.

Word embedding based features: We are using Word2Vec, GloVe, and Native context-word2vec in the semantical category. To see the effect of each word embedding methods in the accuracy of our model, we eliminate each of them from the set of features and evaluate the accuracy of the model. The result of this experiment is given in Table 3.5.

According to Table 3.5, using pre-trained Word2Vec model gives the highest accuracy among all word embeddings approaches because eliminating it from the set of features reduces the accuracy to 69.2% where eliminating GolVe pre-trained model reduces the accuracy to 74.3%. It shows that feature vector of words in pre-trained Word2Vec model have more context-based semantic to words from our existing corpus and it produces high quality word embeddings. We use posts and comments related to the BBC News agency page on Facebook and they have more common context and words with Google News corpus which is used to train Word2Vec model. Therefore, eliminating this feature has a negative effect on capturing semantic between posts and comments and reduces the accuracy of the model. On the other hand, eliminating Native context word2vec has the lowest effect on the accuracy because our corpus, posts and all comments, is small and provides insufficient

Table 3.6 – Performance metrics evaluation in different approaches

	the proposed method	**Xie et al. [90]**
Accuracy(%)	86.1	76.5
Precision(%)	85.5	74.0
Recall(%)	84.4	77.8
F1-Measure(%)	84.9	75.8

information for Word2Vec training model to extract the underlying feature vector of words robustly. By using Native context word2vec we can alleviate missing words from two previous pre-trained models because Native context word2vec model trains a feature vector for each word in the corpus according to its context and semantic.

Previous research: Xie et al. [90] proposed a model to derive context-dependent (i.e. context-aware) semantics of short comments and detect short irrelevant texts. They leveraged both native context and transferred contexts, the neighboring comments on a specific topic instead of all comments in the corpus, based on LDA topic similarity between articles and following comments. To compare our model with this study, we crawled the entire story of each post in train dataset from the BBC news agency webpage and applied context-aware approach proposed in [90], the results are shown in Table 3.6.

From 3.6, we observe that our proposed method performs better in terms of evaluation metrics. As context-aware approach proposed by Xie et al. [90] represents comments and the whole content of the post just by building vectors based on term frequencies and then applies matrix factorization to build topics, they can not include the semantic behind the related but different words in their model. Therefore, it causes to lower precision and recall. In addition, the lower precision in Xie et al. [90] approach shows that using LDA alone without word vector embeddings extracted from semantic relation between words in both total comments and pre-trained word embedding models, leads to more false positive rate in identifying related/unrelated comments.

3.4.4 Case Study

We apply the learned classifier on the rest of our dataset (278,370 pairs of posts and comments) to predict their labels as *related/unrelated* comments written to the post. The classifier's result shows 41% of all comments are related and 59% of them are unrelated. This is an interesting observation that shows around 60% of the written comments to the posts in a news agency account are not related to the actual post in terms of the topic of discussion. This huge number of unrelated comments potentially biases a lot the readers perspective on the posts and provides a large noise on the available users' feedback. By analyzing the distribution of related/unrelated comments across the posts, we observe that

Table 3.7 – Four sampled posts from BBC news agency page on Facebook.

Posts	Text
post 1	School pupils read out some of the worst comments they've seen posted online for Safer Internet Day.'BBC Own It' is a new website to help young people stay safe online and navigate their digital lives with confidence.
post 2	Indian police have arrested a man who allegedly shot dead his neighbor by mistake at a pre-wedding party.
post 3	US President Donald Trump has sparked a backlash from UK politicians by attacking the National Health Service.
post 4	Who says make-up is just for girls?? South Korean men spend more on beauty and skincare than anywhere else in the world. Take a look at their quest to challenge beauty standards.

news posts containing a specific action or speech of popular people in a specific time have more unrelated comments than the posts which are announcing a fact or telling a story of daily events.

To investigate how the content of related and unrelated comments are different from the topic of the posts, how they are spreading during the lifetime of posts, and how they are similar to each other we analysis 4 randomly chosen posts with all their comments (after applying the learned classifier) as follows:

Content analysis of written comments under a post: To understand better the relation of written comments to the posts, we sampled 4 posts randomly and investigated the discussed topics on each two group of identified comments (related/unrelated). The texts of sampled posts are shown in Table 3.7. Post 1 is mainly related to students, young people, and their usage of safe internet. Post 2 and Post 4 are announcing some daily events or facts and post 3 is related to a political issue. We create a word cloud from related and unrelated comments following the 4 selected posts to show which topics are more discussed among related and unrelated comments in each post depicted in Figures 3.2 and 3.3.

By considering the word cloud from related comments shown in Figure 3.2, it is obvious that users are discussing explicit subjects related to the topics of each post. For example in post 1, the most frequently used words in related cluster are "children", "Kid", "parent", "school", "internet", "bullying", "social media", etc. which are mainly discussing the topic of post 1 and they give readers significant information related to the post. Or in post 2, people are using words such as "people", "Indian", "gun", "celebration", "wedding", "culture" and etc. in their comments. For post 3, the words in larger size such as "Trump", "NHS", "people", "government", "healthcare", "insurance", "hospital", etc. are closely related to the topic of post 3. Finally, in the word cloud of related comments written under post 4, users are using "men", "women", "makeup", "wear", "look like", etc. words more frequently in their

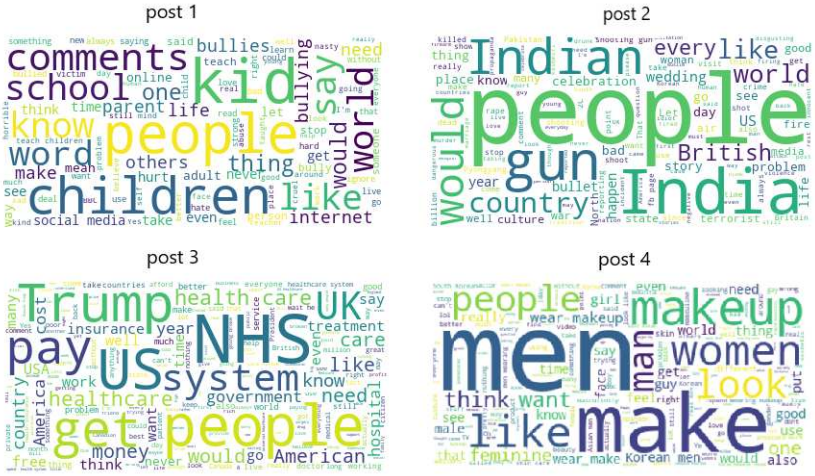


Figure 3.2 – WordCloud of related comments following the sampled posts; the more important a word makes the larger its size.

comments to discuss the topic of post 4. Since readers are more interested in reading strictly on-topic information from the comment section, filtering the related cluster for each post can be very useful and informative to users.

By investigating the word clouds from unrelated comments of the four sampled posts in Figure 3.3, we observed different kinds of unrelated comments written under the posts. For example words such as “love”, “sad”, “right”, “wrong”, “worse”, “oh”, “idiot”, “stupid”, “lol”, etc. are more frequently used words in unrelated comments. This observation shows that users are mostly expressing their opinion or point of view related to the posts’ entities (here India, Trump, Korea as the posts’ content are mostly about them) or other comments written by users which do not have significant information for readers because they do not discuss the topic of posts. Another interesting observation is that some most frequently used words such as “snowflakes” in post 1 are completely far from the topic of the post and they come from unrelated comments such as advertisements or bot-generated contents. For example in post 1, we observed that there are some comments in unrelated cluster that were advertising about “Amazing Macro Photographs of Snowflakes”. On the other hand in post 2, a lot of comments are targeting BBC news agency in Facebook since the words “News” and “BBC” are one of the most frequent used words in the word cloud from unrelated cluster.

Analyzing the content of related/unrelated written comments under the posts shows that most of related comments are objective and more topically coherent with posts’ content in terms of topics whereas unrelated comments usually contain subjective and very general words expressing users’ feedback without any focus on the subject of the posts. In unrelated cluster the most frequent words are not mainly related to the posts’ topics and commentators

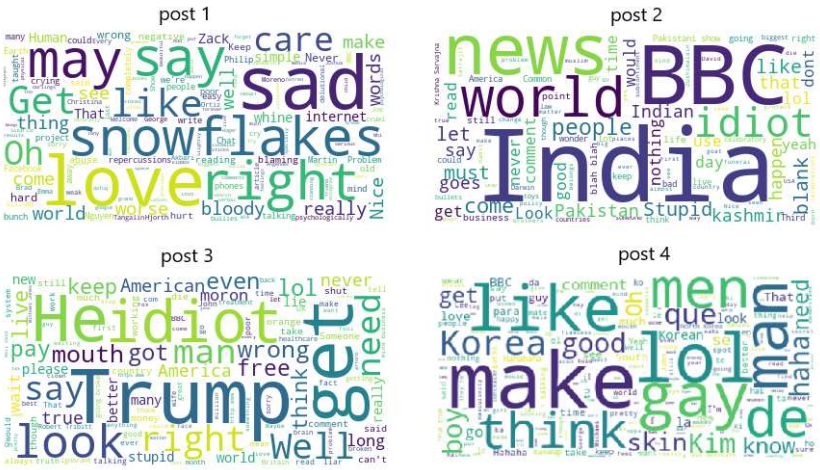


Figure 3.3 – WordCloud of unrelated comments following the sampled posts; the more important a word makes the larger its size.

are generally discussing similar topics which show personal feelings or opinions, or they are arguing about news agency itself. There are also some completely unrelated comments under posts that may be generated by users or bots for advertising or spreading information across different posts on Facebook that our model could identify them correctly. Since this type of comments are not informative and maybe readers are not interested in reading such off-topic information, it is better to identify and filter out these unrelated comments.

Timestamp analysis of written comments under posts: To see how users are disseminating related/unrelated comments under posts, we first look at the distribution of related/unrelated comments within a period of 24 hours after publishing each post (on the rest of our dataset: 278,370 pairs of posts and comments). For each comment following a post, the difference between a timestamp when the post was uploaded and the timestamp of the written comment is considered. Figure 3.4 depicts the portion of related/unrelated comments written under posts within the first 24 hours. It is evident that the portion of unrelated comments written under all posts, in general, are more than related one in the first hours after publishing posts however the number of written comments under each post are diverse and we can not say that this evidence is true through all posts.

To go more deeply into this subject and see how related/unrelated comments are spreading per post, we look at the portion of related/unrelated comments following each sampled post based on their written time within a period of 12 hours. Figure 3.5 depicts the portion of related/unrelated comments written under posts during the first 12 hours after each post's creation time.

As it is obvious from Figure 3.5, there is not a specific pattern among all posts in

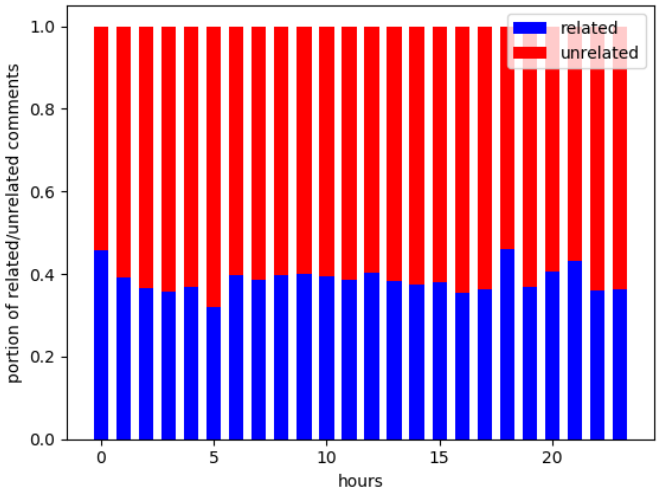


Figure 3.4 – Distribution of related/unrelated comments following all the posts within a period of 24 hours.

spreading related/unrelated comments. However, an interesting observation is that in some posts such as post 1 and post 2, the portion of related comments are more than unrelated comments in the first hours. Then by passing the time, the portion of unrelated comments increases. Whereas, in post 3 and post 4 the portion of unrelated comments are more than related comments over the period of 12 hours. By considering the text of sampled posts (Table 3.7), it can be inferred that the topic of a post plays an important role in the content of the following comments. For example, the topic of two first posts are about a scientific context or daily event, commentators are more discussing the topics in the first hours. Whereas in the two last posts, commentators are posting unrelated comments more than related comments in the first hours since the topics of post 3 and post 4 are more attractive to different users in terms of topics; they are related to politic and gender issues. A lot of users come to these hot topic posts to just show their feeling by putting uninformative comments or attract other users' attention by putting advertisements or off-topic comments to the post.

Similarity within related/unrelated comments: Another aspect that we aim to study is to understand the similarity degree of comments inside related/unrelated clusters. To see how similar a comment is to other comments following a post, based on word feature vector similarity, we extract comments with a degree of similarity more than 90% to another comment following the same post. The result shows that only 0.4% of comments in related cluster and 0.7% of comments in unrelated cluster have degree of similarity more than

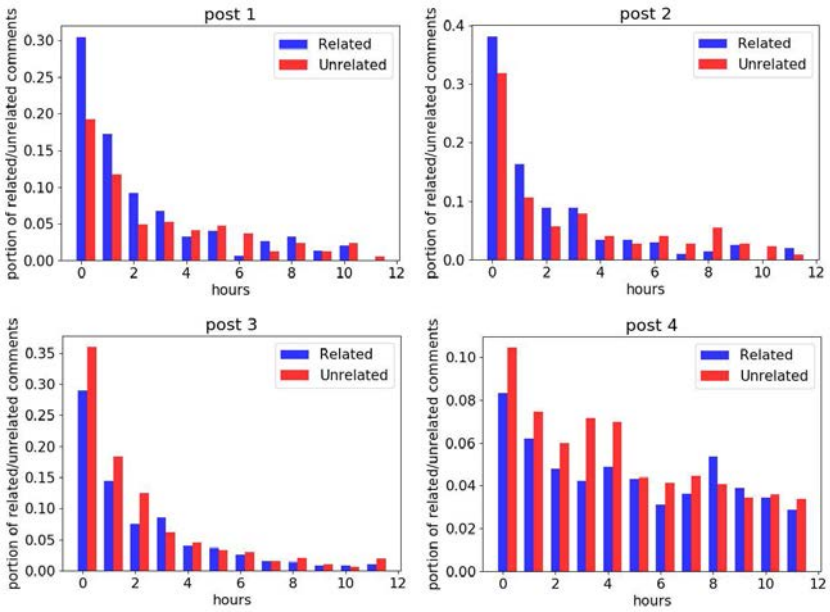


Figure 3.5 – The portion of related/unrelated comments written under 4 sampled posts within the first 12 hours.

90% to at least another comment. To go deeper into details and see when these similar comments are published, we explore unrelated comments in sampled posts. In average 0.8% of unrelated comments in the sampled posts are similar to each other with a degree of similarity more than 90%. By checking these types of comments, we find that they are frequently duplicate comments posted by users within a duration in seconds. In addition, they are also short texts with common words. Since the number of these types of comments are very low, in general, they cannot be generated for a specific purpose by bots. It can be inferred that users are posting this kind of duplicate content to emphasize their feedback and feeling or it happens during the commenting process in social media with their faults.

Characterizing toxicity in comments: Commenting environments in Facebook pages of news outlets are a potential home for writing and spreading uncivil, insulting, and hostile content [118]. Hence, in this section, we provide a brief insight into the content of BBC page's comments in terms of toxicity. We use a publicly available tool named Perspective API[2] created by Jigsaw and Google's Counter Abuse Technology team in Conversation-AI in 2017. Perspective API uses different machine learning models to identify and filter online insults, harassment, and abuse on social media. Given a comment as input, it returns a *score* from 0 to 1 that shows the probability of toxicity regarding the similar toxic comments previously seen by model. Comments with high probability of toxicity get a higher

[2] `https://www.perspectiveapi.com/`

Table 3.8 – Definition of toxicity attributes from perspective API.

Attributes	**Definition**
Toxicity	A rude, disrespectful, or unreasonable comment that is likely to make people leave a discussion.
Severe Toxicity	A very hateful, aggressive, disrespectful comment or otherwise very likely to make a user leave a discussion or give up on sharing their perspective. This attribute is much less sensitive to more mild forms of toxicity, such as comments that include positive uses of curse words.
Insult	Insulting, inflammatory, or negative comment towards a person or a group of people.
Profanity	Swear words, curse words, or other obscene or profane language.
Identity Attack	Negative or hateful comments targeting someone because of their identity.
Threat	Describes an intention to inflict pain, injury, or violence against an individual or group.

score. More specifically, by defining toxicity as "a rude, disrespectful, or unreasonable comment that is likely to make one leave a discussion", Perspective provides scores for other attributes including severe toxicity, insult, profanity, identity attack, and threat. Table 3.8 describes the exact definition of each attribute that we report from Perspective webpage[3].

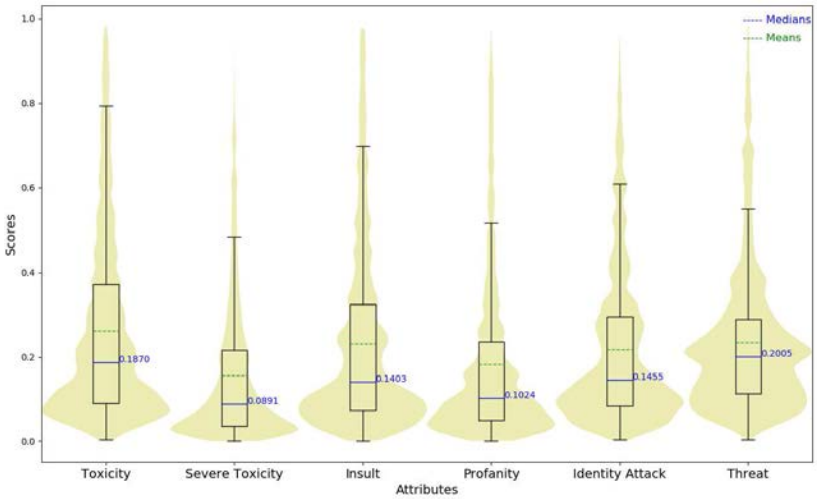


Figure 3.6 – Distribution of toxicity scores for different attributes derived from Perspective API.

We use all attributes mentioned in Table 3.8, to analyze different levels of toxicity in the comments of the BBC page on Facebook . For each comment in the dataset, we extract the probability of being similar to each attribute using Perspective API. To better explain the distribution of toxicity scores related to each attribute among comments, we generated

[3] `https://developers.perspectiveapi.com/s/about-the-api-attributes-and-languages`

violin plots depicted in Figure 3.6. Comparing the distribution of different attributes shows that threat is the most popular type of toxicity among comments where its median is 0.2005, however, severe toxicity with median 0.0891 is the least one. Considering wider sections of the plots, we can see that a very large number of comments have scores in range [0.0-0.4] where most of the values clustered around the median of each attribute. However, the skinnier sections are mostly in range [0.4,1.0] and indicate that the level of toxicity varies in comments (regarding attributes) and there is a considerable amount of comments with a high score for different attributes in our dataset; which signifies the necessity of having an accurate algorithm for detecting and eliminating such content.

Regarding the definition of attributes, we can understand that some concepts are common among attributes. For example, both thread and insult potentially target an individual or group as well as identity attack that targets individuals based on their identity. Therefore, having a brief definition of toxic content in general and hate speech in particular is difficult and challenging.

3.5 Conclusion

In this chapter, we built a model to identify related and unrelated comments to the corresponding posts on Facebook by considering the content of the comments. The framework consisted of three categories of features: lexical, topic, and semantic. To be independent of the entire story of a post or external webpage contents related to the post, we used a combination of word embeddings in both topical and lexical features. The results showed that the model can identify related/unrelated comments written to the posts with more than 85% accuracy. We next investigated the distribution of the related/unrelated comments across the posts and also looked to the main discussed topics in each cluster to get a better understanding of the phenomena of unrelated comments in social media. Furthermore, the toxicity analysis of comments indicated that hatred and toxicity are a common phenomena in social media which is worthy of further investigation.

In the next chapter, we will mainly focus on automatically identification of hate speech content based on a Transfer Learning approach and analysis of potential racial bias in the data and model.

Chapter 4

Monolingual Hate Speech Detection

Contents

4.1 Overview **68**

4.2 A BERT-Based Transfer Learning Approach for Hate Speech Detection **68**

4.2.1 Introduction 68

4.2.2 Related Work 70

4.2.3 BERT-Based Hate Speech Detection Module 72

4.2.4 Experiments and Results 75

4.2.5 Conclusion 87

4.3 Racial Bias Mitigation in Social Media based on BERT Model **87**

4.3.1 Introduction 87

4.3.2 Related Work 88

4.3.3 Bias Mitigation Module 90

4.3.4 Discussion and Challenges 97

4.3.5 Conclusion 99

4.4 Summary and Discussion **100**

4.1 Overview

Hateful and toxic content generated by a portion of users in social media is a rising phenomenon that has motivated researchers to dedicate substantial efforts to the challenging direction of hateful content identification. We need not only an efficient automatic hate speech detection model based on advanced ML and NLP techniques, but also a sufficiently large amount of annotated data to train a model. On the other hand, disparate biases associated with datasets and trained classifiers in hateful and abusive content identification tasks have raised many concerns recently [87,119]. Therefore, the lack of a sufficient amount of labeled hate speech data, along with the existing biases, have been the main issue in this domain of research.

In this chapter, we tackle the problem of hate speech detection and racial bias mitigation in online social media in a monolingual setting in which we use only English datasets to train our model. In Section 4.2, we introduce a novel transfer learning based approach leveraging the pre-trained language model BERT to identify hate speech. In addition, we introduce a bias alleviation mechanism to mitigate the effect of bias in training set during the fine-tuning of our proposed BERT-based model for hate speech detection in Section 4.3.

4.2 A BERT-Based Transfer Learning Approach for Hate Speech Detection

4.2.1 Introduction

People are increasingly using social networking platforms such as Twitter, Facebook, YouTube, etc. to communicate their opinions and share information. Although the interactions among users on these platforms can lead to constructive conversations, they have been increasingly exploited for the propagation of abusive language and the organization of hate-based activities [70, 80], especially due to the mobility and anonymous environment of these online platforms. Violence attributed to online hate speech has increased worldwide. For example, the US has been a marked increase in hate speech and related crime following the Trump election[1]. Therefore, governments and social network platforms confronting the trend must have tools to detect aggressive behavior in general, and hate speech in particular, as these forms of online aggression not only poison the social climate of the online communities that experience it, but can also provoke physical violence and serious harm [80].

Recently, the problem of online abusive detection has attracted scientific attention. Proof of this is the creation of the third Workshop on Abusive Language Online[2] or Kag-

[1] Hate on the rise after Trump's election: `http://www.newyorker.com`
[2] `https://sites.google.com/view/alw3/home`

gle's Toxic Comment Classification Challenge that gathered 4,551 teams[3] in 2018 to detect different types of toxicities (threats, obscenity, etc.). Hate speech detection is not a stable or simple target because misclassification of regular conversation as hate speech can severely affect users' freedom of expression and reputation, while misclassification of hateful conversations as unproblematic would maintain the status of online communities as unsafe environments [87].

To detect online hate speech, a large number of scientific studies have been dedicated by using NLP in combination with ML and DL methods [2,4,19,70,79,84]. Although supervised machine learning-based approaches have used different text mining-based features such as surface features, sentiment analysis, lexical resources, linguistic features, knowledge-based features, or user-based and platform-based metadata [3, 120, 121], they necessitate a well-defined feature extraction approach. The trend now seems to be changing direction, with deep learning models being used for both feature extraction and the training of classifiers. These newer models are applying deep learning approaches such as CNNs, LSTMs, etc. [70, 84] to enhance the performance of hate speech detection models, however, they still suffer from lack of labeled data or inability to improve generalization property.

In this section, we propose a transfer learning approach for hate speech understanding using a combination of the unsupervised pre-trained model BERT [6] and some new supervised fine-tuning strategies. As far as we know, it is the first time that such exhaustive fine-tuning strategies are proposed along with a generative pre-trained language model to address the problem of hate speech detection and improve performance of the task. Our main contributions are:

- We propose a transfer learning approach using the pre-trained language model BERT learned on English Wikipedia and BookCorpus to enhance hate speech detection on publicly available benchmark datasets. Toward that end, we introduce new fine-tuning strategies to examine the effect of different embedding layers of BERT in hate speech detection task.

- We conduct a comprehensive experiment to inspect the impact of our transfer learning approach in a shortage of labeled data and in capturing syntactical and contextual information of BERT embeddings.

- Our experiment results show that using the pre-trained BERT model and fine-tuning it on the downstream task by leveraging syntactical and contextual information of all BERT's embeddings layers outperforms previous works in terms of F1-measure. Furthermore, examining the results shows the ability of our model to detect some

[3] https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/

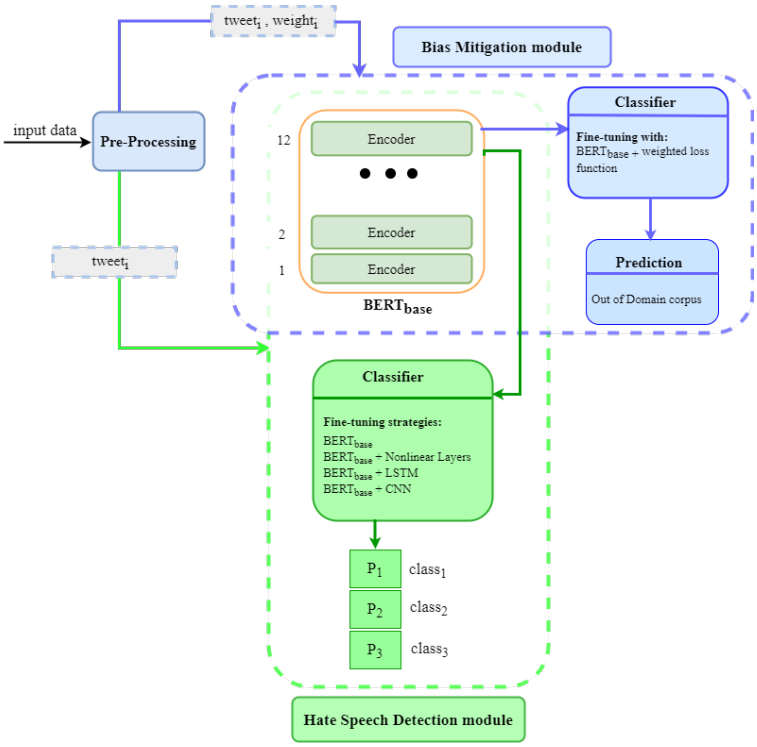


Figure 4.1 – The proposed framework for hate speech detection and bias mitigation tasks. It consists of two different modules: Hate Speech Detection and Bias Mitigation with different inputs as a result of different pre-processing approaches. The pre-trained BERT$_{base}$ is a common component between two modules that is fine-tuned differently in respect of each module's goal.

biases in the process of collecting or annotating datasets. It can be a valuable clue in using pre-trained BERT model for debiasing hate speech datasets in future studies.

As we mentioned in Section 4.1, in this chapter we tackle both the problem of hate speech detection and racial bias mitigation in online social media. We depict our proposed framework for hate speech detection and unintentional bias analysis and mitigation in Figure 4.1. The framework contains two main modules: **(1) Hate Speech Detection module** and **(2) Bias Mitigation module** that will be explained in Sections 4.2 and 4.3, respectively.

4.2.2 Related Work

Researchers have been studying hate speech on social media platforms such as Twitter [3], Reddit [122,123], and YouTube [124] in the past few years. The features used in traditional machine learning approaches are the main aspects distinguishing different methods, and

surface-level features such as bag of words, word-level and character-level n-grams, etc. have proven to be the most predictive features [2,19,79]. Apart from features, different algorithms such as Support Vector Machines [68], Naïve Bayes [80], and Logistic Regression [2,3], etc. have been applied for classification purposes.

As a baseline, Waseem et al. [2] addressed the problem of hate speech detection in Twitter by making a general definition of hateful content in social media based on guidelines inspired by Gender Studies and Critical Race Theory (CRT). Regarding that, they tried to annotate a corpus of 16,849 tweets as "Racism", "Sexism", and "Neither" by themselves, and the labels were inspected by a 25-year-old woman studying gender studies and a non-activist feminist for identifying potential sources of bias. To train their model, they used different sets of features such as word and character n-grams up to 4, gender, length, and location and investigated the impact of each feature on the classifier performance. Their results indicated that character n-grams are the most indicative features. Furthermore, Davidson et al. [3] studied hateful and offensive contents in Twitter by sampling and annotating a $24K$ corpus of tweets as "Hate", "Offensive", and "Neither". They developed a variety of multi-class classifiers such as Logistic Regression, Naïve Bayes, Decision Trees, Random Forests, etc. on a set of features including Term Frequency–Inverse Document Frequency (TF-IDF), weighted n-grams, Part Of Speech (POS) tagging, sentiment scores, some tweet-level metadata such as the number of hashtags, mentions, re-tweets, and URLs, etc.. Although their results illustrated that Logistic Regression with L2 regularization performs the best in terms of accuracy, precision, and F1-measure, there are some social biases regarding anti-black racism and homophobia in their algorithm. Malmasi et al. [68] proposed an ensemble-based system that uses some linear SVM classifiers in parallel to distinguish hate speech from general profanity in social media.

As one of the first attempts in neural network models, Djuric et al. [86] proposed a neural network-based model advantaging pagraph2vec embeddings to distinguish between hate speech and clean content. The proposed model incorporated two steps: in the first step, paragraph2vec embeddings were extracted from a continuous bag of words model, and in the second hateful and non-hateful contents were identified by applying a binary classifier counting on the extracted embeddings. Badjatiya et al. [70], who experimented on the dataset provided by Waseem et al. [2], investigated three deep learning architectures: FastText, CNN and LSTM. They used a combination of randomly initialized or GloVe-based embeddings with an LSTM neural network and a gradient boosting classifier. Their results outperform the baseline provided in [2]. Different feature embeddings such as word embeddings and character n-grams were defined by Gambäck et al. [84], to solve the problem of identification of hate speech based on a CNN model. Afterward, a CNN+GRU (Gated Recurrent Unit network) neural network model was proposed by Zhang et al. [4] in

which the model captured both word/character combinations (e. g., n-grams, phrases) and word/character dependencies (order information) with employing a pre-trained Word2Vec embeddings. Using raw texts and domain-specific metadata from Twitter, Founta et al. [81] proposed a unified classification model at which different types of abusive language such as cyberbullying, hate, sarcasm, etc. were efficiently performed.

For the first time, Waseem et al. [121] applied a multi-task learning strategy as a transfer learning model to transfer knowledge between two different hateful and offensive datasets with solving two hate speech detection tasks simultaneously and utilizing similarities between these two tasks as auxiliary and primary. Their results indicated the ability of multi-task learning to generalize to new datasets and distributions in hate speech detection. Afterward, using a combination of GloVe word embeddings [36] and Embedding from Language Models (ELMO) [37], Rizoiu et al. [125] proposed a transfer learning approach for hate speech and abusive language detection by using two datasets provided in [2, 3]. To adjust the ELMO representation to the hate speech detection domain, they applied a bi-LSTM layer independently trained left-to-right and right-to-left on both tasks simultaneously and then extracted sentence embedding using a max-pooling approach. At the end, a specific classifier was trained for each task. Due to the jointly solving both tasks, the insights learned from one task can be transferred to the other task. Comparing the results from these two transfer learning-based studies indicates that the approach of Waseem et al. [121] outperforms Rizoiu et al. [125], therefore we consider their approach as our baseline here and compare our proposed method with that.

4.2.3 BERT-Based Hate Speech Detection Module

This section is dedicated to the first module of our framework depicted in Figure 4.1. First we detail the BERT model and its objectives and then explain the proposed fine-tuning strategies.

According to Figure 4.1, given tweets in training set as input data, the pure texts of them are extracted from the pre-processing component regarding a set of specific rules, described in the related subsection. Then, the processed tweets are fed into the pre-trained BERT model to be fine-tuned according to different strategies with task-specific modifications. At the end, using the trained classifiers we predict the labels of the test set and evaluate the results.

4.2.3.1 BERT

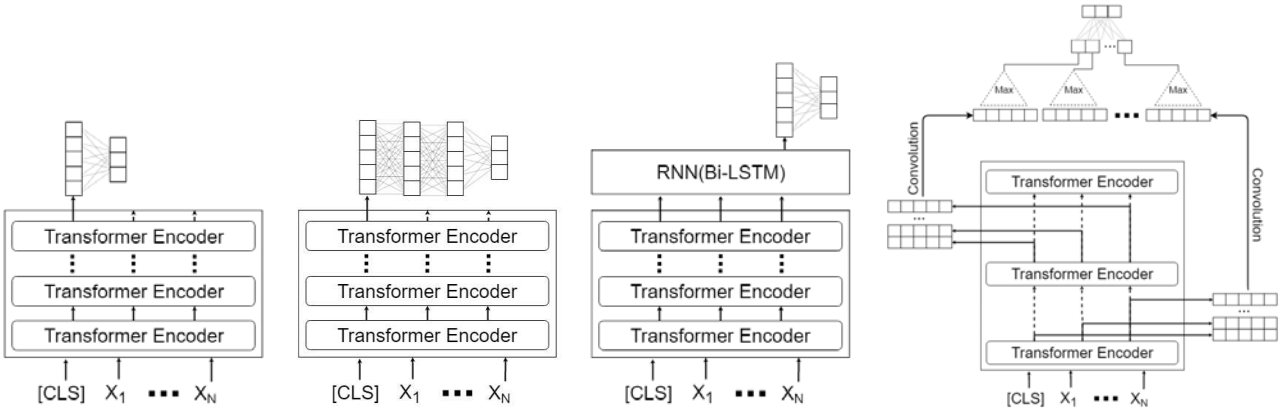
BERT is a multi-layer bidirectional transformer encoder trained on the English Wikipedia and the Book Corpus containing 2,500M and 800M tokens, respectively, and has two models named $\text{BERT}_{\text{base}}$ and $\text{BERT}_{\text{large}}$. $\text{BERT}_{\text{base}}$ contains an encoder with 12 layers (transformer

blocks), 12 self-attention heads, and 110 million parameters whereas $\text{BERT}_{\text{large}}$ has 24 layers, 16 attention heads, and 340 million parameters. Each of $\text{BERT}_{\text{base}}$ and $\text{BERT}_{\text{large}}$ has two versions: uncased and cased where uncased version has only lowercase letters. In this study, we use the uncased version of the pre-trained $\text{BERT}_{\text{base}}$ model. As the BERT model is pre-trained on general corpora, and for our hate speech detection task we are dealing with social media content, therefore as a crucial step, we have to analyze the contextual information extracted from BERT' s pre-trained layers and then fine-tune it using annotated datasets. By fine-tuning we update weights using a labeled dataset that is new to an already trained model. A sequence of tokens, as a pre-processed sentence, in maximum length 512 is fed to the BERT model as input. Then two segments are added to each sequence as [CLS] and [SEP] by BERT tokenizer. [CLS] embedding which is the first token of the input sequence, is used as a classification token since it contains specific classification information in each layer. The [SEP] token, an artifact of two-sentence tasks, separates segments and we will not use it in our classification because we have only single-sentence inputs. As the output, BERT produces a 768-dimensional vector to represent each input sequence. To perform the hate speech detection task, we use $\text{BERT}_{\text{base}}$ model to classify each tweet as Racism, Sexism, Neither or Hate, Offensive, Neither in our datasets. In order to do that, we focus on fine-tuning the pre-trained $\text{BERT}_{\text{base}}$ parameters. By fine-tuning, we mean training a classifier with different layers of 768 dimensions on top of the pre-trained $\text{BERT}_{\text{base}}$ transformer to minimize task-specific parameters.

4.2.3.2 Fine-Tuning Strategies

As we are dealing with textual content from social media in our task and the BERT model is pre-trained on general corpora, it is crucial to analyze the contextual information extracted from pre-trained BERT' s transformer layers. Different levels of syntactic and semantic information are encoded in different layers of the BERT model, and according to [6] the lower layers of the BERT model may contain information that is more general whereas the higher layers contain task-specific information. Hence, we need to fine-tune it on our hate speech detection task with annotated datasets. Here, four different fine-tuning approaches are implemented that exploit pre-trained $\text{BERT}_{\text{base}}$ transformer encoders for our classification task. In the fine-tuning phase, the model is initialized with the pre-trained parameters and then are fine-tuned using the labeled datasets. Different fine-tuning approaches on the hate speech detection task are depicted in Figure 4.2, in which X_i is the vector representation of token i in a tweet sample, and are explained in more detail as follows:

1. BERT based fine-tuning: In the first approach, which is shown in Figure 4.2a, very few changes are applied to the $\text{BERT}_{\text{base}}$. In this architecture, only the [CLS] token output provided by BERT is used. The [CLS] output, which is equivalent to the [CLS]



(a) $BERT_{base}$ fine-tuning (b) Insert nonlinear layers (c) Insert Bi-LSTM layer (d) Insert CNN layer

Figure 4.2 – Fine-tuning strategies

token output of the 12th transformer encoder, a vector of size 768, is given as input to a fully connected network without hidden layer. The softmax activation function is applied to the hidden layer to classify.

2. Insert nonlinear layers: Here, the first architecture is upgraded and an architecture with a more robust classifier is provided in which instead of using a fully connected network without hidden layer, a fully connected network with two hidden layers in size 768 is used. The first two layers use the Leaky Rectified Linear Unit (Relu) activation function with negative slope = 0.01, but the final layer, as the first architecture, uses softmax activation function as shown in Figure 4.2b.

3. Insert Bi-LSTM layer: Contrary to the previous architectures that only use [CLS] as the input for the classifier, in this architecture all outputs of the latest transformer encoder are used in such a way that they are given as inputs to a bidirectional recurrent neural network (Bi-LSTM) as shown in Figure 4.2c. After processing the input, the network sends the final hidden state to a fully connected network that performs classification using the softmax activation function.

4. Insert CNN layer: In this architecture shown in Figure 4.2d, the outputs of all transformer encoders are used instead of using the output of the latest transformer encoder. So that the output vectors of each transformer encoder are concatenated, and a matrix is produced. The convolutional operation is performed with a window of size (3, hidden size of BERT which is 768 in $BERT_{base}$ model). Then, by applying a MaxPooling method on the convolution's outputs, the maximum values of each transformer encoder are extracted, and a vector is generated to be fed as input to a fully connected neural network. In the end, the classification function is performed by applying a softmax activation function.

4.2.4 Experiments and Results

This section details the datasets used in our study and then investigates the different fine-tuning strategies for hate speech detection task. Furthermore, we include the details of our implementation and error analysis in the respective subsections.

4.2.4.1 Dataset Description

In this study, we experiment with three widely-studied public datasets from Twitter provided by Waseem and Hovy [2], Waseem [65] and Davidson et al. [3], which are detailed in the following:

Waseem and Hovy [2]/Waseem [65]: Within two months period, Waseem and Hovy [2] collected 136,052 tweets from Twitter and, after some filtering, annotated a corpus containing 16,914 tweets as "Racism", "Sexism", and "Neither". First using an initial ad-hoc approach, they tried to search common slurs and terms related to religious, sexual, gender, and ethnic minorities. Secondly, from the first results, they identified the most frequent terms in tweets containing hate speech. For example, hashtag "#MKR" which was related to a public Australian TV show, My Kitchen Rules, and caused many sexist tweets directed at the female participants. At the end to make their sampling process more general, they crawled more tweets containing clearly abusive words and potentially abusive words but they are not abusive in context, as negative sampling. The final collected corpus ($16K$) was annotated by experts and ascertained by a 25 years old woman studying gender studies and non-activist feminist to reduce annotator bias. Waseem [65] also provided another dataset to investigate the impact of expert and amateur annotators on the performance of classifiers trained for hate speech detection. Therefore, they collected 6,909 tweets for hate speech and annotated them as "Racism", "Sexism", "Neither", and "Both" by amateurs from CrowdFlower crowdsourcing platform and experts having a theoretical and applied knowledge of the abusive language and hate speech. Their efforts resulted in a set of 4,033 tweets where there was an overlap of 2,876 tweets between their new dataset and the one provided by Waseem and Hovy [2]. Since both datasets are overlapped partially and they used the same strategy in definition of hateful content, we merged these two datasets following Waseem et al. [121] to make our imbalance data a bit larger (we followed all the rules provided in Section 3.2 of Waseem et al. [121] paper to merge two datasets. For more details, please refer to that paper). In the rest of this chapter, we refer to this aggregated dataset as **Weseem-dataset**.

Davidson et al. [3]: Employing a set of particular terms from a pre-defined lexicon of hate speech words and phrases, called HateBase[4], Davidson et al. [3] crawled 84.4 million

[4] `https://hatebase.org/`

Table 4.1 – Datasets description. The columns show the total number of tweets, the different categories and the percentage of tweets belong to each one in the datasets, respectively.

Dataset	**#Tweets**	**Classes and percentage of membership**
Waseem-dataset [2] [65]	19697	Racism (10.73%)
		Sexism (21.15%)
		Neither (68.12%)
Davidson-dataset [3]	24783	Hate (5.77%)
		Offensive (77.43%)
		Neither (16.80%)

tweets from 33,458 twitter users. To annotate collected tweets as "Hate", "Offensive", or "Neither", they randomly sampled $25k$ tweets and asked users of CrowdFlower crowdsourcing platform to label them. After labeling each tweet by annotators, if their agreement was low, the tweet was eliminated from the sampled data. In the rest of this chapter, we refer to this dataset as **Davidson-dataset**.

Table 4.1 shows a brief description of classes' distribution in both datasets.

Pre-processing For simplicity and generality, we consider the following criteria in order to filter the raw dataset and make it clean as the input of our model:

- Converting all tweets to lower case.
- Replacing mentions of users with token <user>, for the sake of protecting the user's identities.
- Replacing embedded URLs in tweets' content with token <url>
- Replacing numbers in tweets' content with token <number>
- Removing common emoticons, because in this study we do not consider emotions in our analysis.
- Identifying elongated words and converting them into short and standard format; for example, converting "yeeeessss" to "yes".
- Removing hashtag signs (#) and replacing the hashtag texts by their textual counterparts, where there is not any space between them; for example, we convert hashtag "#notsexist" to "not sexist".
- Removing all punctuation marks, unknown uni-codes and extra delimiting characters.
- Keeping all stop words, because our model trains the sequence of words in a text directly.
- Eliminating tweets with a length of less than 2 after applying all aforementioned pre-processing steps.

4.2.4.2 Implementation

For the implementation of hate speech detection module, we use publicly available pytorch-pretrained-bert library[5]. We utilize the pre-trained BERT model, text tokenizer, and pre-trained WordPiece provided in the library to prepare the input sequences and train the model. As an input, we tokenize each tweet with the BERT tokenizer. It contains invalid characters removal, punctuation splitting, and lowercasing the words. Following the original BERT [6], words are split to sub-words by employing WordPiece tokenization. Due to the shortness of input sentences' length, the maximum sequence length is set to 64 and in any case of shorter or longer length, it will be padded with zero values or truncated to the maximum length, respectively. We train our classifiers with different fine-tuning strategies with a batch size of 32 for 3 epochs on Google Colaboratory tool[6] with an NVIDIA Tesla K80 GPU and 12G RAM; as the implementation environment. During training, we use an Adam optimizer with a learning rate of 2e-5 to minimize the Cross-Entropy loss function. Furthermore, the dropout probability is set to 0.1 for all layers.

Evaluation metrics In general, classifiers with higher precision and recall scores are preferred in classification tasks. Regarding Table 4.1, we are dealing with imbalance datasets with various classes' distribution, and since hate speech and offensive language are real phenomena, we do not perform oversampling or undersampling techniques to adjust the classes' distribution and try to supply the datasets as realistic as possible. Therefore, due to the imbalanced classes in our datasets, we tend to make a trade-off between precision and recall measures. Hence, we summarize models' performance into macro averaged F1-measure, which is the geometric mean of precision and recall and gives more insights into the performance characteristics of each classification model.

4.2.4.3 Results

In this section, we investigate the impact of using a pre-trained BERT-based model with different fine-tuning strategies on the hate speech detection task. Additionally, we show different aspects of our transfer learning-based approach by analyzing the proposed model deeply.

We consider 80% of each dataset as training data to update the weights in the fine-tuning phase, 10% as validation data to measure the out-of-sample performance of the model during training, and 10% as test data to measure the out-of-sample performance after training. To prevent overfitting, we use stratified sampling to select 0.8, 0.1, and 0.1 portions of tweets from each class (racism/sexism/neither or hate/offensive/neither) for train, validation, and test. Classes' distribution of train, validation, and test datasets are shown in Table 4.2.

[5] `https://github.com/huggingface/pytorch-pretrained-BERT`
[6] `https://colab.research.google.com`

Table 4.2 – Dataset statistics for Waseem-dataset and Davidson-dataset. Splits are produced using stratified sampling to select 0.8, 0.1, and 0.1 portions of tweets from each class (racism/sexism/neither or hate/offensive/neither) for train, validation, and test samples, respectively.

	Racism	**Sexism**	**Neither**	**Total**
Train	1693	3337	10787	15817
Validation	210	415	1315	1940
Test	210	415	1315	1940
Total	2113	4167	13417	

(a) Waseem-dataset

	Hate	**Offensive**	**Neither**	**Total**
Train	1146	15354	3333	19832
Validation	142	1918	415	2475
Test	142	1918	415	2475
Total	1430	19190	4163	

(b) Davidson-dataset

We consider models proposed by Davidson et al. [3] and Waseem et al. [121] as our baselines in which a classical machine learning method and a deep neural network model are created respectively. To do so, following the original work [3], we create an SVM classification method proposed by the authors and we train a machine learning model using a multi-task learning framework proposed by Waseem et al. [121]. In addition to these two baselines, we compare our results with the methods proposed in [2, 4, 5, 69] on the corresponding datasets. Using two hate speech datasets, we examine the performance of our model, with different fine-tuning strategies, in contrast to the baselines and state-of-the-art approaches. The evaluation results on the test sets of two datasets are reported in Tables 4.3 and 4.4, in terms of macro averaged F1-measure. The differences between some results provided in these Tables and what were reported in the original works are due to we implemented some models and report macro averaged F1-measures (the source code was not made public with authors).

Table 4.3 – The performance of different trained classifiers on Waseem-dataset in terms of F1-measure.

Model	**F1-Measure**
Waseem and Hovy [2]	75
Waseem et al. [121]	80
Zhang et al. [4]	82
Park et al. [5]	83
$\text{BERT}_{\text{base}}$	81
$\text{BERT}_{\text{base}}$ + Nonlinear Layers	76
$\text{BERT}_{\text{base}}$ + bi-LSTM	86
$\text{BERT}_{\text{base}}$ + CNN	88

Table 4.4 – The performance of different trained classifiers on Davidson-dataset in terms of F1-measure.

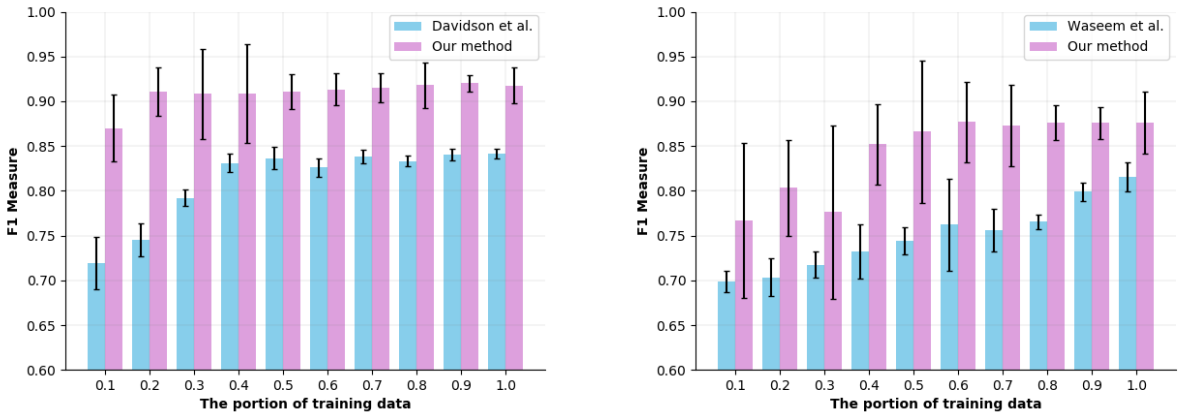
Model	**F1-Measure**
Davidson et al. [3]	84
Zhang et al. [4]	94
Waseem et al. [121]	89
MacAvaney et al. [69]	77
$\text{BERT}_{\text{base}}$	91
$\text{BERT}_{\text{base}}$ + Nonlinear Layers	87
$\text{BERT}_{\text{base}}$ + bi-LSTM	92
$\text{BERT}_{\text{base}}$ + CNN	92

The results show that, in both datasets, all the BERT-based fine-tuning strategies except BERT + nonlinear classifier on top of it outperform the existing approaches or they achieve competitive results. According to Table 4.3 , on Waseem-dataset, the highest F1-measure value is achieved by $\text{BERT}_{\text{base}}$ + CNN which is 88% and there is a 5% improvement from the best performance achieved by Park et al. [5] method. In addition, applying different models on Davison-dataset, reported in Table 4.4 , also confirms the previous observation and shows that using the pre-trained BERT model as initial embeddings and fine-tuning the model with a CNN yields the best performance in terms of F1-measure; where it is 92%. On Davidson-dataset, comparing the best F1-measure value achieved by $\text{BERT}_{\text{base}}$ + CNN model with the best-performed model proposed by Zhang et al. [4] indicates that our model achieved a 2% decrease in performance than [4]; where the F1-measure is 94%. We posit that this is due to the fact that Zhang et al. [4] have merged the Hate and Offensive classes of Davidson-dataset together and solved the problem of hate speech detection as a binary classification which it made the task more simplified counter to our specific multi-class classification approach.

From deep learning neural network perspective, according to the literature [126], CNN works well with data that have a spatial relationship. In hate speech classification tasks, there is an order relationship between words in a document and CNN learns to recognize patterns across space. In the combination of BERT + CNN, although convolutions and pooling operations lose information about the local order of words, it has already captured by BERT encoders and its position embeddings in different layers. On the other hand, from the language modeling perspective, BERT + CNN uses all the information included in different layers of pre-trained BERT during the fine-tuning phase. This information contains both syntactical and contextual features coming from lower layers to higher layers of BERT. Therefore, this model performs the best among all models.

4.2.4.4 Performance Evaluation with a Limited Amount of Training Data

In common practice the more the fraction of training set is, the higher the performance of algorithms will be. One advantage of leveraging the pre-trained model is to be able to train a model for downstream task within a small training set. Due to the lack of a sufficient amount of labeled data in some classification tasks, mainly hate speech detection here, using the pre-trained BERT model can be effective. We inquire into the performance of hate speech detection models in terms of F1-measure when the amount of labeled data is restricted. Figure 4.3 shows the evaluation results of the baselines and our pre-trained BERT-based model on different portions of training examples, over a certain concentration range $[0.1 - 1.0]$. We train and test each model 10 times and report the results in terms of their mean and standard deviation. For each dataset, we select training and test sets



(a) Davidson-dataset (b) Waseem-dataset

Figure 4.3 – The performances of hate speech detection models trained with a variation of training sets on Davidson and Waseem datasets. The x-axis is the portion of the training and validation sets used for training our BERT-based model and the baselines, the y-axis shows the F1-measure.

according to the description included in Section 4.2.4.3. We do not use the validation set (10% of the dataset) for Davidson et al. [3] baseline model but it is used in Waseem et al. [121] baseline. In Waseem et al. [121] baseline model we are dealing with a multi-task learning approach, therefore in each iteration, the training and validation sets of a specific task which is going to be trained are selected. For our proposed method, we report the performance of the pre-trained BERT model fine-tuned with inserting a CNN layer on top of it; the best performing fine-tuning strategy. To see how the models perform on different portions of training and validation sets, we restrain the amount of training and validation sets in such a way that only a specific portion of them are available for the models during the training.

The experiment results demonstrate that our pre-trained BERT-based model brings a significant improvement to small size data and has comparable performance on different portions of training data in comparison to the baseline models. According to Figure 4.3a, the smallest portion of training data, which is 0.1, used in the training phase of our model is able to yield the F1-measure of almost 87% where it is 72% for Davidson baseline. By increasing the portion of training data, the performance of the Davidson baseline gradually increases up to 83% (where the portion of the training set is 0.5) and then remains considerably stable, whereas the performance of our model does not significantly improve. This finding supports the theory that using a pre-trained BERT-based model causes a decrease in the size of the required training data to achieve a specific performance. From Figure 4.3b, we

can observe that the performance of the multi-task learning approach proposed by Waseem et al. [121] gradually increases and it depends on the portion of training data. However, the performance of our model is mostly stable during the growth of training data, especially by including more than 0.3 of training data.

4.2.4.5 BERT Embeddings Analysis

To see how informative different 12 layers of transformer encoders of the BERT model are, we extract embeddings for each sentence in our datasets, from pre-trained BERT model before and after fine-tuning. Here, we use the uncased $\text{BERT}_{\text{base}}$ model with 12 transformer blocks, 12 attention heads, and a hidden layer size of 768. For this purpose, we use an online service called bert-as-service[7] to map a variable-length sentence into a fixed-length vector representation and extract sentence embeddings from different layers of the BERT model.

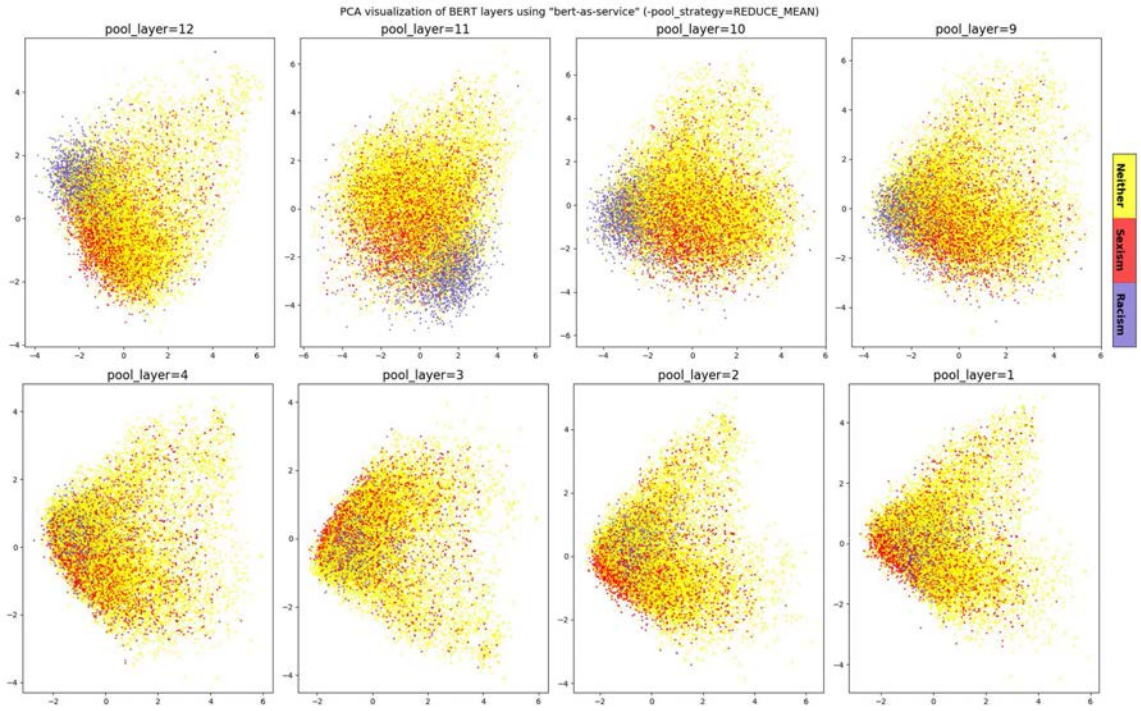
We extract the vector representation of all samples in Davidson and Waseem datasets separately from the original pre-trained BERT model and the one we fine-tuned on our downstream tasks. Each sample is translated into a 768-dimensional vector. As [CLS] special token appeared at the start of each sentence does not have richly contextual information before fine-tuning the model on a specific classification task, we take all the tokens' embeddings in a sentence and apply a REDUCE-MEAN pooling strategy to get a fixed representation of a sentence. Given the sentence representations from the pre-trained BERT model before and after fine-tuning, Principal Component Analysis (PCA) builds a mapping of 768-dimensional vector's representation to a 2D space shown in Figures 4.4 and 4.5 for Waseem-dataset and Davidson dataset, respectively. There are three classes of the data, illustrated in purple, red, and yellow corresponding to Racism, Sexism, and Neither classes in Waseem-dataset and Hate, Offensive, Neither in Davidson-dataset, respectively.

Sentence Embeddings from the first 4 layers (1-4) and the last 4 layers (9-12) of pre-trained BERT model before fine-tuning on Waseem-dataset are represented in Figure 4.4a Regarding the fact that different pre-trained BERT layers capture different information, we can see that sentences' representation from each class in the first 4 layers is highly sparse which means the Euclidean pairwise distance between sentences in each class is large in the high dimensional space. However, the sentence embeddings in the last 4 layers are a bit more clustered in comparison to the first 4 layers according to the class which they belong to; Especially for Racism samples. This observation is on the grounds that, pre-trained BERT model is trained on Wikipedia and Book Corpus data and encodes enough prior knowledge of the general and formal language into the model. However, this knowledge is not specific to a particular domain; here hate speech contents form social media with informal language. Therefore, before fine-tuning the model on our task different layers of

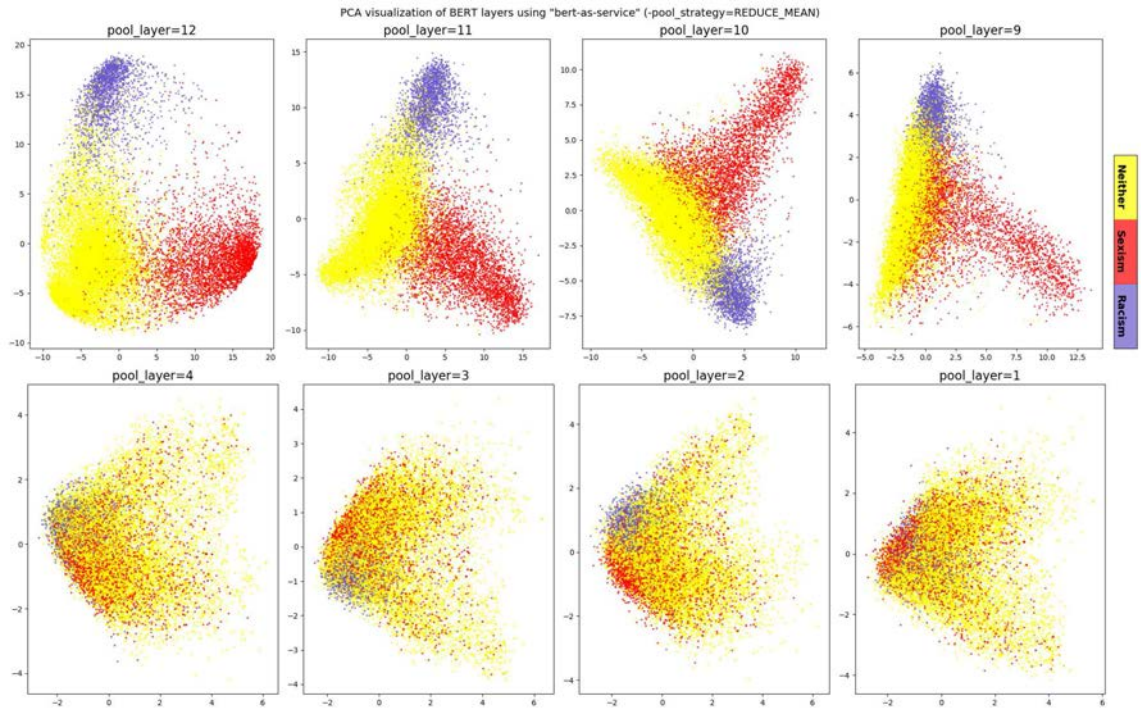
[7] `https://github.com/hanxiao/bert-as-service`

BERT cannot capture the contextual and semantic information of samples in each class and cannot congregate similar sentences in a specific class.

After fine-tuning our model, on Waseem-dataset, with $\text{BERT}_{\text{base}}$ + CNN strategy, which performs as the best fine-tuning strategy on both datasets, we can observe in Figure 4.4b that the model captures contextual information in which Racism, Sexism, and Neither content exist and clusters samples strongly tight in the last 4 layers. It causes the high-performance evaluation result using this fine-tuning strategy in our study. The same result is yielded by Davidson-dataset's embeddings visualization depicted in Figure 4.5.

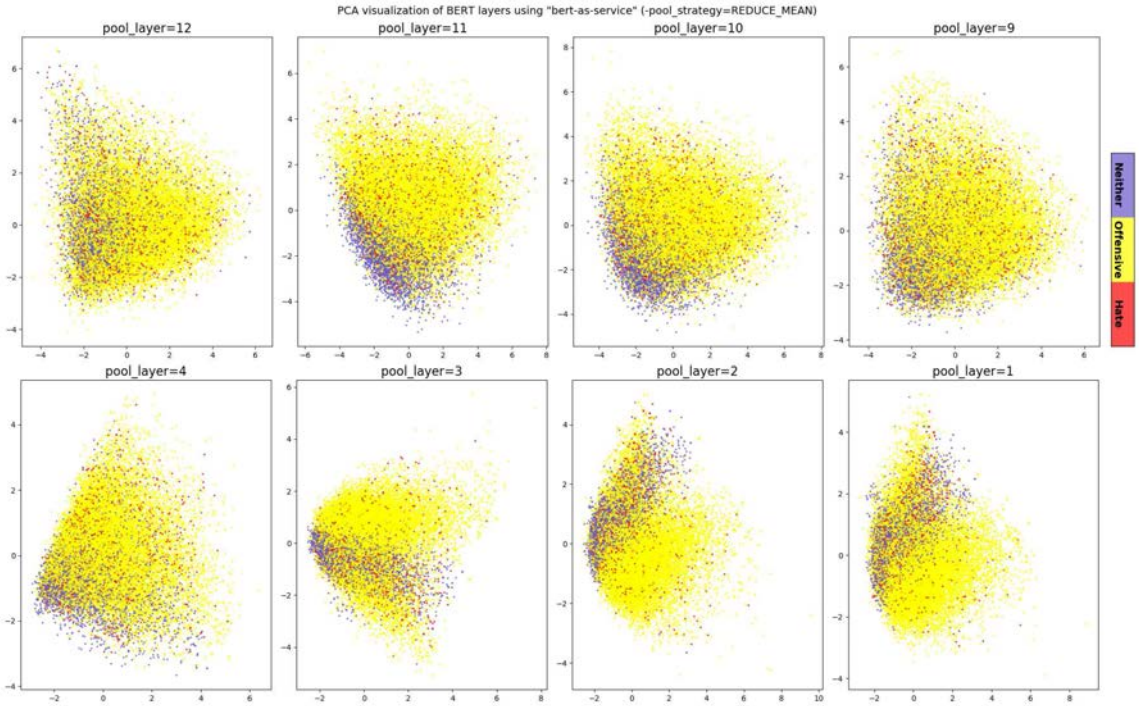


(a) Before fine-tuning

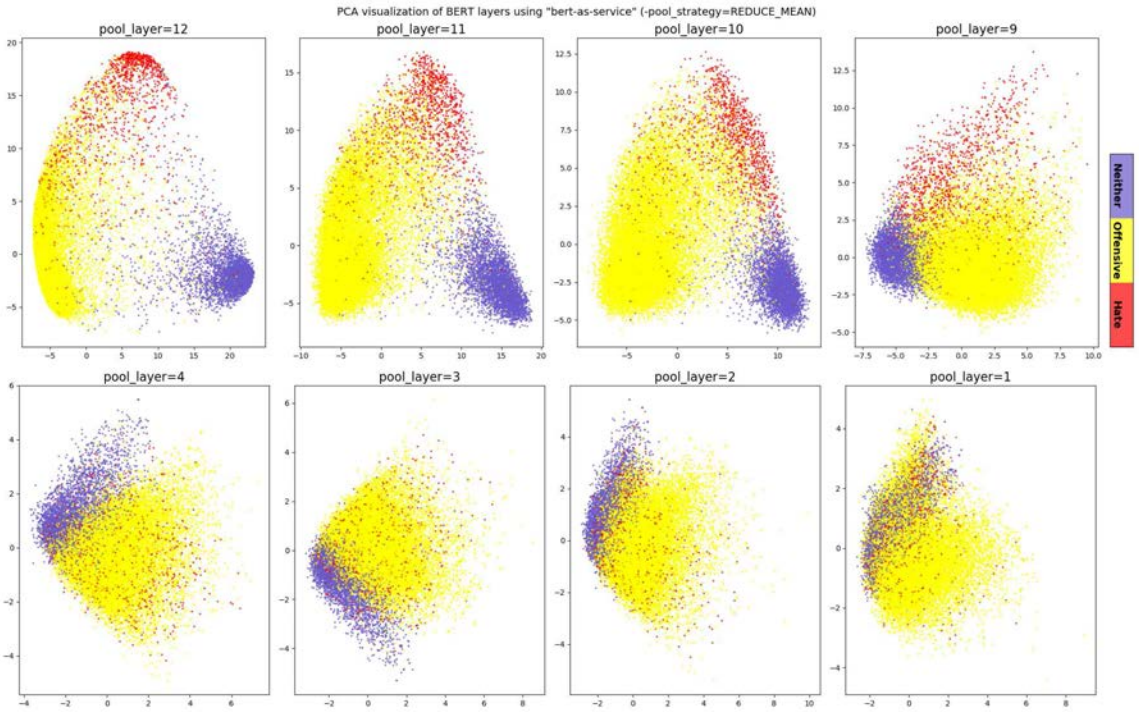


(b) After fine-tuning

Figure 4.4 – Waseem-samples' embeddings analysis before and after fine-tuning. To investigate the impact of information included in different layers of BERT, sentence embeddings are extracted from all the layers of the pre-trained BERT model fine-tuning, using the bert-as-service tool. Embedding vectors of size 768 are visualized to a two-dimensional visualization of the space of all Waseem-dataset samples using PCA method. For sake of clarity, we just include visualization of the first 4 layers (1-4), which are close to the training output, and the last 4 layers (9-12), which are close to the word embedding, of the pre-trained BERT model before and after fine-tuning.



(a) Before fine-tuning

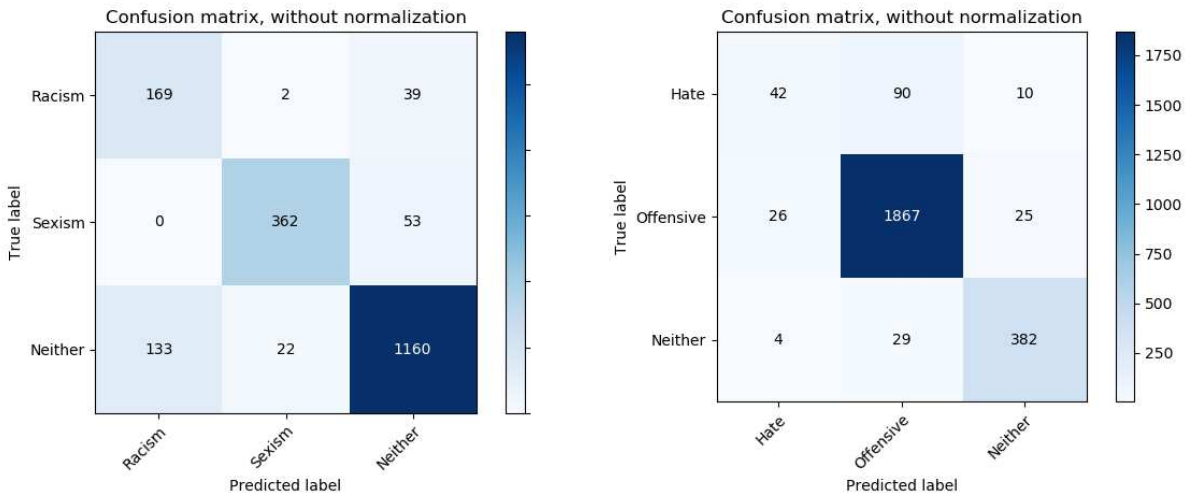


(b) After fine-tuning

Figure 4.5 – Davidson-samples' embeddings alanysis before and after fine-tuning. To investigate the impact of information included in different layers of BERT, sentence embeddings are extracted from all the layers of the pre-trained BERT model before (a) and after (b) fine-tuning, using the bert-as-service tool. Embedding vectors of size 768 are visualized to a two-dimensional visualization of the space of all Davidson-dataset samples using PCA method. For sake of clarity, we just include visualization of the first 4 layers (1-4), which are close to the training output, and the last 4 layers (9-12), which are close to the word embedding, of the pre-trained BERT model before and after fine-tuning.

4.2.4.6 Error analysis

Although we have very interesting results in terms of F1-measure, it is needed to examine how the model predicts false positives and false negatives. To understand better this phenomenon, in this section we perform an analysis on error of the model. We investigate the test datasets and their confusion matrices resulted from the $\text{BERT}_{\text{base}}$ + CNN model as the best fine-tuning approach that are depicted in Figure 4.6 . According to Figure 4.6a for Waseem-dataset, it is obvious that the model can separate sexism from racism content properly. Only two samples belonging to racism class are misclassified as sexism and none of the sexism samples are misclassified as racism. A large majority of the errors comes from misclassifying hateful categories (racism and sexism) as hatless (neither) and vice versa. 0.9% and 18.5% of all racism samples are misclassified as sexism and neither respectively whereas it is 0% and 12.7% for sexism samples. Almost 12% of neither samples are misclassified as racism or sexism. As Figure 4.6b makes clear for Davidson-dataset, the majority of errors are related to hate class where the model misclassified hate content as offensive in 63% of the cases. However, 2.6% and 7.9% of offensive and neither samples are misclassified respectively.



(a) Heat map of the Waseem's test set predictions (b) Heat map of the Davidson's test set predictions

Figure 4.6 – Evaluation results by confusion matrix.

To understand better the mislabeled items by our model, we did a manual inspection on a subset of the data and record some of them in Tables 4.5 and 4.6. Considering the words such as "daughters", "women", and "burka" in tweets with IDs 1 and 2 in Table 4.5, it can be understood that our BERT based classifier is confused with the contextual semantic between these words in the samples and misclassified them as sexism because they

Table 4.5 – Misclassified samples from Waseem-dataset.

ID	Tweet	Annotated	Predicted
1	@user Good tweet. But they actually start selling their daughters at 9.	Racism	Sexism
2	RT @user: Are we going to continue seeing the oppression of women or are we going to make a stand? #BanTheBurka http://t.co/hZDx8mlvTv.	Racism	Sexism
3	RT @user: @user my comment was sexist, but I'm not personally, always a sexist.	Sexism	Neither
4	RT @user: @user Ah, you're a #feminist? Seeing #sexism everywhere then, do check my tweets before you call me #sexist	Sexism	Neither
5	@user By hating the ideology that enables it, that is what I'm doing.	Racism	Neither

Table 4.6 – Misclassified samples from Davidson-dataset.

ID	Tweet	Annotated	Predicted
6	@user: If you claim Macklemore is your favorite rapper I'm also assuming you watch the WNBA on your free time fagg*t	Hate	Offensive
7	@user: Some black guy at my school asked if there were colored printers in the library. "It's 2014 man you can use any printer you want" I said.	Hate	Neither
8	RT @user: @user typical c*on activity.	Hate	Neither
9	@user: @user @user White people need those weapons to defend themselves from the subhuman trash your sort unleashes on us.	Neither	Hate
10	RT @user: Finally! Warner Bros. making superhero films starring a woman, person of color and actor who identifies as ""que*r"";	Neither	Offensive

are mainly associated to femininity. In some cases containing implicit abuse (like subtle insults) such as tweets with IDs 5 and 7, our model can not capture the hateful/offensive content and therefore misclassifies them. It should be noted that even for a human it is difficult to discriminate against this kind of implicit abuses.

By examining more samples and with respect to recent studies [87, 88, 119], it is clear that many errors are due to biases from data collection [88] and rules of annotation [119] and not the classifier itself. Since Waseem et al. [2] created a small ad-hoc set of keywords and Davidson et al. [3] used a large crowdsourced dictionary of keywords (Hatebase lexicon) to sample tweets for training, they included some biases in the collected data. Especially for Davidson-dataset, some tweets with specific language (written within the African American Vernacular English) and geographic restriction (United States of America) are oversampled such as tweets containing disparage words "n*gga", "fagg*t", "c*on", or "que*r" result in high rates of misclassification. However, these misclassifications do not confirm the low performance of our classifier because annotators tended to annotate many samples containing disrespectful words as hate or offensive without any presumption about the social context of tweeters such as the speaker's identity or dialect, whereas they were just offensive or even neither tweets. Tweets IDs 6, 8, and 10 are some samples containing offensive words and slurs which are not hate or offensive in all cases and writers of them used this type of language in their daily communications. Given these pieces of evidence, by considering the

content of tweets, we can see in tweets IDs 3, 4, and 9 that our BERT-based classifier can discriminate tweets in which neither and implicit hatred content exist. One explanation of this observation may be the pre-trained general knowledge that exists in our model. Since the pre-trained BERT model is trained on general corpora, it has learned general knowledge from normal textual data without any purposely hateful or offensive language. Therefore, despite the bias in the data, our model can differentiate hate and offensive samples accurately by leveraging knowledge-aware language understanding that it has and it can be the main reason for high misclassifications of hate samples as offensive (in reality they are more similar to offensive rather than hate by considering social context, geolocation, and dialect of tweeters).

4.2.5 Conclusion

Conflating hatred content with offensive or harmless language causes online automatic hate speech detection tools to flag user-generated content incorrectly. Not addressing this problem may bring about severe negative consequences for both platforms and users such as decreasement of platforms' reputation or users abandonment. In this study, we proposed a transfer learning approach advantaging the pre-trained language model BERT to enhance the performance of a hate speech detection system and to generalize it to new datasets. To that end, we introduced new fine-tuning strategies to examine the effect of different layers of BERT in hate speech detection task. The evaluation results indicated that our model outperforms previous works by profiting the syntactical and contextual information embedded in different transformer encoder layers of the BERT model using a CNN-based fine-tuning strategy. Furthermore, examining the results showed the ability of our model to detect some biases in the process of collecting or annotating datasets. It can be a valuable clue in using the pre-trained BERT model to alleviate bias in hate speech datasets in future studies, by investigating a mixture of contextual information embedded in the BERT's layers and a set of features associated to the different type of biases in data. Next section will address this issue by providing a bias mitigation mechanism based on the proposed BERT-based model for hate speech detection task.

4.3 Racial Bias Mitigation in Social Media based on BERT Model

4.3.1 Introduction

There is a considerable disagreement about what exactly hate speech is [69, 127], and how different terms can be inferred as hateful or offensive in certain circumstances. For example,

some terms such as "n*gga" and "c*on" were used to disparage African American communities, however, they were not known as offensive when used by peoples belonging to these communities [119].

From the bias perspective, despite previous efforts into generating well-performed methods to detect hate speech and offensive language, the potential biases due to the collection and annotation process of data or training classifiers have raised a few concerns. Some studies have ascertained the existence of bias regarding some identity terms (e.g., gay, bisexual, lesbian, Muslim, etc.) in the benchmark datasets and tried to mitigate the bias using an unsupervised approach based on balancing the training set [74] or debiasing word embeddings and data augmentation [75]. Moreover, some racial and dialectic bias exist in several widely used corpora annotated for hate speech and offensive language [87, 88, 119]. Therefore, it is crucial to consider data-driven and algorithm-driven biases included in the hate speech detection system. Additionally, these kinds of race and gender discriminations caused by exciting biases in dataset or classifiers lead to unfairness against the same groups that the classifiers are trained to protect.

In Section 4.2, we proposed a transfer learning approach for identification of hate speech in online social media by employing a combination of the unsupervised pre-trained model BERT [6] and new supervised fine-tuning strategies. In this section, we investigate the effect of unintended bias in our pre-trained BERT-based model and propose a new generalization mechanism in training data by reweighting samples and then changing the fine-tuning strategies in terms of the loss function to mitigate the racial bias propagated through the model.

The primary contributions of this study are:

- A regularization mechanism is used to mitigate data-driven and algorithm-driven bias by reweighting the training data and improving their generalization apart from their classes. We use two publicly available datasets for hate speech and offensive language detection.

- New fine-tuning strategy, in terms of the loss function, is employed to fine-tune the pre-trained BERT model by new re-weighted training data.

- A cross-domain validation approach is performed to show the efficiency of the proposed bias mitigation mechanism.

4.3.2 Related Work

In this section, we present related works have been done on data-driven and algorithm-driven bias analysis for hate speech detection task.

Recently the great efforts have taken to examine the issue of data bias in hate speech and offensive language detection tasks. Dixon et al. [74] confirmed the existence of unintended bias between texts containing general identity terms (e.g. lesbian, gay, Islam, feminist, etc.) and a specific toxicity category; attributed to the disproportionate representation of texts containing certain identity terms through different categories in training data from Wikipedia Talk pages dataset. Therefore, they tried to quantify and mitigate this form of unintended bias by expanding training and test datasets under some generalization strategies for identity terms. Following some debiasing methods (Debiased Word Embeddings, Gender Swap and Bias fine-tuning), Park et al. [75] tried to measure and debias gender bias in abusive language detection system. Afterward, Wiegand et al. [88] conveyed that unintended biases in datasets are not just restricted to the identity terms and gender and they are by cause of focused data sampling approaches. Consequently, the high classification scores on these datasets, mainly containing implicit abuse, are due to the modeling of the bias in those datasets. Datasets containing biased words resulted from biased sampling procedure cause a huge amount of false positives when testing on other datasets. They showed that some query words used for sampling data from Twitter that are not correlated with abusive tweets but are included in tweets with sexist or racist remarks are biased as well. For example, query words such as commentator, sport, and gamergate used by Waseem et al. [2] to sample data from Twitter, are not correlated with Sexism class but are one of the most frequent words in this category. Furthermore, Badjatiya et al. [128] proposed a two-step bias detection and mitigation approach. At first, various heuristics were described to quantify the bias and a set of words in which the classifier's stereotypes were identified. Then, they tried to mitigate the bias by leveraging knowledge-based generalization strategies in training data. The results show that their approach can alleviate the bias without reducing the model performance significantly.

Recently, Davidson et al. [87] and Sap et al. [119] investigated the racial bias against African American English (AAE) dialects versus Standard American English (SAE) in the benchmark datasets with toxic content, especially from the Twitter platform. They declared that the classifiers trained on these datasets tend to predict contents written in AAE as abusive with strong probability. Furthermore, Sap et al. [119] introduced a way of mitigating annotator bias through dialect but they did not mitigate the bias of the trained model.

We propose a pre-trained BERT-based model to address unintended bias in data or trained model and try to mitigate the racial bias in our pre-trained BERT-based classifier. Our bias mitigation approach is close to what Davidson et al. [87] did at which they just addressed the racial bias in the benchmark hate speech datasets. However, in this study, we use a bias mitigation mechanism to alleviate racial bias included in datasets and trained classifiers by leveraging a regularization mechanism in training set proposed by Schuster et

al. [129] for alleviating the bias in fact verification tasks.

4.3.3 Bias Mitigation Module

As depicted in Figure 4.1, our proposed framework consists of two main modules. This section concentrates on the bias mitigation module at which we address the problem of data-driven and algorithm-driven biases in hate speech detection. We explore existence bias in the datasets and then try to mitigate the bias in the proposed pre-trained BERT-based model by applying a generalization mechanism.

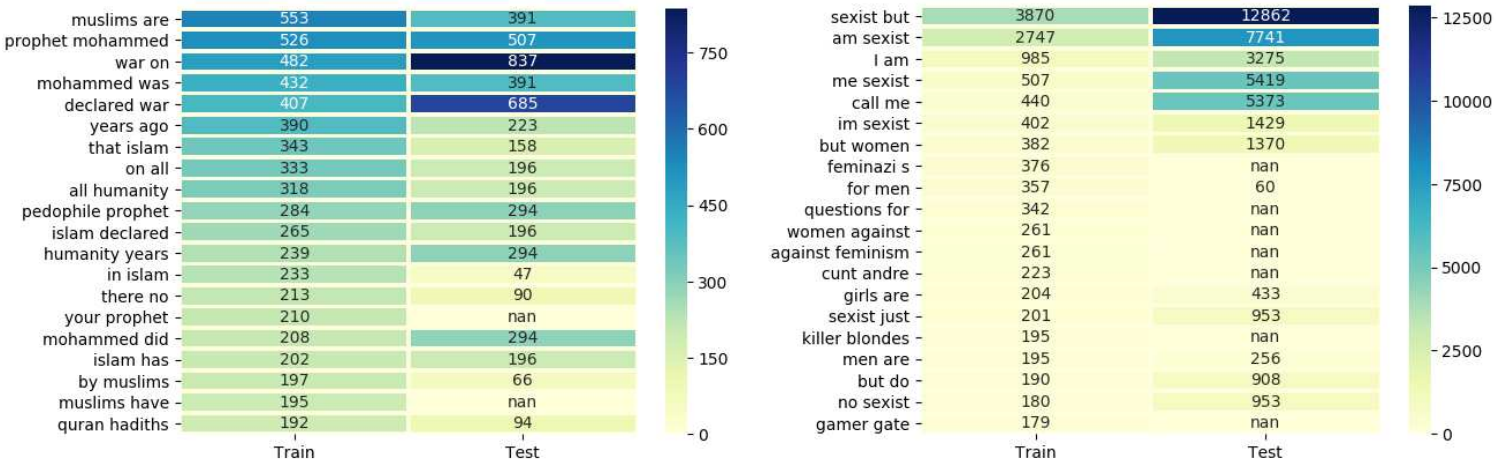
4.3.3.1 Towards Unbiased Training Data

To the best of our knowledge, it is the first time that we are addressing bias mitigation through trained classifier rather than data sampling and annotation process. Here, we try to improve the generalization in the existence of the racial and dialect bias by proposing a new generalization mechanism in the training data. To mitigate the bias propagated through the models on which the benchmark datasets are trained, we leverage a re-weighting mechanism, by inspiring from the recent work of Schuster et al. [129]. First, we assess the explicit bias in the datasets and investigate phrases in training set causing it. Then, we reweight the samples in training and validation sets to make smooth the correlation between the phrases in training samples and the classes to which they belong. After optimizing the bias in the training set, we acquire re-weighted scores for each sample and feed our pre-trained BERT-based model with new training and validation sets (as depicted in Figure 4.1, where tweets and corresponding weights are as an input of the Bias Mitigation module). During the fine-tuning, the loss function of the classifier will be updated with re-weighted scores to alleviate the existing bias in training samples.

The high classification scores in hate speech detection and offensive language systems are likely due to modeling the bias from training datasets. Therefore, we assess the explicit bias in Davidson and Waseem datasets and investigate phrases in training sets causing it. To do so, the n-gram distribution in training and test sets is inspected and the high frequently n-grams, that are extremely correlated with a particular class, are extracted. We use the Local Mutual Information (LMI) [130] to extract high frequently n-grams in each class. For any given n-gram w and class c, LMI between w and c is defined as follows:

$$LMI(w,c) = p(w,c).log(\frac{p(c|w)}{p(c)}) \tag{4.1}$$

where $p(c|w)$ and $p(c)$ are calculated by $\frac{count(w,c)}{count(w)}$ and $\frac{count(c)}{|D|}$, respectively. Furthermore, $p(c)$ and $p(w|c)$ are calculated by $\frac{count(c)}{|D|}$ and $\frac{count(w,c)}{|D|}$, respectively. $|D| =$ is the number of occurrences of all n-grams in the training set.



(a) Racism class

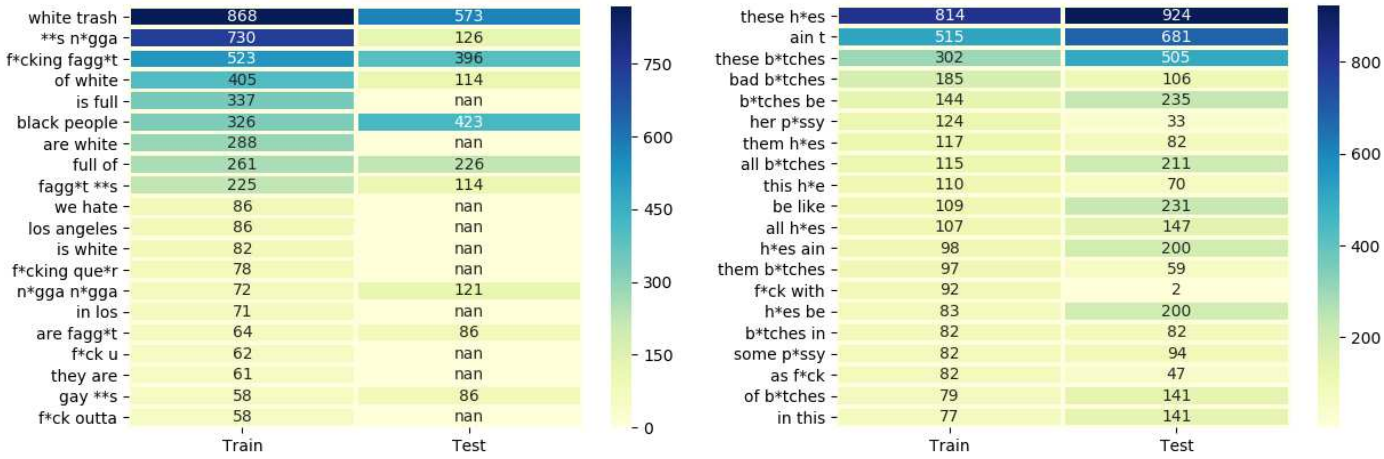
(b) Sexism class

Figure 4.7 – The top 20 LMI-ranked n-grams ($n = 2$) that are highly correlated with the negative classes of Waseem-dataset (Racism and Sexism) in the training and test sets. nan value denotes computationally infeasible, as the occurrence is zero in the test set.

Figures 4.7 and 4.8 exhibit the top 20 LMI-ranked n-grams ($n = 2$) that are highly correlated with the Racism and Sexism classes of Waseem-dataset and Hate and Offensive classes of Davidson-dataset in the training and test sets, respectively. Using training and test data, a heat map with legend color bar, column and row side annotations is generated in Figures 4.7a and 4.7b for Racism and Sexism and Figures 4.8a and 4.8b for Hate and Offensive classes. The legend color bar indicates the correlation between LMI values and colors, and the colors are balanced to ensure the light yellow color represents zero value. LMI values indicate with $LMI.10^{-6}$.

Illustrating the most frequently 2-grams in Racism class in Figure 4.7a shows that tweets in this class are containing some domain-specific expressions such as 'islam' and 'muslims' at which they are likely to be associated with Racism class (as hateful class). On the other hand, in Figure 4.7b some general keywords such as 'women', 'feminism' and 'sexist' are highly associated with Sexism class. These kinds of correlations are true for both training and test sets' samples except some phrases in which there is not any occurrence in the test set and is indicated as nan value. Therefore, it is perceived that there are some idiosyncrasies in the dataset construction for each class and they are described as stereotype bias in the rest of this chapter.

The same stereotype bias exists in Hate and Offensive classes of Davidson-dataset, depicted in Figure 4.8, where samples containing specific terms such as "n*gga", "fagg*t", "que*r", etc., are highly correlated with Hate class. On the other hand, the samples con-



(a) Hate class

(b) Offensive class

Figure 4.8 – The top 20 LMI-ranked n-grams ($n = 2$) that are highly correlated with the negative classes of Davidson dataset (Hate and Offensive) in the training and test sets. nan value denotes computationally infeasible, as the occurrence is zero in the test set.

taining terms such as "h*es" and "b*tch" are associated with Offensive class. This kind of stereotype bias can be transferred to the classifier during the training process and creates a tendency for predicting new samples containing this stereotype as a negative class.

4.3.3.2 Re-weighting Mechanism

This section presents the mechanism to alleviate the bias in our hate speech detection model. We describe how samples belonging to each class are assigned a positive weight according to their correlation with the different classes. After that, samples with new weights are fed to our pre-trained BERT-based model. To mitigate the bias initiated by n-grams high correlated to each class in our proposed model, we use an algorithm introduced by Schuster et al. [129] for debiasing a fact verification model, to reweight the samples. We believe that it is the first attempt to reduce the systematic bias existing in hate speech datasets with such kind of re-weighting mechanism.

Bias made by high frequently 2-grams per class in training and validation sets can be constrained by defining a positive weight α^i for each sample x^i, tweet in training and validation sets, in such a way that the importance of tweets with different labels containing these phrases are increased. Considering each sample as x^i, its label as y^i and each 2-gram in training set as w_j, we define a bias toward each class c using Eq 4.2 [129] .

$$b_j{}^c = \frac{\sum_{i=1}^{n} I_{[w_j^{(i)}]}(1+\alpha^{(i)}) I_{[y^{(i)}=c]}}{\sum_{i=1}^{n} I_{[w_j^{(i)}]}(1+\alpha^{(i)})} \tag{4.2}$$

Where $I_{[w_j^{(i)}]}$ and $I_{[y^{(i)}=c]}$ are the indicators for w_j to be in tweet x^i and lable y^i to be in class c.

To find balancing weights α that result in the minimum bias, we have to solve an optimization problem as follows [129]:

$$min(\sum_{j=1}^{|V|} max_c(b_j^c) + \lambda||\overrightarrow{\alpha}||_2) \tag{4.3}$$

It should be noted that we acquire α values in the pre-processing step and before feeding training and validation sets to our BERT-based model. To integrate the weights associated with each sample into our model, the loss function of our pre-trained BERT-based classification model has to be changed. In Section 4.2 we used Cross-entropy loss functionas a loss function when optimizing our classification model on top of the pre-trained BERT model. However, in this section, we change the loss function in such a way that it includes weights as well.

Let $y = y_1, ..., y_n$ be a vector representing the distribution over the classes $1, ..., n$, and let $\widehat{y} = \widehat{y_1}, ..., \widehat{y_n}$ be the classifier output .The categorical cross entropy loss measures the dissimilarity between the true label distribution y and the predicted label distribution $\widehat{y}$, and is defined as cross entropy as follows:

$$\text{Loss}_{cross-entropy}(\widehat{y}, y) = -\sum_{i=1}^{n} y_i \log(\widehat{y}) \tag{4.4}$$

While for the re-weighted approach, the training objective is reweighted from the Eq 4.4 to:

$$\text{Weighted-Loss}_{cross-entropy}(\widehat{y}, y) = -\sum_{i=1}^{n} (1+\alpha^{(i)}) y_i \log(\widehat{y}) \tag{4.5}$$

4.3.3.3 Scrutinizing Bias Mitigation Mechanism

To further analyze the impact of the proposed regularization mechanism through training and validation sets and reweighting the samples for bias mitigation, we investigate how the models trained on samples with and without weights predict on new datasets (cross-domain data). We use a dataset collected from twitter by Blodgett et al. [131] including a demographically associated dialectal language named African American English (AAE),

known as Black English, which is a dialect of American English spoken by millions of black people across the United States. They exploited a set of geo-located tweets by leveraging a distantly supervised mapping between authors and the demographics of the place in which they live. They filtered out 16 billion collected tweets in such a way that tweets geo-located with coordinates that matched a U.S. Census blockgroup remained; which contains 59.2 million publicly available tweets. Consequently, four different demographic categories of non-Hispanic whites, non-Hispanic blacks, Hispanics, and Asians are created using the information about population ethnicity and race from the U.S. Census. They proposed a probabilistic mixed-membership language model to learn demographically aligned language models for each of the four demographic categories utilizing words associated with particular demographics. At the end, they calculated a posterior proportion of language from each category in each tweet. Following Davidson et al. [87] recent work, to analysis racial bias propagated with the pre-trained BERT-based model with and without the re-weighting mechanism, we define two categories of tweets as follows:

AAE-aligned: filtering the tweets with the average posterior proportion greater than 0.80 for the non-Hispanic black category and less than 0.10 for Hispanic + Asian together to address the African Americal English language (AAE).

White-aligned: filtering the tweets with the average posterior proportion greater than 0.80 for the non-Hispanic white category and less than 0.10 for Hispanic + Asian together to address the Standard American English (SAE).

After filtering out the tweets not satisfying the above conditions, we result in a set of 14.5m and 1.1m tweets written in non-Hispanic white (White-aligned) and non-Hispanic black (AAE-aligned) languages, respectively. These two new categories show the racial alignment of the language that their authors used. In the following, we explain how we use these datasets to evaluate our pre-trained BERT-based classifier with and without re-weighting mechanism to alleviate racial bias.

Research Question: Our research question here is that, whether or not our BERT-based classifiers trained on Waseem and Davidson datasets with and without the re-weighting mechanism, have any preference in assigning tweets from AAE-aligned and White-aligned categories to a negative class (Racism, Sexism, Hate or Offensive). If it is yes, how our proposed bias alleviation mechanism reduces this tendency.

Considering each tweet t in AAE-aligned dataset as t_{black} and in White-aligned dataset as t_{white}, we define two hypotheses $H1$ and $H2$ for each class c_i where $c_i = 1$ denotes membership of t in class i and $c_i = 0$ in the opposite. Therefore, $H1$ is equivalent to $P(c_i = 1|black) = p(c_i = 1|white)$ in which the probability of t to be a member of a negative class i is independent of the racial group at which it belongs to. $H2$ is equivalent to $P(c_i = 1|black) > p(c_i = 1|white)$ or $P(c_i = 1|black) < p(c_i = 1|white)$ in which the

probability of t to be a member of a negative class i is dependent on the racial group at which it belongs to.

To assess our hypotheses, we conduct an experiment in which we sample 10000 tweets from each AAE-aligned and White-aligned groups and feed them as a test set to our pre-trained BERT-based classifiers trained on Davidson and Waseem datasets, separately, with and without the re-weighting mechanism to predict the membership probability of each tweet in each class. For each classifier, trained on Waseem and Davidson datasets, we create a vector containing the membership probability p_i of each class i in size of the number of samples in each group (10000). Indeed, we obtain one vector per each class i for tweets in two AAE-aligned and White-aligned groups and calculate the portion of tweets assigned to each class i for each group as follows:

$\widehat{p_{i_{black}}} = \frac{1}{n}\sum_{j=1}^{n} p_{ij}$ where j denotes the samples from AAE-aligned and $\widehat{p_{i_{white}}} = \frac{1}{n}\sum_{j=1}^{n} p_{ij}$ where j denotes the samples from White-aligned and $n = 10000$. To examine the racial bias tendency of each classifier on each class i, we also calculate $\frac{\widehat{p_{i_{black}}}}{\widehat{p_{i_{white}}}}$ as an indicator. If this portion is greater than 1 then it indicates that our classifier has a higher propensity to assign AAE-aligned tweets to a specific class i rather than White-aligned tweets.

To see how significant the differences between $\widehat{p_{i_{black}}}$ and $\widehat{p_{i_{white}}}$ are, we apply an independent samples t-test between two groups which results in t and p values, where t indicates the difference between two groups and the difference within the groups and p indicates the probability that the results from the tweets samples occurred by chance. A low value of p shows that our membership probabilities assigned with the classifiers did not occur by chance (Here, the p values for all the classes are less than 0.001 which indicated as *** in Table 4.7).

All the results are shown in Table 4.7, where we computed the aforementioned statistics with and without including the bias alleviation mechanism in our pre-trained BERT-based models trained on different datasets. Statistics signed with $*$ indicate the values after debiasing the training sets. For fine-tuning the pre-trained BERT model, we have tried all fine-tuning strategies, but report the results from the best performing strategy in bias mitigation task which is $\text{BERT}_{\text{base}}$ fine-tuning strategy. The first row shows the performance of classifier trained on Waseem dataset on two-race groups before and after reweighting. The second row indicates the same results for Davidson dataset. In all cases, the tweets belonging to AAE-aligned group are more frequently predicted as a member of negative classes than White-aligned which indicates existing of systematic bias in two datasets.

Surprisingly, there is a significant difference across AAE-aligned and White-aligned groups in Racism class' s estimated rates. Our classifier on Waseem-dataset classifies tweets in AAE-aligned group as Racism 10.5 times more probably than White-aligned without

Table 4.7 – Racial bias analysis before and after reweighting the training data. To quantify the impact of the re-weighting mechanism in alleviating the racial bias propagated through trained classifiers, we examine our BERT-based classifiers trained on Davidson and Waseem datasets with and without re-weighting mechanism on AAE-aligned and SAE-aligned samples.

		Before reweighting					**After reweighting**				
Dataset	**Class**	$\widehat{p_{i_{black}}}$	$\widehat{p_{i_{white}}}$	t	p	$\frac{\widehat{p_{i_{black}}}}{\widehat{p_{i_{white}}}}$	$\widehat{p_{i_{black}}}^*$	$\widehat{p_{i_{black}}}^*$	t^*	p^*	$\frac{\widehat{p_{i_{black}}}}{\widehat{p_{i_{white}}}}^*$
Waseem-dataset	Racism	0.049	0.005	10.450	***	10.593	0.028	0.007	6.852	***	3.726
	Sexism	0.162	0.055	31.715	***	2.923	0.235	0.092	15.949	***	2.561
Davidson-dataset	Hate	0.058	0.026	84.986	***	2.230	0.043	0.031	1.815	***	1.384
	Offensive	0.360	0.143	17.913	***	2.515	0.193	0.106	120.607	***	1.823

We just consider negative classes and "Neither" class in both datasets is excluded.

reweighting, which indicates potential bias carried with our trained model and not dataset itself. However, after applying bias alleviation mechanism by reweighting the samples and decreasing the correlation between high frequently 2-grams and each negative class, we can observe that our model decreases $\frac{\widehat{p_{i_{black}}}}{\widehat{p_{i_{white}}}}^*$ by 6.8 times for Racism class. This kind of racial bias reduction is true for Sexism class as well.

For Davidson-dataset, we observe that tweets in AAE-aligned are classified as Hate and Offensive more frequently than White-aligned. The classifier trained on Davidson-dataset before applying the re-weighting mechanism gives Hate label to AAE-aligned tweets with 5.8% and to White-aligned tweets with 2.6%, as opposed to 4.3% and 3.1% in re-weighted classifier. Consequently, $\frac{\widehat{p_{i_{black}}}}{\widehat{p_{i_{white}}}}^*$ gets down by 0.85 times in comparison with $\frac{\widehat{p_{i_{black}}}}{\widehat{p_{i_{white}}}}$ in Hate class. For Offensive class, the bias mitigation rate is 0.70 where the probability of assigning AAE-aligned samples to Offensive class reduces from 36% to 19%. Comparing results for Hate and Offensive classes shows that the classifiers trained on Davidson-dataset classify AAE-aligned tweets more frequently as Offensive rather than Hate; which is the result of the unbalanced dataset we used to train the classifiers.

From Table 4.7 it is inferred that substantial racial bias perseveres even after using our bias alleviation mechanism, however, it is generally reduced for cases in which classifiers are trained with re-weighted samples. It means that still, our re-weighted classifiers favor assigning tweets from AAE-aligned more probably to negative classes rather than White-aligned after bias mitigation. Given our cross-domain approach for evaluating the bias mitigation mechanism, we hypothesize that differences between Davidson and Waseem datasets' keywords and language and AAE-aligned and White-aligned languages, which are not included in our bias mitigation mechanism, lead classifiers to classify tweets written by African-Americans (AAE-aligned group) as negative classes excessively.

Table 4.8 – Performance evaluation after applying the re-weighting mechanism. To quantify the impact of the re-weighting mechanism in the performance of our pre-trained BERT-based model (with $\text{BERT}_{\text{base}}$ strategy for fine-tuning), we examine the classifier trained on Waseem and Davidson datasets with and without re-weighting mechanism on the training set in terms of macro precision, recall, and F1-measure.

	Before reweighting			**After reweighting**		
Dataset	Precision	Recall	F1-measure	Precision	Recall	F1-measure
Wassem-dataset	81	81	81	76	79	78
Davidson-dataset	91	90	91	85	88	86

We investigate the performance of the pre-trained BERT-based model explained in Section 4.2, with $\text{BERT}_{\text{base}}$ strategy for fine-tuning, after applying the proposed re-weighting mechanism on the in-domain dataset as well; where test data come from Waseem-dataset and Davidson-dataset. Performance evaluation of the classifier before and after reweighting is showed in Table 4.8 in terms of macro precision, recall, and F1-measure.

According to Table 4.8, reweighting the training data has a negative effect on the performance of our classifier in detecting Racism, Sexism, Hate, and Offensive classes. In Waseem-dataset, F1-measure drops 3.7% after reweighting highly correlated 2-grams to the Racism and Sexism classes whereas this reduction is more for Davidson-dataset. After reweighting highly correlated 2-grams to the Hate and Offensive classes in Davidson-dataset, F1-measure drops 5.5%. The main intuition behind this phenomenon is that both training and test sets have the same phrase distribution per class as shown in Figures 4.7 and 4.8. Due to the high correlation between specific 2-grams and a class label, reweighting the training samples results in reducing this correlation and increasing misclassification cases for the test set. Results indicate that this kind of correlation between specific words and labels in Davidson-dataset is higher than Waseem-dataset because the performance reduction is more by applying the re-weighting mechanism.

4.3.4 Discussion and Challenges

Although our pre-trained BERT-based model has achieved promising results in terms of F1-measure on Waseem and Davidson test sets, presented in Tables 4.3 and 4.4, the existing biases in data cannot be captured and measured by a test set at which there is the same biased distribution as training and validation sets. Therefore, we use a cross-domain approach to evaluate our de-biased model. Using the cross-domain approach and demonstrating the results reveals that our classifiers trained on these datasets have systematic and substantial biases where tweets written in AAE are particularly predicted as negative

Table 4.9 – Top 20 unigrams and 2-grams highly correlated with AAE and SAE languages.

	unigrams	**2-grams**
AAE-aligned	(lol, 726); (sh*t, 653); (u, 574); (get, 528); (like, 504); (got, 483); (n*gga, 450); (**s, 428); (im, 366); (f*ck, 314); (go, 312); (know, 291); (b*tch, 290); (n*ggas, 285); (bout, 272); (need, 264); (good, 254); (back, 232); (love, 223);(w*t, 218)	(good_morning, 50); (feel_like, 38); (sh*t_sh*t, 32); (go_sleep, 31); (f*ck_w*t, 30); (talking_bout, 27); (talkin_bout, 26); (look_like, 25); (wanna_go, 23); (last_night, 22); (yo_**s, 22); (u_got, 21); (gotta_get, 19); (worried_bout, 18); (go_back, 17); (**s_n*gga, 17); (real_n*gga, 16); (give_f*ck, 15); (lil_n*gga, 14); (aint_sh*t, 14); (sh*t_like, 13)
SAE-aligned	(like, 574); (get, 475); (go, 407); (love, 372); (good, 361); (one, 339); (day, 311); (time, 282); (know, 271); (night, 260); (lol, 248); (today, 246); (really, 236); (back, 231); (right, 231); (people, 228); (see, 226); (got, 212); (life, 184); (come, 181)	(last_night, 59); (feel_like, 55); (let_us, 34); (wish_could, 26); (go_home, 25); (go_back, 24); (best_friend, 24); (wanna_go, 24); (need_get, 24); (wait_see, 22); (thank_god, 20); (looks_like, 20); (good_day, 20); (first_time, 20); (good_night, 19); (fall_asleep, 18); (good_luck, 17); (come_back, 15); (great_day, 15);(high_school, 15);(holy_sh*t, 13)

classes (racism, sexism, hate, or offensive contents) compared with SAE, as presented in Table 4.7. To get more insight into the differences between dialects used in tweets written in AAE and SAE, we extracted top 20 unigrams and 2-grams highly correlated with AAE and SAE languages and the number of occurrences, included in Table 4.9. We found that there are particular words and phrases, which are more frequently used by AAE rather than SAE, and they are more related to negative classes in training datasets. For example, some particular phrases such as "n*gga", "b*tch", "sh*t", "f*ck_w*t", "**s_n*gga", etc., are common in AAE dialects and are highly correlated with negative classes (Racism, Sexism, Hate and Offensive) in hate and offensive datasets.

We inspected the samples in both AAE and SAE groups that are predicted as racism by applying trained classifiers with and without re-weighting mechanism. The classifier trained on Waseem-dataset without reweighting, surprisingly classifies AAE samples as racism with a higher rate than SAE (Almost 10 times). However, for both AAE-aligned and SAE-aligned groups, the number of samples assigned to racism class is very low, which can be owing to two presumptions. The first is the characteristics associated with racism samples in training data in Waseem-dataset where the majority of samples comprise religion and anti-Muslim contents, which are totally different from anti-black language used in AAE and SAE groups. The second one is mainly related to contextual knowledge derived from the pre-trained BERT model. We investigated the AAE samples assigned to racism class by trained classifier, without re-weighting mechanism, and most of them contain some racial slurs such as "n*gga" and "'b*tch" that are contextually related to racial contents. However, after applying re-weighting mechanism these numbers of samples are reduced and result in a trade-off between AAE and SAE samples assigned to racism class and alleviating racial bias in our trained classifier with re-weighting mechanism. Although we

achieve a particular reduction in racial bias included in trained classifier by applying the generalization mechanism, reweighting the training data, we believe that still some biases exist in our trained classifiers after reweighting the samples that are associated with the general knowledge of pre-trained BERT model and it should be considered as future work.

Analyzing the samples in AAE group predicted as sexism reveals that our classifier trained on training data without leveraging the re-weighting mechanism, has a high tendency to classify AAE-aligned samples containing common words in AAE language and related to feminism as sexism. However, after reducing the effect of most frequently used n-grams ($n = 2$) in training data with applying the re-weighting mechanism, this likelihood is reduced. As Park et al. [75] asserted the existence of gender biases in Waseem-dataset, it can be inferred that our re-weighting mechanism needs to address the gender bias in training data as long as most frequently used n-grams to alleviate the bias in trained model more efficiently for sexism class.

Turning to the Davidson-dataset, we observed reducing the racial bias for both Hate and Offensive classes after applying the re-weighting mechanism (Table 4.7). Given the words associated with AAE language and highly correlated to the Hate and Offensive classes in Davidson-dataset such as "n*gga" and "b*tch" [121], a substantially higher rate of AAE-aligned samples classified as hate and offensive than SAE-aligned can be justified; where the number of tweets containing "n*gga" and "b*tch" in AAE-aligned samples is thirty and five times more than SAE-aligned samples. As it is noted in [119,121], these kinds of words are common in AAE dialects and used in daily conversations, therefore, it more probably will be predicted as hate or offensive when are written in SAE by associated group.

In summary, we should consider in future studies paying substantial attention to sexual and gender identities as long as dialect and social identity of the speaker in concert with highly correlated n-grams with the negative classes to make the bias alleviation mechanism more precise and effective. On the other hand, using pre-trained language modeling approaches such as BERT may include some general and external knowledge to the classifier, which may be a source of bias itself and it is worth further investigation.

4.3.5 Conclusion

This study reveals that the benchmark datasets for hate speech and offensive language identification tasks are containing oddities that cause a high preference for classifiers to classify some samples to the specific classes. These oddities are mainly associated with a high correlation between some specific n-grams from a training set and a specific negative class. Employing a cross-domain evaluation approach by using the classifiers trained on these datasets, demonstrates some systematic biases in these classifiers. Therefore, we propose a bias alleviation mechanism to decrease the impact of oddities in training data using a pre-

trained BERT-based classifier, which is fine-tuned with a new reweighted training set. The experiments show the ability of the model in decreasing racial bias. We believe our results have made an important step towards debiasing the training classifiers for hate speech and abusive language detection tasks where the systematic bias is an intrinsic factor in hate speech detection systems. An interesting direction for future research would be to consider sexual and gender identities as long as the dialect and social identity of speakers along with n-grams to make the re-weighting mechanism more general and independent from training data. Furthermore, investigating the effect of samples' weights in the compatibility function of the BERT model rather than in the classification loss function maybe improve the result. Most work has so far focused on AAE/SAE language, but it remains to be seen how our debiasing approach or any of the other prior approaches would fare in other cross-domain datasets containing different language dialects.

4.4 Summary and Discussion

To summarize and conclude, this chapter presented a general overview of the second contribution related to the hate speech detection and racial bias mitigation into two parts.

In the first part, we tried to exploit the pre-trained knowledge in different layers of BERT to solve hate speech detection as a downstream task. To that end, we proposed four different strategies to fine-tune BERT model. To evaluated the performance of the model, we used two publicly available dataset from Twitter annotated as racism, sexism, and neither or hate, offensive, and neither, respectively. We investigated the effect of proposed fine-tuning strategies and different portion of available training data on hate speech detection.

In the second part, we examined potential existing bias in the training datasets and proposed a mechanism to ease this bias. To evaluate the bias mitigation mechanism, we used a cross-domain approach in which we use the trained classifiers on the aforementioned datasets to predict the labels of two new datasets from Twitter, AAE-aligned and White-aligned groups, which indicate tweets written in African-American English (AAE) and Standard American English (SAE), respectively.

In the next chapter, we will discuss the identification of hate and offensive content in a low-resource setting, where there is not enough amount of labeled data for hate or offensive content in a specific language.

Chapter 5

Multilingual Hate Speech Detection

Contents

5.1 Overview . . . 102

5.2 Offensive Language Detection in Low Resource Languages: a use case of Persian language . . . 102

5.2.1 Introduction . . . 102

5.2.2 Related Work . . . 104

5.2.3 Dataset Description . . . 106

5.2.4 Methodology . . . 109

5.2.5 Experiments and Results . . . 116

5.2.6 Conclusion . . . 123

5.3 Cross-Lingual Hate Speech Detection using Meta Learning . . . 124

5.3.1 Introduction . . . 124

5.3.2 Related work . . . 125

5.3.3 Methodology . . . 128

5.3.4 Dataset Description . . . 133

5.3.5 Experiments and Results . . . 134

5.3.6 Conclusion . . . 142

5.4 Summary and Discussion . . . 143

5.1 Overview

Different types of abusive content such as offensive language, hate speech, aggression, etc. have become prevalent in social media and many extorts have been dedicated to automatically detect this phenomenon in different resource-rich languages such as English. This is mainly due to the comparative lack of annotated data related to offensive language in low-resource languages. To reduce the vulnerability among social media users with different languages, it is crucial to address the problem of hate speech and offensive language in low-resource languages.

In this chapter, we mainly focus on hate speech in a low-resource setting in which a language lacks manually crafted labeled data sufficient for building an automatic hate speech detection system. In Section 5.2, we provide a dataset for offensive language detection task in Persian as a low-resource language. Then, in Section 5.3, we investigate the problem of limited labeled data in low-resource languages for hate speech and offensive language detection tasks by leveraging a cross-lingual approach based on meta-learning method.

5.2 Offensive Language Detection in Low Resource Languages: a use case of Persian language

5.2.1 Introduction

Although a major research effort has been dedicated into the investigation of hate speech and offensive language in English [2,3,11,132], creating annotated corpora and analyzing hateful and offensive content in other languages such as Danish [12], Italian [133], Spanish [134], Mexican Spanish [135], Greek [136], Arabic [137,138], and Turkish [139] have raised many concerns recently. However to the best of our knowledge, no prior work has contributed in offensive language detection with exploring the Persian language.

In this section, we tackle the problem of offensive language detection in Persian language by introducing the first Persian annotated corpus collected from Twitter and annotated by a team of volunteers. We investigate the usage of monolingual and multilingual pre-trained language models specially ParsBERT (Transformer-based Model for Persian Language Understanding) [140], ALBERT-Persian [141], Multilingual BERT (mBERT) [6], and XLM-RoBERTa [142] along with different ML and DL models in the performance of identifying offensive language in our Persian corpus, as a low-resource language in this area. We compare different classical ML and DL algorithms with monolingual and multilingual pre-trained language models and report the performance results of the different settings and discuss how different approaches perform in identifying offensive language in three levels of annotation schema. In addition, to boost the performance of our classification

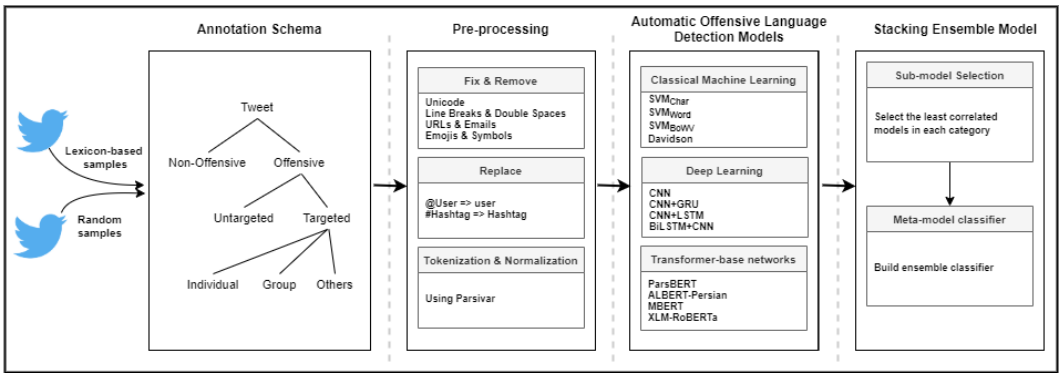


Figure 5.1 – The Proposed workflow of the offensive language detection methodology in Persian language.

task, we introduce an ensemble stacking model in which we leverage the output probability predictions of single classifiers as base-level classifiers to train a meta-level classier to identify offensive vs non-offensive, targeted insult vs untargeted offensive content, and targeted offensive towards individual or group more precisely and robustly.

Figure 5.1 depicts an overview of the proposed framework in this study, at which we address the problem of offensive language in Persian. First, we collect data from Twitter and annotate it according to a three-levels annotation schema. After pre-processing step, different classical ML, DL, and transformer-based neural network models will applied to the annotated corpus to look into the impact of these models in identification of offensive language. Finally, to leverage different strengths and weaknesses of the considered models, we combine them in an ensemble model to improve the performance of the offensive language detection task. The datasets created in this study will be made publicly available at `https://github.com/firstauthorpage/Persian_offensive_language_data`.

The main contributions in this study are as follows:

- Building and sharing the first Persian offensive language corpus along with describing the methodology for collecting data from Twitter and annotation guidelines.

- Performing comprehensive experiments on annotated Persian corpus to investigate the ability of classical ML, DL, and transformer-based neural network models in addressing Persian offensive language identification task in social media. Furthermore, for the first time, we focus on transfer learning approach using advanced monolingual and multilingual pre-trained language models such as ParsBERT, ALBERT-Persian, mBERT, and XLM-RoBERTa for Persian offensive language detection task.

- Introducing a stacked ensemble methodology to improve the performance of the proposed offensive language detection models.

5.2.2 Related Work

In this section, we discuss a concise overview of available corpora for hate speech and offensive language in other languages (low-resource languages) rather than English; where there is no or few labeled data available.

Although many efforts have been dedicated to address the problem of hate speech and offensive language detection in high-resource languages such as English [2, 3, 143], recently concerns have been raised about other languages as well. Emerging recent shared tasks and academic events such as Jigsaw Multilingual Toxic Comment Classification challenge, Automatic Misogyny Identification (AMI) at IberEval [134] and EVALITA [133] including Spanish and Italian languages respectively, identification of offensive language at GermEval [12, 144] in German language, identification of offensive language at SemEval-2019 [143] for English and SemEval-2020 [10] for Arabic, Danish, English, Greek, and Turkish languages, proceedings of the Workshop on Trolling, Aggression and Cyberbullying Workshops [14,145], and proceedings of the Workshop on Abusive Language Online [146–148] shows the raising concerns towards hate speech and offensive language detection in different languages. Table 5.1 summarizes the main concerns of the above events and the datasets and languages that are investigated in these tasks.

Our survey on events and shared tasks show that they mainly focused on different types of this phenomenon such as hate, offensive, misogyny, aggression, etc. in variety of languages, and it indicates that the attentions towards languages with limited resources such as Greek, Arabic, Danish, etc. are increasing, and providing annotated data for abusive content in this kind of languages is principal. Mubarak et al. [137] provided a list of obscene words and hashtags, which are common patterns in offensive and rude communications, from Twitter along with a large corpus of annotated user comments for obscene and offensive language detection in Arabic language. Guellil et al. [152] investigated the problem of hate speech against politicians in YouTube's comments considering comments written with Arabic, Arabizi, Arabic word written with Latin letters, French, and English. Mubarak et al. [138] proposed a method to build an offensive dataset in Arabic language and analyzed the topics, dialects, and gender mostly associated to offensive content. Pitenis et al. [136] introduced the first Greek annotated dataset for offensive language detection on Twitter, named the Offensive Greek Tweet Dataset (OGTD). Experimenting different ML and DL models on Greek offensive language dataset indicated that LSTM and GRU with attention model results in the best performance. Furthermore, a large corpus from Twitter containing 36 232 tweets in Turkish language was created by [139] to address the problem of offensive language in Turkish for the first time. In [135] authors proposed a BERT-based approach along with data augmentation techniques to identify aggressive from non-aggressive tweets written in Mexican Spanish. Considering the automatic detection of hate speech in a code-

Table 5.1 – Shared tasks in identification of abusive language in different types and languages.

Event	Task description	Languages (#sampels)	Platform	year
Kaggle's Toxic Comment Classification Challenge	**Identification of Different Types of Toxicity** | Threats | Obscenity | Insults | Identity-based hate |	English(300k)	Wikipedia	2017
AMI at IberEval [134]	**Automatic Misogyny Identification** **Subtask A** - Misogyny Identification: | Misogyny | Non-misogyny | **Subtask B** - Misogynistic Behavior and Target Classification: 1) *Misogynistic Behavior*: | Dominance| Derailing| Discredit | | Stereotype and Objectification | Sexual Harassment and Threat of Violence | 2) *Target Classification*: | Active (individual) | Passive (generic) |	English(3977) Spanish(4138)	Twitter	2018
AMI at EVALITA [133]	**Automatic Misogyny Identification** **Subtask A** - Misogyny Identification: | Misogyny | Non-misogyny | **Subtask B** - Misogynistic Behavior and Target Classification: 1) *Misogynistic Behavior*: | Dominance | Derailing | Discredit | | Stereotype and Objectification | Sexual Harassment and Threat of Violence | 2) *Target Classification*: | Active (individual) | Passive (generic) |	English(5000) Italian(5000)	Twitter	2018
HaSpeeDe at EVALITA [149]	**Hate Speech Detection on Facebook and Twitter** **Task A** - Hate Speech Detection on Facebook: | Hate | Non-hate | **Task B** - Hate Speech Detection on Twitter: | Hate | Non-hate | **Task C** - Cross-Hate Speech Detection: 1) *Cross-HaSpeeDe-FB*: Train on Facebook and Test on Twitter 2) *Cross-HaSpeeDe-TW*: Train on Twitter and Test on Facebook	Italian: Twitter(4000) Facebook(4000)	Twitter Facebook	2018
TRAC 2018 [145]	**Aggression Identification** | Overtly Aggressive | Covertly Aggressive | Non-aggressive |	English(15000) Hindi(15000)	Facebook	2018
TRAC 2020 [14]	**Aggression Identification** **Subtask A** - Aggression Identification: | Overtly Aggressive | Covertly Aggressive | Non-aggressive | **Subtask B** - Misogynistic Aggression Identification : | Gendered | Non-gendered |	English(5000) Bangla(5000) Hindi(5000)	YouTube	2020
GermEval 2018 [144]	**Identification of Offensive Language** **Subtask A** - Coarse-grained Binary Classification: | Offensive | Non-offensive | **Subtask B** - Fine-grained 4-way Classification: | Profanity | Insult | Abuse | Other |	German(8541)	Twitter	2018
GermEval 2019 [12]	**Identification of Offensive Language** **Subtask A** - Coarse-grained Binary Classification: | Offensive | Non-offensive | **Subtask B** - Fine-grained 4-way Classification: | Profanity | Insult | Abuse | Other | **Subtask C** - Implicit vs. Explicit Classification: | Implicit | Explicit |	German(9915)	Twitter	2019
HASOC 2019 [150]	**Hate Speech and Offensive Content Identification in Indo-European Languages** **Subtask A** - Hate speech and Offensive language identification: | Hate and Offensive (HOF) | Non Hate-Offensive (NOT) | **Subtask B** - Fine-grained 3-way classification: | Hate speech (HATE) | Offenive (OFFN) | Profane (PRFN) | **Subtask C** - Type of Offense Classification: | Targeted Insult (TIN) | Untargeted (UNT) |	English(8000) German(8000) Code-Mixed Hindi(8000)	Twitter Facebook	2019
SemEval 2019 (HatEval) [77]	**Multilingual Detection of Hate Speech against Immigrants and Women** **Subtask A** - Hate Speech Detection against Immigrants and Women: | Hateful | Non-hateful | **Subtask B** - Aggressive Behavior and Target Classification: 1) *Aggression behavior*: | Aggressive | Non-aggressive | 2) *Target Classification*: | Individual | Generic |	English(13000) Spanish(6600)	Twitter	2019
SemEval 2019 (OffensEval) [143]	**Identifying and Categorizing Offensive Language** **Subtask A** - Offensive Language Detection: | Offensive | Non-offensive | **Subtask B** - Automatic Categorization of Offensive: | Targeted Insult | Untargeted | **Subtask C** - Offensive Target Identification: | Individual | Group | Other |	English(14100)	Twitter	2019
SemEval 2020 (OffensEval) [10]	**Multilingual Offensive Language Identification** **Subtask A** - Offensive Language Detection: | Offensive | Non-offensive | **Subtask B** - Automatic Categorization of Offensive: | Targeted Insult | Untargeted | **Subtask C** - Offensive Target Identification: | Individual | Group | Other |	Arabic(10000) Danish(3290) English(14100) Greek(10287) Turkish(35284) + Semi-Supervised OLID English(9089140)	Twitter	2020
OSACT4 [151]	**Arabic Offensive Language Detection** **Subtask A** - Offensive Language Detection: | Offensive | Non-offensive | **Subtask B** - Hate Speech Detection: | Hate | Non-hate |	Arabic(10000)	Twitter	2020

switching environment, where user writes in one language and then switches to another in the same sentence, authors in [153] proposed a pipeline to extract hate speech content in Hindi-English code-switched language (Hinglish) by leveraging profanity modeling, deep graph embeddings, and author profiling.

Low-resource South Asian languages such as Roman Urdu (scripts written in English language characters) and Urdu (scripts written in Urdu language characters) have gained raising attentions recently [154, 155]. Akhter et al. [154] introduced the first annotated corpus for offensive language detection task in Urdu language and provided a profound experiments using ML and DL models to automatically detect abusive comments written in Urdu and Roman Urdu on YouTube's videos. Khan et al. [155] collected and annotated tweets written in Roman Urdu, named Hate Speech Roman Urdu 2020 (HS-RU-20) corpus, in three levels: 1) Neutral or Hostile, 2) Simple or Complex, and 3) Offensive or Hate speech. They applied different ML and DL algorithms including Naïve Bayes, Linear Regression, etc. to investigate the effectiveness of supervised learning techniques for hate speech detection in Roman Urdu.

Finlay, to the best of our knowledge, offensive language detection on Persian language has not been addressed in academic research yet due to the lack of publicly available annotated dataset for offensive or hateful content in this language, and we believe this study provides interesting insights into the research community.

5.2.3 Dataset Description

In this section, we explain the way in which the Persian corpus from Twitter is collected and annotated. In addition, we declare our notice regarding the privacy and ethics aspects of users on Twitter as well as GDPR (General Data Protection Regulation) compliance.

5.2.3.1 Data Collection

We focused on Twitter because it is one of the most widely used microblogging systems and online platforms for sampling offensive and hatred contents in different languages [10], and we retrieved Persian tweets from it using Twitter streaming API. We filtered the stream in Persian language by using both Twister's language identification mechanism (by setting language parameter in the search query as "fa") and some most frequent Persian conjunctions (by setting track parameter in the search query) to prevent crawling samples in other languages similar to Persian such as Urdu. The data was collected using a Python scraper for a two-month interval from June to August 2020. We used two main strategies: (1) random sampling and (2) lexicon-based sampling for data collection, which will be explained in the following.

One of the main difficulties in our data collection process is the fact that Twitter streaming API leads to receiving samples that cover just 1% of all tweets in near to real-time and a very small portion of resulted tweets are included offensive or hatred content usually [156]. To investigate the ability of random sampling tweets in reflecting offensive language in our data collection process, we selected 400 tweets randomly and inspected them by two experts who are native Persian speakers. Scrutinizing randomly sampled tweets by experts revealed that the actual offensive content constitutes a maximum of 2% selected tweets resulting in an unbalanced and inefficient sampling. Furthermore, the vast majority of offensive samples were related to Iranian political parties and governmental issues at that time or a Persian worldwide trending hashtag: (#Don't_execute, #StopExecutionsInIran), which was launched in support of three young protesters in Iran. Therefore, to prevent a bias against some specific topics or targets during data collection, we used a seed lexicon named HurtLex [157], to filter more offensive tweets with diversity in topics and targets. HurtLex[1] is a multilingual computational lexicon of offensive, aggressive, and hateful words organized in 17 categories in over 50 languages including Persian, with two main labels: conservative and inclusive referring to "offensive" senses and "offensive", "not literally pejorative" and "negative connotation" senses, respectively. We considered all conservative and inclusive words in 17 categories as keywords to filter tweets in our lexicon-based sampling strategy. Employing random sampling and lexicon-based sampling leaves us to $320K$ and $200K$ tweets, respectively. Finally, we selected 3000 tweets randomly from each sampling sets (random and lexicon-based) for annotation step.

5.2.3.2 The Ethical Consideration

Although other information rather than tweet' s text such as user demographic statistics, user name, timestamps, location, or social engagement on the platform may result in better understanding of hateful content phenomena, to respect privacy and ethical aspects of users on Twitter as well as GDPR, we did not collect any sensitive and personal information of users. We just collected tweets from public Twitter accounts, eliminating contact information of users, anonymized and converted all mentions containing @username to a specific and fixed term @user. In the open version of dataset, we are going to publish the annotated corpus in terms of "TweetID" and "Label" without the actual text (tweet) and user information.

[1] `https://github.com/valeriobasile/hurtlex`

5.2.3.3 Data Annotation Schema

Abusive language is an umbrella term that encompasses different types of subtasks such as hate speech, cyberbullying, offensive language, etc. with common or different characteristics and there is a considerable overlap between these subtasks. To have some kind of uniform understanding of different subtasks related to abusive language and to prevent overlap of their definition and annotation, Waseem et al. [64] unified these subtasks by proposing a 2-fold typology to categorize abusive language into two majority incorporated groups: (1) the target of abuse (an individual or a group) and (2) the nature of the language (explicit or implicit). In addition, Zampieri et al. [11] considered the problem of abusive language definition as a whole and attempted to model the task hierarchically in which the type and the target of offensive content were identified. They proposed a three-layer hierarchical annotation scheme to label the Offensive Language Identification Dataset (OLID), a new English corpus from Twitter, as offensive or not-offensive, its type, and its target. Therefore, Following [64] and [11], we developed an annotation protocol for our Persian corpus in three levels as follows:

- **Offensive language detection:** in the first step, tweets are distinguished as *offensive* or *non-offensive*. Similar to [11], tweets having any form of explicit or implicit insults, threats, incitement to hatred and violence, dehumanization, or profane language and swear words are considered as *offensive*. On the other hand, tweets without any form of offense, abuse, or profanity are considered as*non-offensive*.

- **Categorization of offensive language:** after discriminating offensive and non-offensive tweets, we categorize the type of offensive tweets as *targeted* or *untargeted*. Offensive tweets without any specific targeted profanity and swearing are considered as untargeted. However, targeted insult refers to any offensive content addressed to an individual, a group, or others.

- **Offensive language target identification:** to make more distinct about the target of offensive contents, similar to [11], we use three target classes: *individual*, *group*, and *other*. If tweets include hateful messages purposely sent to a specific target (e.g., a famous person, a named or unnamed participant in a conversation, etc.), it will be labeled as individual. However, offensive tweets towards many potential receivers as a group of people with the same ethnicity, gender or sexual orientation, political affiliation, religious belief, or other common characteristic are defined as targeted group. Here, we do not consider any crowd of people as a group, but a crowd belongs to a specific unity or individual identity. Therefore, abuse and offense towards some individuals not belonging to our definition of group, is considered as

individual targeted. We consider other category for tweets in which the target of the offensive language does not belong to individual or group categories, and it is a kind of offense toward an organization, event or issue, situation, etc. as non-human entity target. Using different targets of offensive language in our annotation schema results in different concepts of abusive language. For example, offensive tweets targeted at individual is known as cyberbullying whereas insults and threats targeted as a group is defined as hate speech.

5.2.3.4 Annotation Process

Since offensive language is a subjective and contextual-based concept that may differ from person to person, culture to culture, or society to society, and Persian is a low-resource language with fewer speakers all over the world in comparison with high-resource languages (i. e., English), employing Persian native speakers for annotating the corpora is crucial. Therefore, we use expert-level annotation approach as a common approach used for annotating low-resource data previously [136,139,154] to annotate the Persian corpus. Therefore, three highly educated volunteers from the author' s personal contacts, who were Persian native speakers, were enrolled for annotating the corpus. Two of annotators were supposed to annotate all the selected tweets at three levels offensive language detection, categorization, and target identification and in the case of agreement the final label was set. Otherwise, the third annotator was asked for labeling the tweet again and then we took a majority vote. Owing to the subjectivity of offensive language identification in three levels of annotation schema and lack of context in tweets, as a short textual data, annotating process is challenging with low inter-annotator agreement. Annotation consensus for two annotators on three levels of annotation schema was approximately 73%, in which the agreements in the first level of annotation schema (offensive vs non-offensive) was very high as 86%, in the second level 75%, and in the third level 60%. In the case of disagreement, the third annotator judged. The distribution of labeled data in the three levels of annotation is presented in Table 5.2.

Few examples of annotated instances along with their categories for each level of the annotation schema are presented in Figure 5.2. We include both Persian and its English translated version of tweets for ease of reading.

5.2.4 Methodology

In this section, we explain in details different classical machine learning algorithms, deep learning, and transfer learning approaches along with different feature engineering techniques used in this study. Furthermore, we introduce a new meta-model based on an ensemble learning technique to identify offensive language more precisely.

Table 5.2 – Distribution of annotated data in three levels of annotation schema. A set of 6k out of 520k sampled data is randomly selected for annotation process.

Level-1	**Level-2**	**Level-3**	**#Samples**
Offensive	Targeted	Individual	702
Offensive	Targeted	Group	672
Offensive	Targeted	Other	38
Offensive	Untargeted	-	212
Non-Offensive	-	-	4376
Total	-	-	6000

	Level-1	**Level-2**	**Level-3**
@user: زر نزن، انقد زر نزن میمون. @user: Do not bullshit, do not bullshit so much monkey.	Offensive	Targeted	Individual
@user: بچه خواهرم پرسید؛ دایی سلبریتی یعنی چی؟ گفتم یعنی شغال یعنی کفتار. بچه ها باید از کودکی بفهمند این جماعت خوی حیوانیت دارند. @user: My nephew asked me: what does the celebrity mean, uncle? I said: it means a jackal, it means a hyena. Children should know that this group has brutality.	Offensive	Targeted	Group
@user: ر*دم به هر چی نهاد بین‌المللی و بی‌بی‌سی و هر ک*فتی که هست. @user: I am gonna sh*t on every international institution and BBC agency and whatever the sh*t it is.	Offensive	Targeted	Others
@user: ما رو که گ*ییدن! @user: They f*cked us!	Offensive	Untargeted	____
@user: بعد از سی روز بیای سر کار ببینی به گلات آب ندادن، شاکی نمیشی؟ @user: Do not you complain if you come to work after thirty days and see that no one waters your plants?	Not-Offensive	____	____

Figure 5.2 – Tweet samples (original and translated) from the annotated data with their categories for each level of the annotation schema.

5.2.4.1 Classical Machine Learning Models

Initially, we start with a simple linear SVM classifier, as a well performed classifier in this task according to the literature [68], trained with different tweet representations as feature embeddings. We use a set of three feature extraction methods TF-IDF on character n-grams and word n-grams, where n-grams are a contiguous sequence of n characters or words), and Bag of Words Vectors (BoWV) over fastText.

Features: to extract character n-grams, we consider $n = 2$ to $n = 5$. For word n-grams we consider $n = 1$ to $n = 2$ and extract word unigrams and bigrams in each tweet and eliminate words with more that 70% of frequency occurrence in all corpus. At the end, using TF-IDF all word and character n-grams are normalized. We use a logistic regression with L1 regularization, to reduce the dimensionality of the feature vectors of TF-IDF character and word n-grams. Considering the co-occurrences of each word in each document (tweet) in our annotated corpus, we create a document-term matrix and use the pre-trained word embeddings fastText with an embedding dimension of 300 to get initial vector representation of each word in tweet. The fastText is a static word embeddings representation of tweets that is pre-trained on Persian version of Common Crawl and Wikipedia[2] using fastText model [158]. The average of fastText vector of words in each tweet is consider as tweet representation.

To investigate the impact of other text-mining features such as sentiment analysis scores, linguistic features, etc. on offensive language detection in Persian, we re-implement a state-of-the-art SVM-based classifier proposed by Davidson et al. [3] and map its feature extraction part in Persian language. Therefore, different features are extracted using Parsivar[3] Python package [159]. We normalize and tokenize each tweet and calculate: TF-IDF weighted word n-grams (unigram, bigram, and trigram); number of characters, words, and syllables in each tweet; number of user mentions, hashtags, retweets, URLs; TF-IDF weighted of Part Of Speech (POS) tag n-grams (unigrams, bigrams, and trigrams of POS tags) in which we filter any candidates with a document frequency lower than 5. Using pertimental[4] Python package, we also calculate sentiment polarity scores of each tweet as Negative, Positive, and Neutral. Furthermore, readability scores of each tweet are measured using two metrics Flesch-Kincaid Grade Level and Flesch Reading Ease, with common core measures (words and sentences' length) and different weighting factors, to indicate how difficult a tweet in Persian is to understand. To calculate these scores, we consider the number of sentences in each tweet as fixed number one. After reducing the dimensionality of extracted feature vector using a logistic regression with L1 regularization, we apply a

[2] `https://fasttext.cc/docs/en/crawl-vectors.html`
[3] `https://github.com/ICTRC/Parsivar`
[4] `https://github.com/pbarjoueian/pertimental`

Logistic Regression with L2 regularization algorithm to train our classifier.

Thus, we define multiple classifiers named $\mathbf{SVM_{Character\ n\text{-}grams}}$, $\mathbf{SVM_{Word\ n\text{-}grams}}$, and $\mathbf{SVM_{BoWV}}$ accompanying **Davidson** algorithm as classical ML approaches.

5.2.4.2 Deep Learning Models

We employ a static word embeddings (i. e., fastText) representation of tweets to train different DL models combining Convolutional Neural Networks (CNN), Recurrent Neural Networks (GRU), and Long Short-Term Memory networks (LSTM).

Following previous studies on different publicly available datasets in this domain, we implement different DL models proposed by [4,160,161] on Persian annotated data. Authors in [160], proposed a CNN model trained on different features such as character n-grams, word vector embeddings, randomly generated word vectors, and a combination of character n-grams and word vectors to study the problem of hate speech identification. Here, we just use the fastText embeddings of words as word feature vectors, based on semantic information, to train a CNN model. The input of the model uses a 1D convolutional layer with 64 filters with a window size of 4, and it is converted into a fixed length vector using a pooling layer. Then, we add a max pooling layer with a pool size of 2 to capture the most important latent semantic features from the input tweets' sequences. We use the Rectified Linear Unit (ReLU) activation function for CNN layers. To provide output in the form of probabilities for each of two classes in our binary classification task, we use a softmax activation function in the output layer. Finally, we compile the model by adjusting three parameters: loss, optimizer, and metrics. A binary cross-entropy loss function is used along with the Adam optimizer to adjust the learning rate throughout the training and the accuracy (as metric).

CNN together with Gated Recurrent Units (GRUs) [4] or LSTM [161] have also been explored as potential solutions in hate speech detection for other languages. Inspiring [4], we create a deep neural network combining convolutional and GRU neural networks. As the embedding layer, we use the pre-trained fastText embeddings to map each word in tweets' sequences into a fixed dimensional real vectors. To avoid from overfitting, we add a drop-out layer with a rate of 0.2. Then, a 1D convolutional layer with 100 filters with a window size of 4 is added accompanying a ReLU activation function. To reduce the size of each feature map, the amount of parameters and computations, we add a 1D max pooling layer with a pool size of 4. Then, the extracted features are fed into the GRU layer. Using a global max pooling layer, the highest values in each feature dimensions are selected and the output vector is fed into an output layer with a softmax activation function. To train the model and predict probability distribution over two classes, we use the binary cross-entropy loss function and the Adam optimizer.

To create a deep neural network combining convolutional and LSTM neural networks, we use the network architecture proposed in [161]. All the layers, structures, and parameters of this model is the same as CNN+GRU model except for GRU layer. Here, in CNN+LSTM model, we add a LSTM layer instead of GRU to the model.

In addition to the aforementioned models, we introduce a model by combining a bi-directional LSTM (BiLSTM) and CNN networks. As the embedding layer, we use the pre-trained fastText embeddings to map each word in tweets' sequences into a fixed dimensional real vectors. To avoid from overfitting, we add a drop-out layer with a rate of 0.2. A bi-directional LSTM layer with 128 units followed by a 1D convolutional layer with 100 filters with a window size of 2 is added. The output of CNN layer is average-pooled and max-pooled globally and the results are concatenated. Then, Features encoded by CNN layer are fed into a dense layer with 64 units and the ReLU activation function. The dense layer is followed by the output layer with softmax activation function. The network is compiled with a binary cross-entropy loss function and the Adam optimization algorithm.

Thus, we define multiple classifiers named **CNN**, **CNN+GRU**, **CNN+LSTM**, and **BiLSTM+CNN** as DL approaches to identify offensive language. Our main intuition behind using these neural network architectures is to include both local and global contextual features in our offensive language detection problem. The convolution layer (CNN) will extract local and position-invariant features whereas the LSTM layer considers a long range of context dependencies, semantically, rather than local key-phrases.

5.2.4.3 Monolingual and Multilingual Transformer-Based Networks

Here we utilize different transformer-based models (e.g., ParsBERT, ALBERT-Persian, mBERT, and XLM-RoBERTa) and fine-tune different pre-trained contextual representations by training them on our offensive language detection task's data. Table 5.3 summarizes the information of different models used in this study, including their configuration, learning parameters, and training corpora.

ParsBERT [140]: In this approach, we use a monolingual BERT model pre-trained on large corpora from numerous subjects (e.g., scientific, novels, and news) with more than 2M documents, crawled from Internet's web pages in Persian language called ParsBERT[5]. ParsBERT is a monolingual pre-trained language model based on BERT architecture with the same configurations as $\text{BERT}_{\text{base}}$ [6] for Masked Language Modeling (MLM) and Next Sentence Prediction (NSP) tasks. Before fine-tuning ParsBERT model in our down stream task, we first format the input sequences in such a way that each sequence is splitted into tokens, perpended with the classification token [CLS] to the start and appended the [SEP] token to the end. Then, the sequences are padded to the fixed maximum length

[5] `https://github.com/hooshvare/parsbert`

Table 5.3 – Description of the transformer-based neural network models used in identification of offensive language in Persian.

Name	Provider	Architecture	Method	Configuration	Training corpora
ParsBERT	Hooshvare Lab	- Google's BERT -Transformer-based - Monolingual	- Masked Language Modeling - Next Sentence Prediction	hidden layers: 12 attention heads: 12 hidden sizes: 768 parameters:110M vocabulary: 100K	Persian corpora (14GB): Wikidumps, MirasText, and six manually crawled text data from a various type of websites
ALBERT-Persian	Hooshvare Lab	- Google's ALBERT base - Transformer-based - Monolingual	- Masked Language Modeling - Sentence Ordering Prediction	hidden layers: 12 attention heads: 12 hidden size: 768 parameters: 12M vocabulary: 100K	Persian corpora (14GB): Wikidumps, MirasText, and six manually crawled text data from a various type of websites
mBERT	Google	- Transformer based - Multilingual	- Masked Language Modeling - Next Sentence Prediction	hidden layer: 12 attention heads: 12 hidden sizes: 768 parameters: 172M vocabulary: 110K	Entire Wikipedia dump: 104 languages
XLM-RoBERTa	Facebook AI team	- Transformer based - Multilingual	- Translation Language Modeling - Causal Language Modeling - Masked Language Modeling	hidden layer: 12 attention heads: 12 hidden size: 768 parameters: 270M vocabulary: 250K	CommonCrawl data (2.5TB): 100 languages.

of input sequences and attention masks are added to them. Here, we set the maximum sequence length to 128. After feeding input data to the pre-trained model, additional untrained classification layer will be trained for the downstream task. We consider the final hidden state corresponding to the classification token ([CLS]) as the aggregate sequence representation for our offensive language detection classification task. Therefore, we fine-tune ParsBERT on the input data splitted into 90% and 10% as training and validation sets, respectively, by just adding an output layer as a single linear classifier on top of the pre-trained BERT model.

ALBERT-Persian [141]: is a monolingual pre-trained language model with A Lite BERT (ALBERT) architecture [52] which is trained on a massive amount of Persian public corpora, Persian Wikidumps and MirasText, and six other manually crawled text data from a various type of Persian websites: BigBang Page scientific, Chetor lifestyle, Eligasht itinerary, Digikala digital magazine, Ted Talks general conversational, Books novels, storybooks, and short stories from old to the contemporary era. ALBERT-Persian has significantly fewer parameters than a traditional BERT architecture. We fine-tune ALBERT-Persian by exactly the same way as ParsBERT.

mBERT [6]: is a multilingual task-agnostic language representation model with a 12 layer bidirectional transformer trained on Wikipedia pages of 104 languages with a shared word piece vocabulary. This model that is pre-trained in two tasks masked language model and next sentence prediction can be fine-tuned for text classification in any of 104 languages including Persian as well. Here we use mBERT to circumvent having to train a monolingual model for Persian language as a low-resource language and fine-tune it using a single linear classifier on top of the model.

XLM-RoBERTa [142]: is a transformer-based multilingual masked language model

pre-trained on 100 languages, including Persian, using more than two terabytes of filtered CommonCrawl data. To fine-tune XLM-RoBERTa model on our target classification task, we add a linear layer on top of the pooled output, same as previous models.

As showed in Table 5.3, we use the Base version of ParsBERT, ALBERT-Persian, mBERT, and XLM-RoBERTa pre-trained models and more details regarding fine-tuning the models and hyperparameters used in this study are included in Section 5.2.5.1.

5.2.4.4 Stacking Ensemble Model

Apart from single classifiers in our classical ML, DL, and monolingual and multilingual transformer based neural network approaches, we use an ensemble learning technique to improve accuracy of the offensive language detection task with a combination of the aforementioned classifiers. Ensemble learning is a technique in which applying multiple learning algorithms and aggregating their decisions somehow results in better predictive performance than using any of constituent learning algorithms alone [162]. A variety of ensemble techniques have been applied in different applications and problems [163], specifically in offensive language detection [17], aggression identification [164], and hate speech detection [68] to achieve better performance to single classification methods.

According to the feature extraction and learning mechanisms, different aforementioned classifiers capture different aspects of offensive language detection task. For instance, classical ML approaches advantage syntactical and hand crafted features such as character and word n-grams, number of hashtags and mentions, number of exclamation marks, etc. to understand obfuscated and complex words, but they cannot capture contextual or semantical aspects of offensive language in social media content. On the other hand, in offensive language, context is very domain specific and a lack of vector embedding for some words in fastText pre-trained embeddings may effect in ML performance while DL models may suffer from generalization due to the lack of enough training data. Furthermore, transformer based pre-trained language models (e.g., BERT) have a pre-knowledge of a large corpus and can deal with context better even when there is not a large amounts of annotated data [72]. Hence, different advantages and drawbacks of different classical ML, DL, and transformer-based pre-trained language models prompt us to make an ensemble classifier out of them to improve the performance of offensive language detection task.

Here we use a stacking ensemble technique in which, using a parallel architecture, all classifiers called base-level classifiers are performed independently and their predictions are fed in a meta learner called meta-level classifier to learn how to best combine the predictions from the classifiers. We consider different models in classical ML, DL, and transformer based neural networks as base-level classifiers and a SVM as meta-level classifier. The stacking ensemble algorithm is summarized in Algorithm 1. First, we extract all features for different

Algorithm 1: Stacking with K-Fold Cross Validation

Input : Training data $\mathcal{D} = \{x_i, y_i\}_{i=1}^{m}$, where $x_i \in$ labeled data and $y_i \in [0, 1]$, and T base-level classifiers
Output: An ensemble meta-level classifier H

1 ***Step 1:*** *Extract required features for classical ML and DL algorithms in base-level classifiers*
2 ***Step 2:*** *Adopt a cross validation approach to prepare a training set for meta-level classifier*
3 Randomly split $\mathcal{D}$ into K equal-size subsets: $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_K\}$
4 **for** $k \leftarrow 1$ *to* K **do**
5 | *Step 2.1: Learn base-level classifiers*
6 | **for** $t \leftarrow 1$ *to* T **do**
7 | | Learn a classifier h_{kt} from $\mathcal{D} \setminus \mathcal{D}_k$
8 | **end**
9 | *Step 2.2: Construct a training set for meta-level classifier*
10 | **for** $x_i \in \mathcal{D}_k$ **do**
11 | | Get a record $\{x'_i, y_i\}$, where $x'_i = \{h_{k1}(x_i), h_{k2}(x_i), ..., h_{kT}(x_i)\}$
12 | **end**
13 **end**
14 ***Step 3:*** *Select $T/2$ of base-level classifiers as T'*
15 Select two least correlated base-level classifiers in each model category: Classical ML, DL, and Transformer-based DL
16 ***Step 4:*** *Learn a meta-level classifier among with selected base-level classifiers*
17 Learn a new classifier h' based on the newly constructed dataset: $\{x'_i, y_i\}$, where x'_i is from T'
18 ***Step 5:*** *Re-Learn base-level classifiers*
19 **for** $t \leftarrow 1$ *to* T' **do**
20 | Learn a classifier h'_t based on $\mathcal{D}$
21 **end**
22 **Return** $H(x) = h'(h_1(x), h_2(x), ..., h_{T'}(x))$

This algorithm is updated from the original stacked ensemble algorithm proposed in [165].

classical ML models, and prepare the input of DL and transformer-based neural networks according to previous subsections. Given the labeled dataset, we use a k-fold cross validation approach on entire data to learn base-level classifiers separately and use their out-of-fold predictions as training features for meta-level classifier.

5.2.5 Experiments and Results

This section presents an extensive set of experiments in identification of offensive language in our Persian corpus comprising of classical ML, DL, transformer-based models, and the introduced ensemble model. Three binary classifiers are trained for different levels of annotation. The first classifier discriminates *offensive* tweets from *non-offensive*, the second one determines whether an offensive tweet is *targeted* or *untargeted*, and the third one predicts the target of offensive content as *individual* or *group*. We eliminated tweets labeled as

other due to the sparsity of this class in our dataset (only 38 tweets), and considered only individual and group classes in the third classifier.

Pre-processing After collecting and annotating our Persian corpus, we perform several text pre-processing steps including: 1) using Parsivar NLP Toolkit for normalizing and tokenizing the text; 2) fixing Unicode; 3) removing line breaks, double spaces, emails, URLs, currency symbols, all tweet specific tokens (namely mentions, re-tweet tag, etc.), emoji, punctuation marks, numbers, and non-Persian characters; 4) removing hashtag sign and replacing the hashtag texts by their textual counterparts. In Persian, generally, a hashtag is concatenated with multiple words separated by '_'. Therefore, we split the strings after '#' symbol into their constituent words by removing '_'; 5) correcting the spell of words using SpellCheck module in Parsivar NLP Toolkit. We keep stop words in tweets to extract more contextual informations from pre-trained language models such as ParsBERT. To fine-tune the transformer-based models, we use specific tokenizer and vocabulary provided by pre-trained models and we did not remove punctuation marks and numbers.

5.2.5.1 Training Procedure

All classical ML models are performed using scikit-learn[6] python package. The word embeddings dimension in SVM_{BoWV} is fixed to 300. For DL models, we use Keras[7] python package. The initial embedding layer is seeded with a matrix of one embedding in size 300, derived from Persian pre-trained fastText, for each word in the training dataset. All input tweets are padded to sequences of 128 words and in case of a longer or shorter length, truncating or padding with zero values will be applied, respectively. Models are trained for 8 epochs with a batch size of 16. The learning rate of Adam optimizer is set to 1e-5 leading to obtain more accurate results.

Transformer-based models are fine-tuned employing publicly available transformers[8] library for Pytorch (namely pytorch-pretrained-bert). For all considered pre-trained language models, we utilize the pre-trained model, text tokenizer, and pre-trained WordPiece provided in each pre-trained model to prepare the input sequences for training. The maximum sequence length of the input sentences is set to 128 and in case the input length is shorter or longer, it will be padded with zero values or truncated to the maximum length, respectively. Models are fine-tuned with a batch size of 16 for 3 epochs. An Adam optimizer with a learning rate of 2e-5 is used to minimize the Cross-Entropy loss function. Furthermore, the dropout probability is set to 0.1 for all layers. As offensive language detection

[6] `https://github.com/scikit-learn/scikit-learn`
[7] `https://github.com/keras-team/keras`
[8] `https://github.com/huggingface/transformers`

is a classification task, we directly modify and fine-tune classification classes of each model (*BertForSequenceClassification* in models with BERT architecture and *XLMRobertaForSequenceClassification* for XLM-RoBERTa model, in which a linear classification layer is added on top of the pooled output.

As the implementation and execution environment, we use Google Colaboratory tool with an NVIDIA Tesla K80 GPU and 12G RAM.

To evaluate different models in a single or ensemble configuration, we use a k-fold cross validation approach. Due to the imbalance data that we have, we use Precision (P), Recall (R), and F1-score (F1) per class and macro-averaged F1-score as performance evaluation metrics. For classical ML models, we split data to train and test sets by 0.8 and 0.2. To train DL models and fine-tune other models based on monolingual or multilingual pre-trained language models, we consider 0.1 of train set as dev set for hyper-parameter tuning. The reported results are based on the test set.

5.2.5.2 Single Models Results

Regarding the classical ML, DL, and transformer-based models described in Section 5.2.4, the first experiment aims to assess and compare the performance of different models along with different features in offensive language detection task in three different levels (*offensive* vs *non-offensive*, *targeted* vs *untargeted*, and *individual* vs *group*). The results of the experiments under k-fold cross-validation ($k = 5$) for three classifiers are demonstrated in Tables 5.4, 5.5, and 5.6 in terms of P, R, F1, and macro F1-score. In all tables, first column indicates the category of trained classifier using classical ML, DL, or transformer-based neural network algorithms. Second and third columns show performance of classifiers per each class in different annotation levels, respectively. Final column indicates the macro-averaged F1-score.

Regarding Table 5.4, among all models, word n-grams are the most discriminative features for identification of offensive content in Persian where SVM classifier trained on word n-grams ($n = 1$ to $n = 2$), achieves the best performance 90.8% in terms of macro F1-score. The second best performing model is ParsBERT obtaining 87.8%, following SVM_{Char} and ALBERT-Persian with macro F1-score 87.0% and 84.9%, respectively. Among DL models, CNN+GRU outperforms other models with F1-score 82.7% which confirms the results of previous study [4] for English offensive language detection task. Although SVM_{Word} outperforms other models, a possible reason can be the problem of over-fitting in this traditional classification technique. Comparing the results of different DL and transformer-based models specifies that pre-trained language models such as ParsBERT and ALBERT-Persian that rely on their pre-knowledge existing in their embeddings layers perform better than DL models with fastText embeddings for each word. Furthermore, we can see that all models

Table 5.4 – Results of offensive language identification (first level). The bold and underline numbers represent the first and second best scores, respectively, in each category: classical ML, DL, and transformer-based neural networks.

Model		Non-Offensive			Offensive			
		P	**R**	**F1**	**P**	**R**	**F1**	**F1 Macro**
		Baselines						
Classical ML	SVM_{Char}	0.944	0.971	0.958	0.844	0.730	0.783	<u>0.870</u>
	SVM_{Word}	0.995	0.930	0.961	0.759	0.979	0.855	**0.908**
	SVM_{BoWV}	0.911	0.975	0.942	0.841	0.574	0.682	0.812
	Davidson [3]	0.896	0.985	0.938	0.866	0.460	0.601	0.770
DL	CNN [160]	0.909	0.969	0.938	0.807	0.567	0.666	<u>0.802</u>
	CNN + GRU [4]	0.925	0.959	0.942	0.782	0.655	0.713	**0.827**
	CNN + LSTM [161]	0.889	0.970	0.927	0.753	0.429	0.547	0.737
	BiLSTM + CNN	0.907	0.975	0.939	0.846	0.545	0.652	0.796
	Monolingual/Multilingual language models							
Transformer-based DL	ParsBERT	0.953	0.959	0.956	0.812	0.790	0.801	**0.878**
	ALBERT-Persian	0.930	0.971	0.950	0.840	0.675	0.749	<u>0.849</u>
	mBERT	0.902	0.928	0.915	0.609	0.528	0.566	0.740
	XLM-RoBERTa	0.881	0.935	0.906	0.562	0.373	0.411	0.659

perform better at identifying non-offensive content compared to offensive where P, R, and F1-score of Non-offensive class are higher than offensive class.

Table 5.5 – Results of targeted offensive language identification (second level). The bold and underline numbers represent the first and second best scores, respectively, in each category: classical ML, DL, and transformer-based neural networks.

Model		Untargeted			Targeted			
		P	**R**	**F1**	**P**	**R**	**F1**	**F1 Macro**
		Baselines						
Classical ML	SVM_{Char}	0.760	0.904	0.826	0.983	0.951	0.967	**0.896**
	SVM_{Word}	0.645	1.000	0.784	1.000	0.912	0.953	<u>0.869</u>
	SVM_{BoWV}	0.440	0.130	0.184	0.859	0.971	0.910	0.547
	Davidson [3]	0.529	0.478	0.482	0.912	0.908	0.909	0.695
DL	CNN [160]	0.573	0.115	0.180	0.859	0.979	0.914	<u>0.547</u>
	CNN + GRU [4]	0.496	0.203	0.271	0.867	0.963	0.911	**0.591**
	CNN + LSTM [161]	0.269	0.350	0.304	0.890	0.848	0.868	0.586
	BiLSTM + CNN	0.461	0.200	0.279	0.818	0.939	0.874	0.576
	Monolingual/Multilingual language models							
Transformer-based DL	ParsBERT	0.533	0.402	0.457	0.907	0.944	0.925	**0.691**
	ALBERT-Persian	0.347	0.186	0.227	0.868	0.974	0.917	<u>0.572</u>
	mBERT	0.261	0.117	0.157	0.837	0.984	0.904	0.531
	XLM-RoBERTa	0.000	0.000	0.000	0.862	1.000	0.925	0.462

Although the binary classification in the first level of annotation, offensive vs non-offensive, is an important task with a high performance, going deeper in the classification

Table 5.6 – Results of target type of offensive language identification (third level). The bold and underline numbers represent the first and second best scores, respectively, in each category: classical ML, DL, and transformer-based neural networks.

Model		Individual			Group			
		P	**R**	**F1**	**P**	**R**	**F1**	**F1 Macro**
Baselines								
Classical ML	SVM_{Char}	0.800	0.903	0.848	0.877	0.754	0.811	**0.829**
	SVM_{Word}	0.699	0.584	0.633	0.612	0.730	0.662	0.648
	SVM_{BoWV}	0.739	0.721	0.724	0.694	0.720	0.701	0.712
	Davidson [3]	0.887	0.983	0.933	0.849	0.427	0.568	<u>0.751</u>
DL	CNN [160]	0.711	0.711	0.702	0.677	0.678	0.667	<u>0.685</u>
	CNN + GRU [4]	0.784	0.772	0.778	0.722	0.735	0.728	**0.753**
	CNN + LSTM [161]	0.657	0.707	0.681	0.612	0.555	0.582	0.632
	BiLSTM + CNN	0.676	0.741	0.707	0.686	0.614	0.648	0.677
Monolingual/Multilingual language models								
Transformer-based DL	ParsBERT	0.753	0.833	0.787	0.786	0.702	0.736	**0.772**
	ALBERT-Persian	0.765	0.790	0.777	0.763	0.736	0.749	<u>0.763</u>
	mBERT	0.716	0.693	0.704	0.672	0.696	0.684	0.694
	XLM-RoBERTa	0.521	0.891	0.654	0.293	0.108	0.106	0.374

of targeted insult vs untargeted offensive content in the second level of annotation is more challenging. Given Table 5.5, it is obvious that different classifiers with different features have lower results in identifying whether an offensive tweet is a targeted insult towards an individual or group or it is an untargeted one with general abuse content.

The best macro F1-score, 89.6%, is achieved by training a SVM classifier on character n-grams ($n = 2$ to $n = 5$) features. Model trained using word n-grams ($n = 1$ to $n = 2$) follows this number by achieving 86.9%. Both ParsBERT and Davidson models provide nearly the same results whereas in DL models there is roughly a 14% reduction (or decrease) in the performance. Among transformer-based models, monolingual pre-trained language models ParsBERT and ALBERT-Persian outperform multilingual pre-trained language models mBERT and XLM-RoBERTa by achieving macro F1-score 69.1% and 57.2% in comparison with 53.1% and 46.2%, respectively. On the other hand, XLM-RoBERTa as a multilingual pre-trained language model performs the worst among all cases. Although the imbalance data that we have in the second level of annotation gives rise to decreasing performance among DL and transformer-based models, mBERT are better than XLM-RoBERTa in capturing contextual information in Persian as a low resource language.

The results from Table 5.6 show that identifying target of offensive content as individual or group in the third level of annotation are more precise than the second level classification results, especially for DL and transformer-based models. SVM classifier trained on character n-grams ($n = 2$ to $n = 5$) performs the best and XLM-RoBERTa performs the worst among all models. Again pre-trained language model ParsBERT surpasses other transformer-based

and DL models by achieving 77.2% F1-score, where ALBERT-Persian, CNN+GRU, and Davidson follow it by achieving 76.3%, 75.3%, and 75.1%.

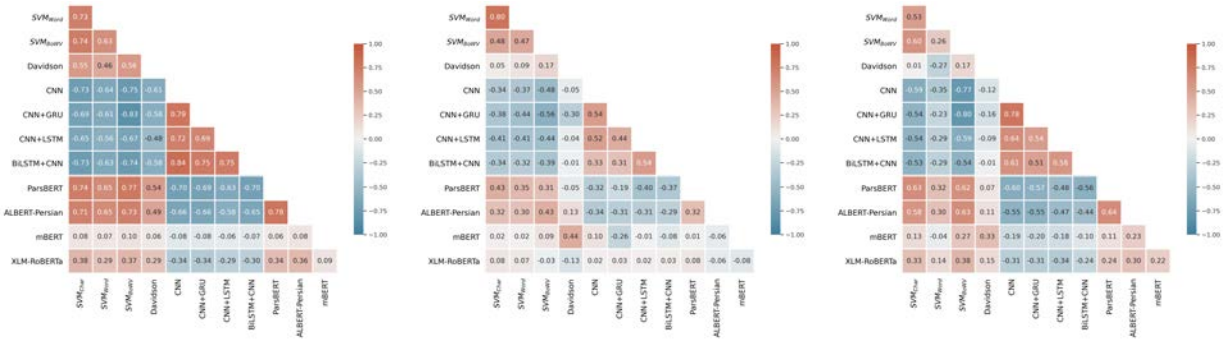
Overall we observe that there is no one single model outperforming others in identification of offensive vs non-offensive content, targeted insult vs untargeted offense, and targeted offensive language as individual or group. However, SVM trained on character and word n-grams seems to be reliable in most cases where pre-trained language model ParsBERT is the second model with promising results in all three levels of classifications. On the other hand, we believe that the performance of DL models could be improved with a larger amount of labeled data in Persian offensive content, in company with better word embeddings such as fastText embeddings trained on a specific Persian textual content of social media. Generally, it can be conveyed that it is not easy to distinguish between targeted insult or thread and untargeted offensive language by applying the single models where the performance metrics of Untargeted class, in Table 5.5, are lower than Targeted class.

5.2.5.3 Ensemble Model Results

In the second experiment, we investigate the stacked ensemble classifier using a combination of individual classical ML, DL, and transformer-based classifiers. Firstly, we divide input data into an 90:10 split as training and hold-out test sets. Then, we run k-fold cross validation ($k = 5$) on training set to create out-of-fold predictions per each model as base-level classifier. These predictions will be selected and used as training features for meta-level classifier. To create test features for meta-level classifier, we make predictions for the test set (in each fold) and average all 5 predictions per model.

As mentioned in Section 5.2.4, different models capture different characteristics of offensive language and they are skillful on this task in different ways. Therefore, obtaining an appropriate combination of base-level classifiers for ensemble learning is a challenge. As the training data for meta-level classifier is generated from the probability predictions of base-level classifiers' outputs, we consider the correlation between predicted probabilities of each classifier as a linear discriminative metric for base-level models selection.

As depicted in Figure 5.3, we examine the pairwise Pearson Correlation Coefficient between the predicted probabilities of all base-level classifiers in three levels of annotation. The values range between -1.0 and 1.0 where a value of 1.0 indicates a total positive linear correlation, 0.0 shows a no linear correlation, and -1.0 shows a total negative linear correlation. Here, we consider positive linear correlation or no linear correlation scores for base-level model selection and do not include models with negative linear correlation in generating the ensemble model. From Figure 5.3a, it is observed that different classical ML, DL, and transformer-based models have different correlations. In classical ML and DL models, there are positive correlations between models' predictions whereas in transformer-



(a) First level (b) Second level (c) Third level

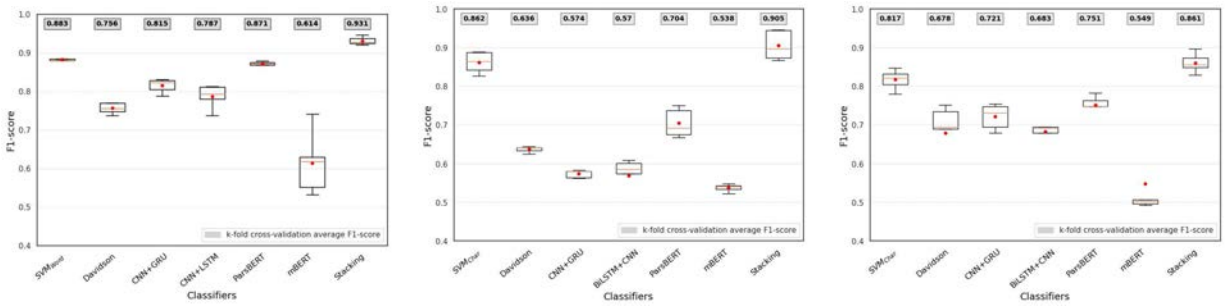
Figure 5.3 – Pairwise Pearson Correlation Coefficient between the predicted probabilities of different single classifiers on out-of-fold test set. First level (a) shows the correlation between the output predictions of classifiers trained on offensive vs non-offensive annotated data. Second level (b) shows the correlation between the output predictions of classifiers trained on targeted vs untargeted samples. Third level (c) shows the correlation between the output predictions of classifiers trained on targeted offensive towards individual or group.

based models this value is low except for ParsBERT and ALBERT-Persian models. This is the same for classifiers in the second-level and third-level of annotation in Figures 5.3b and 5.3c except for SVM_{Word} and Davidson models in the third-level. CNN and BiLSTM+CNN models have the highest correlation contrary to SVM_{BoWV} and CNN+GRU models that have the lowest correlation among all first-level classifiers. In the second-level, SVM_{Char} and SVM_{Word} models have the highest correlation where for the third-level CNN and CNN+GRU models are the most correlated.

In this study, we presume that the low correlated or uncorrelated classifiers with high macro F1-score complement each other in an ensemble configuration. Hence, we select two least correlated models in each classical ML, DL, and transformer-based categories as the input of meta-level classifier in stacked ensemble model.

Figure 5.4 demonstrates the comparison of the ensemble classifier with the individual classifiers selected based on their correlations. It shows the distribution of macro F1-scores in k-fold cross-validation ($k = 5$). Different individual classifiers that are selected as the base-level classifiers for ensemble stacking among with their performance on out-of-fold test set in terms of averaged-macro F1-score are depicted in Figures 5.4a, 5.4b, and 5.4c for three levels of annotation. For more emphasis, we include the average of macro F1-scores from k-fold cross-validation runs and compare the final performance based on that.

For the classification task in the first level of annotation, as shown in Figure 5.4a, we can see that the macro F1-score of offensive vs non-offensive language detection task has increased by 5% of its value where the best performing base-level classifier, SVM_{Word},



(a) First level (b) Second level (c) Third level

Figure 5.4 – Offensive language identification performance among all models in three levels of annotation. First level (a), Second level (b), and Third level (c) indicate performance of selected base-level classifiers accompanying stacking ensemble classifier in identification of offensive vs non-offensive, targeted vs untargeted offensive content, and the target of offensive language towards individual or group, respectively.

achieved 88.3% while stacking ensemble classifier achieves 93.1%. As shown in Figure 5.4b, for the second level classifier, stacking ensemble model outperforms all single base-level classifiers by achieving 90.5% F1-score while on the contrary the best performing base-level classifier, SVM_{Char}, achieves 86.2%. As shown in Figure 5.4c, ensemble stacking classifier outperforms the best performing single base-level classifier, SVM_{Char} with performance 81.7%, in identifying targeted offensive towards individual or group with 5% of improvement.

In summary it is noticeable that the stacking ensemble method that combines the least correlated classifiers in each category (classical ML, DL, and transformer-based models), with a variety of knowledge representation and different learning biases, has achieved the highest macro F1-score among the selected classifiers that performed as an individual classifier. Due to the lower noise included in aggregated results of multiple models in comparison with the results of single models, the stacked ensemble classifier has more stability and robustness in its predictions in the identification of offensive language task.

5.2.6 Conclusion

Automatic detection of offensive language and hate speech on social media for low-resource languages, beyond English, is a rising area of concern among academic researchers with regard to a lack of labeled corpora in such low-resource languages. In this study, to the best of our knowledge, we addressed the problem of offensive language detection in Persian as a low-resource language for the first time. We collected a Persian corpus in size of $520K$ from Twitter using both random and lexicon-based sampling techniques and selected $6k$ samples out of it to be annotated with three volunteers who were native Persian speakers. The

corpus was annotated through an existing three-level annotation schema named offensive vs non-offensive, targeted vs untargeted offensive content, and offensive language towards individual, group, or others. Afterwards, we conducted several experiments for offensive language detection in Persian language and evaluated the performance of a diverse set of classical ML, DL, and transformer-based neural network models individually. Furthermore, we built an ensemble stacking model to increase the performance of the classification task by selecting the least correlated single classifiers with different skills on the problem of offensive language detection. The results signify that among single models, the SVM model trained on character or word n-grams followed by pre-trained monolingual model ParsBERT perform the best in identification of offensive vs non-offensive content, targeted vs untargeted offensive content, and targeted offensive content towards individual or groups in almost all cases. Furthermore, using an ensemble stacking model results in increasing the F1-score of the classification task over single classifiers. We believe that considering linguistic-based characteristics of offensive language in Persian as a low-resource language in future researches, would give more precise results in detecting such content on social media.

In the next section, we will use this dataset together with some datasets from other languages, annotated following the same annotation schema, to address the problem of cross-lingual offensive language detection in social media.

5.3 Cross-Lingual Hate Speech Detection using Meta Learning

5.3.1 Introduction

There are two crucial challenges in the automatic detection of hate speech and offensive language, which have made this problem far from being solved at scale. The first one is the multilinguality of social media where these platforms foster their users to interact in their primary languages. Hence, it is essential to have automatic detection tools to protect users with different languages, other than English, against hateful and abusive content. The second one is the lack of sufficient annotated data containing hatred, offense, and abuse for low-resource languages, because collecting and labeling data is a labor- and time-consuming work. In addition, the complex, subjective, and implicit nature of hate speech makes the annotation process more difficult.

Regarding the aforementioned challenges, we investigate the problem of the limited availability of labeled training datasets in low-resource languages for hate speech detection by proposing a few-shot cross-lingual approach based on meta learning. Meta learning is an effective solution proposed for few-shot learning problems, in which we have a few

labeled data for a target task, and it has shown a great performance in different computer vision tasks, such as classifying new image classes with a few available instances of that class [7, 166]. Recently, meta learning has raised attentions regarding few-shot learning problems in NLP tasks as well, where a diverse tasks with different numbers of labels across tasks were studied [167, 168]. However, to the best of our knowledge, this is the first attempt to investigate the feasibility of meta learning in cross-lingual hate speech detection in order to tackle the problem of low availability of labeled data. Here, we study two popular tasks hate speech and offensive language detection, separately, and try to transfer knowledge from resource-rich languages to a low-resource target language by leveraging a meta-learning approach derived from two optimization-based and metric-based methods; Model-Agnostic Meta-Learning (MAML) [7] and Proto-MAML [169].

The primary contributions of this study are:

- It evaluates the feasibility of a meta learning approach in few-shot cross-lingual hate speech detection and demonstrates its effectiveness on different languages with a low-resource setting. Simple but effective modifications are applied on two existing meta learning methods (MAML and Proto-MAML) to accomplish this goal.
- The first large-scale analysis of few-shot cross-lingual hate speech and offensive language detection is realized by assessing the performance of meta learning-based models over transfer learning models (e.g., XLM-RoBERTa) on two diverse collections of different publicly available corpora comprising 15 datasets across 8 languages for hate speech and 6 datasets across 6 languages for offensive language.
- An evaluation using a few-shot setting in which only k samples per class are available from a target language is performed. The experiments demonstrate the superiority of the meta learning approach to generalize quickly to a new language in our few-shot classification tasks in comparison to the transfer learning-based baselines.

5.3.2 Related work

In this section we mainly focus on the studies have been done for multilingual and cross-lingual hate speech detection and few-shot classification tasks.

Multilingual and Cross-Lingual Hate Speech Detection The multilingual nature of social media has underscored the importance of hate speech detection in multilingual settings. Several studies have investigated the multilingual classification of hate speech and offensive language using multilingual, cross-lingual, or joint-learning approaches. We summarize the works in multilingual and cross-lingual settings separately below.

1. Multilingual Building multilingual classifiers to automatically detect hate speech is a very recent topic that would be a notable step forward in this area. Ranasinghe et al. [170]

employed a cross-lingual contextual word embeddings model, XLM-RoBERTa, to transfer knowledge from a rich-resourced language, English, to a lower-resource language (i.e., Bengali, Hindi, or Spanish) to predict offensive content in less-resourced languages. Corazza et al. [171] proposed a robust recurrent neural architecture to identify hate speech in different languages (i.e., English, German, and Italian), and also evaluated the effect of different type of embeddings, additional features (word-level, tweet-level, or emotion-based), and hashtag and emoji normalization in the architecture' s performance. Vashistha et al. [172] proposed a hierarchical deep neural network for the identification of hate speech in English, Hindi, and Hindi code-mix language to investigate the effect of a combination of CNN filters or pre-trained BERT embedding into a biLSTM model. Ibrohim et al. [173] investigated the effect of the machine translation approach in multilingual hate speech detection in Hindi, English, and Indonesian, by comparing classifiers trained with/without translating samples. Ousidhoum in [174] presented the first multilingual multi-aspect hate speech analysis dataset in English, French, and Arabic tweets and evaluated several multilingual multi-task learning approaches for the identification of hate in a multilingual setting.

Multilingual Offensive Language Identification in Social Media (OffensEval-2020) [10] is a pioneering effort to analyze multilingual offensive language in social media by providing multilingual datasets in five languages: Arabic, Danish, English, Greek, and Turkish. Using the English dataset annotated with a three-level annotation scheme to identify offense content, the target audience and the type of offense, participants contributed in this task for a variety of traditional machine learning and deep neural network models. For the languages other than English, data was annotated in one level as either offensive or non-offensive content. More than half of the contributions associated pre-trained transformer-based models: BERT, mBERT, RoBERTa, XLM-RoBERTa, ALBERT, etc. with fine-tuning and data-augmentation strategies to tackle the problem of offensive language detection. Wang et al. [175] proposed a multi-lingual method leveraging the transformer-based pre-trained model XLM-R and ERNIE to predict offensive language and its target and type. Wiedemann et al. [176] performed an exhaustive experimental evaluation using different transformer models such as BERT-base and BERT-large, RoBERTa-base and RoBERTa-large, XLM-RoBERTa, and different version of the ALBERT model to fine-tune the models on offensive English language data and found that using an ensemble combining different ALBERT models outperforms other models.

2. Cross-Lingual The cross-lingual setting in which there are few or non-existent training data sets in the target language is a relatively new concept in the hate speech detection domain. Some recent works have discussed the use of cross-lingual models, along with few-shot or zero-shot learning methods for addressing the problem of hate speech identification across different languages. Stappen et al. [177] proposed an architecture for cross-lingual

zero-shot and few-shot hate speech detection from English to Spanish and vice versa. Their system used a frozen transformer language model, BERT or XLM, to extract the contextual representation of input samples without fine-tuning the models. Their next step utilized an attention-based classification block, Attention-Maximum-Average Pooling (AXEL), as a trainable layer to condense hate speech specific representations from general text representations of BERT or XLM. Aluru et al. [178] analyzed hate speech in a multilingual setting by considering 9 languages from 16 publicly-available hate-speech datasets on Facebook and Twitter. They considered datasets of $n-1$ languages as training and an nth language as the target language (test) to train models based on multilingual embedding models LASER and BERT, using an incremental approach to include target language samples in the training process. Pamungkas et al. [179] employed a machine translation mechanism and proposed two joint-learning architectures based on a multilingual pre-trained model called MUSE (Multilingual Unsupervised and Supervised Embeddings) with an LSTM network and a multilingual pre-trained BERT model to identify hate content among 11 publicly available datasets in 7 different languages. To configure a zero-shot setting, these researchers considered English as the training set and other languages as the test sets. Although this model has yielded a cross-domain robust system, there is a limitation attributed to potential excessive data noise which is produced during the translation and is propagated to downward learning modules. Therefore, in this paper, we do not use translation mechanism in our few-shot setting.

Few-Shot Learning Establishing ways to classify inputs based on only a limited number of samples, known as few-shot learning, has attracted much attention in the research community. One of the most popular solutions for few-shot learning is meta learning, or learning-to-learn, mainly used in the computer vision area [7, 166]. Meta learning has also become popular recently for few-shot learning problems in NLP. Gu et al. [167] introduced a MAML-based meta learning method for low-resource neural machine translation by exploiting large samples of high-resource languages pairs to learn how to adapt to target languages.

Regarding multiple-tasks and monolingual settings, Dou et al. [180] explored multiple MAML-based approaches for low-resource Natural Language Understanding (NLU) tasks on the General Language Understanding Evaluation (GLUE) benchmark, but only for English. Bansal et al. [168] introduced a new MAML-based meta learning model to perform few-shot learning across 17 NLP tasks with different numbers of classes. To learn the interactions between tasks and languages in a meta-learning setting, Nooralahzadeh et al. [181] studied a cross-lingual meta-learning method based on MAML for few-shot and zero-shot learning in Natural Language Inference and Question Answering tasks by pre-training on a high-resource language, English, meta-learning using low-resource languages, auxiliary languages,

and zero-shot or few-shot learning on the target languages. Meanwhile, Tarunesh et al. [182] proposed a meta-learning model to more effectively share parameters across multi tasks and languages by experimenting on five different tasks and six different languages from the XTREME multilingual benchmark dataset.

Based on our literature review, the potential of applying meta learning algorithms to address the problem of few-shot learning in cross-lingual hate speech and offensive language detection tasks has not yet been thoroughly explored. Furthermore, no study has been devoted to investigating cross-lingual hate speech and offensive language detection as two separate tasks with large-scale datasets. Our approach could thus be the first step towards creating a benchmark dataset in hate speech detection similar to other NLP tasks (e.g., GLUE).

5.3.3 Methodology

In this section, we introduce the terminology and definitions related to few-shot learning and meta learning and describe the adaptation of few-shot learning concepts and meta learning approaches to our cross-lingual problem.

Deep neural networks' requirement of large amounts of training data to achieve promising results makes these models inefficient when there is a lack of training data. Meanwhile, hate speech and offensive language are a common phenomenon in social media that does not respect language barriers, so that the lack of sufficient labeled data in some languages, mainly low-resource ones, rendering automatic detection algorithms impractical. Meta learning is thus a potential answer to this training data lacunae.

In this setting, we have a dataset including labeled samples in different languages. We formulate our cross-lingual problem for each target language as an N-way K-shot classification, given:

1. A support set composed of K labeled samples per each N classes for a target language; and

2. A query set composed of Q unlabeled samples of a target language.

Where we aim to classify the Q unlabeled samples of target language into the N classes given the $N \times K$ labeled samples in the support set during the training. Given the insufficient training data that we have in each target language (K samples per each of N classes), we characterize our few-shot learning problem as a meta learning problem in which training on other languages from similar problem helps to achieve better results in a target language. Since we consider our problem as a binary classification (hate/non-hate or offensive/non-offensive), the number of classes N is fixed at two. Our assumption here is that all the K samples in each target language are new.

5.3.3.1 Meta Learning

Meta learning, or learning to learn, is a general paradigm for few-shot learning that learns to quickly adapt to new tasks. Given a classification problem, classical learning algorithms learn how to classify from the training data, and evaluate the performance of a task using test data. However, a meta learning algorithm learns to learn on a diverse set of training tasks and then evaluate new tasks at test time [183].

We consider a model f parameterized by θ to map each training sample with input vector x to output label y; f is often referred to as the *base-learner*. In a meta learning scenario, the model is trained to learn to adapt to a large number of tasks. Therefore, we assume a set of M related tasks in our formulation as $\mathcal{T} = \{\tau_1, \tau_2, ..., \tau_M\}$ with a distribution over tasks $\tau_i \sim p(\mathcal{T})$, where each task potentially has a large amount of training data $\mathcal{D}_i \in \mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_M\}$, containing feature vectors and ground truth labels $\mathcal{D}_i = \{(\mathbf{x}_i, y_i)\}$. Each $\mathcal{D}_i$ is divided into a training set $\mathcal{D}_i^{train}$ (or support set) to adjust the model parameters to the specific task and a test set $\mathcal{D}_i^{test}$ (or query set) to evaluate the performance, denoted as $\mathcal{D}_i = \langle \mathcal{D}_i^{train}, \mathcal{D}_i^{test} \rangle$. In each meta-training step (i.e., an *episode*) a task τ_i is sampled from $p(\mathcal{T})$. Then, considering task τ_i as an N-way K-shot task, the model f is trained with K samples (per N classes) from $\mathcal{D}_i^{train}$ using feedback from the corresponding loss function $\mathcal{L}_i$ from τ_i and evaluated on $\mathcal{D}_i^{test}$ to compute a loss with respect to the model' s parameter initialization. The loss on $\mathcal{D}_i^{test}$ is used to adjust the model parameters. The validation error of the sampled tasks τ_i serves as the training error of the meta learning process in which updating the parameters of the base-learner (f_θ) continues by performing the described episodic training process until some stop criterion is reached. Finally, to generalize the model on a new task $\tau_{M+1} \sim p(\mathcal{T})$, the model uses its learning procedure to adapt to the task τ_{M+1} with only K samples per class of its train set. An overview of our cross-lingual meta learning-based framework is depicted in Figure 5.5 where each task is mapped to a language and a multilingual pre-trained language model is used as the base-learner f. Using a few-shot learning fashion, we use a divers set of tasks in different languages to train a model in meta-training step, and then in meta-testing step the model is further fine-tuned in only k labeled samples from unseen target language.

There are three different approaches for performing meta learning: metric-based [184, 185], model-based [186], and optimization-based [7,187]. In this study, we propose to use an optimization-based meta learning algorithm, Model-Agnostic Meta-Learning (MAML), for our few-shot classification task due to its superior performance at several computer vision tasks [7]. In addition, an adaptation of the MAML method, Proto-MAML [169] is also investigated. In the following sections, we introduce the respective characteristics of these algorithms.

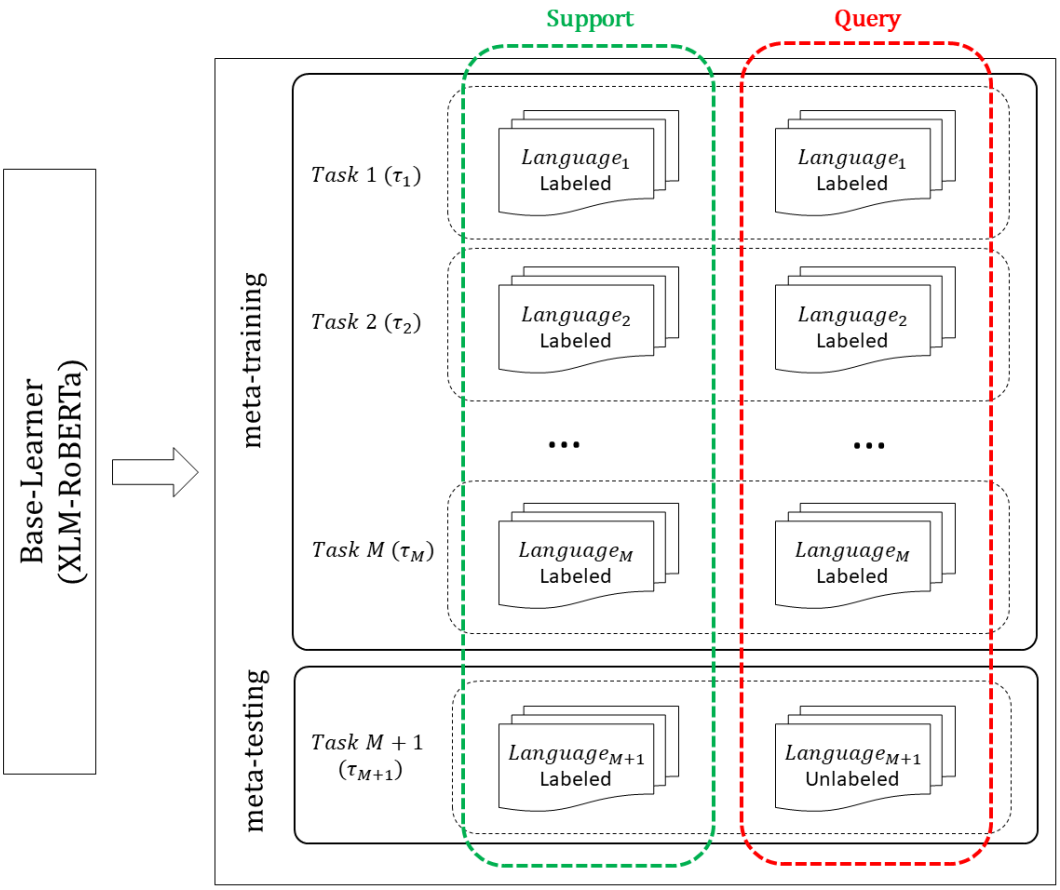


Figure 5.5 – An overview of the cross-lingual meta learning-based framework for few-shot hate speech classification task.

5.3.3.2 Model-Agnostic Meta-Learning

The idea of MAML is to perform meta learning by finding a good initialization of parameters through multiple tasks and then quickly adapting to new tasks with relatively few training samples [7]. Considering a model represented by a parametrized function f_θ with parameters θ, in general, for a single training dataset for one task, neural network parameters are randomly initialized and optimized via gradient descent; however, MAML extends the gradient descent by optimizing parameters θ to yield good performance on a set of related tasks $\mathcal{T} = \{\tau_1, \tau_2, ..., \tau_M\}$.

Given a sampled task τ_i from the distribution $p(\mathcal{T})$, the parameters θ of model f for τ_i are updated to $\theta_i^{'}$ using one or a few gradient descent steps on the $\mathcal{D}_i^{train}$ of task τ_i, as follows:

$$\theta_i^{'} = \theta - \alpha \nabla_\theta \mathcal{L}_{\tau_i} \left(f_\theta \right) \tag{5.1}$$

where α is the step size (learning rate), f_θ is the learned model, and $\mathcal{L}_{\tau_i}$ is the loss on the specific test set $\mathcal{D}_i^{test}$ of task τ_i. The model parameters θ are trained to optimize the performance of the base-learner $f_{\theta_i^{'}}$ on the unseen test examples $\mathcal{D}_i^{test}$ in order to generalize on the specific task τ_i. This step is known as inner-loop optimization. Considering a distribution of tasks $p\left(\mathcal{T}\right)$ the meta learning objective is:

$$min \sum_{\tau_i \sim p(\mathcal{T})} \mathcal{L}_{\tau_i}(f_{\theta_i^{'}}) = \sum_{\tau_i \sim p(\mathcal{T})} \mathcal{L}_{\tau_i}(f_{\theta - \alpha \nabla_\theta \mathcal{L}_{\tau_i}(f_\theta)}) \tag{5.2}$$

Finally, MAML performs meta-optimization, known as outer-loop optimization, across tasks via a stochastic gradient descent (SGD) as follows [7]:

$$\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\tau_i \sim p(\mathcal{T})} \mathcal{L}_{\tau_i}(f_{\theta_i^{'}}) \tag{5.3}$$

where β is the meta step size (learning rate) and the α and β may be fixed as hyper-parameters or be meta-learned.

Although MAML is an elegant and very powerful method that has produced state-of-the-art-results in different settings for computer vision tasks [7], it suffers from some drawbacks such as instability during training, limitations on the model generalizability, high computational requirements in both training and inference times, and being costly in terms of hyperparameter tuning. To address these disadvantages, Antoniou et al. [188] proposed various modifications to MAML that stabilize the system as well as improve the generalization performance, convergence speed and computational efficiency. Following [188], we adapt some modifications to our MAML-base few-shot learning model as follows:

Regarding Equation 5.3, optimizations through gradient update steps in MAML require computing second-order derivatives, which is very expensive. One possible solution is to compute only the first-order approximation of the gradient derivatives to speed up the process, however, this can have a negative impact on the final generalization error [188]. Therefore, we use a derivative-order annealing approach in which in the early steps of training the first-order gradients are computed to speed up the training process, and then in the later training steps the second-order gradients are computed to improve the generalization performance.

Regarding Equation 5.2, the learning rate α is shared across all update steps and all parameters, which results in a high computational cost for finding the correct learning rate for a specific task. Instead, we use an initial learning rate per layer and per step to be jointly learned during the meta-learning steps of MAML. Furthermore, MAML uses a fixed step

size β to optimize its meta-objective in Equation 5.3 with an Adam optimizer, which results in both generalization performance and computational costs issues. We anneal this learning rate on the optimizer by utilizing a cosine annealing function proposed by Loshchilov [189], to achieve higher generalization performance. Although we make some modifications to the original MAML method, for simplicity we use MAML to refer to this model in this study.

5.3.3.3 Proto-MAML

Prototypical Network algorithm proposed by Snell et al. [185] is one of the more successful metric-based approaches in meta learning and has yielded substantial improvements in the few-shot learning problem. This approach hypothesizes the existence of an embedding (a prototypical representation) in which all the samples belonging to a specific class cluster around a single prototype representation for that class. Then, a new sample in few-shot learning is classified based on its distance with the prototypical representation of each class. Therefore, this metric-based algorithm requires an embedding function f_θ to extract the embeddings of all samples, and a distance function d to compute the distance between new samples and the prototypical representation of each class. Given an embedding function f_θ and a few-shot classification with support set S and query set Q, the embeddings of all samples in S are encoded by f_θ, and then a prototype $\mathbf{c}_k$ is computed for each class k in S by taking the mean embeddings of samples of the respective class as follows:

$$\mathbf{c}_k = \frac{1}{|S_k|} \sum_{(\mathbf{x}_i, y_i) \in S_k} f_\theta(\mathbf{x}_i) \tag{5.4}$$

where $\mathbf{c}_k$ is the prototype of class k. Given the prototypes of the classes in S, each unlabeled sample in Q is encoded by f_θ and is then classified by:

$$p(y = k|\mathbf{x}) = \frac{exp\left(-d\left(f_\theta(\mathbf{x}), \mathbf{c}_k\right)\right)}{\sum_{k'} exp\left(-d\left(f_\theta(\mathbf{x}), \mathbf{c}_{k'}\right)\right)} \tag{5.5}$$

where d is a distance function, mainly the squared Euclidean distance. To obtain the probability distribution over classes, a Softmax function with a negative log-likelihood loss function is applied on the distance vectors and then the sample is assigned to the class with the highest probability value.

Triantafillou et al. [169] utilized the simple inductive bias of the Prototypical Network algorithms and the simple and flexible adaptation mechanism of MAML to introduce a new meta-learning algorithm, called Proto-MAML. Using the Euclidean distance function in Prototypical Networks makes them a linear model [185] where the equivalent linear layer has weights $\mathbf{W}_k$ and biases b_k corresponding to class k which are computed as follows (regarding Equation 5.4):

$$\mathbf{W}_k \coloneqq 2\mathbf{c}_k \quad \text{and} \quad b_k \coloneqq -\left\|\mathbf{c}_k\right\|^2 \tag{5.6}$$

Therefore, Proto-MAML adapts MAML in such a way that these weights and biases can be employed in the task-specific linear layer of each episode in the MAML instead of using random initializations. This simple modification yields significant improvements in the optimization process of MAML [169]. We adapt this strategy in MAML and use Proto-MAML as a meta learning algorithm in our cross-lingual few-shot classification problem. We apply all the modifications which are used in MAML in Proto-MAML as well. Although we make some modifications to the original Proto-MAML method, we use the same name to refer to this model for simplicity.

5.3.3.4 Base-Learner Model

Since the optimization-based meta learning algorithms used in this study are model-agnostic, they are compatible with any base-learner model that learns through gradient descent. Here, we chose the multilingual pre-trained language model XLM-RoBERTa [63] as the base-learner for hate speech and offensive language classification tasks. The base-learner extracts the last hidden-state layer of the first token of the sequence (the classification token) in size 768 and processes it by a linear layer and a *tanh* activation function.

5.3.4 Dataset Description

Given the varying definition of hate speech and offensive language content in publicly available datasets, we prevent to combine datasets with hatred and offense samples. Hence, we consider two separate tasks, *hate speech detection* and *offensive language detection*, with different datasets in different languages. The hateful datasets consist of insults targeted toward a group based on some characteristics such as sexual orientation, religion, misogyny, nationality, gender, ethnicity, etc., however, offensive datasets contain any form of non-acceptable language or a targeted offense including insults, threats, and posts containing profane language or swear words [10].

5.3.4.1 Hate Speech Data

We use 15 publicly available sources in 8 languages provided by research community. Most of the datasets are selected according to the *hatespeechdata*[9] web page that catalogs datasets annotated for hate speech, online abuse, and offensive language. Although different datasets have different classes, in this study, we only select samples including hateful or normal content. The details regarding all hateful datasets are included in appendix A.1.

[9] `https://hatespeechdata.com`

5.3.4.2 Offensive Language Data

We use the multilingual offensive language dataset provided in OffensEval-2020, a shared task at SemEval-2020, which focused on multilingual offensive language identification in 5 languages Arabic, Danish, English, Greek, and Turkish. In addition, we use our Persian offensive language dataset introduced in Section 5.2.3. The details regarding all offensive datasets are included in appendix A.2.

The statistics of these datasets are presented in Table 5.7. First column represents the datasets from different languages in Hate Speech and Offensive Language categories. The second and third columns represent the number of normal (non-hateful or non-offensive) and hateful/offensive content as Class 0 and Class 1, respectively. The total number of samples in each language is reported in the final column.

5.3.5 Experiments and Results

This section presents the details of different training models including the meta learning models and different baselines used in this study. In addition, it describes experimental setup and implementation details as well as the results of our experiments.

5.3.5.1 Training Models

MAML and Proto-MAML In our cross-lingual few-shot classification task, we have a set of training, validation, and test tasks which are including samples from different languages (mutually exclusive). To investigate the performance of meta learning approach in cross-lingual hate speech detection, we divide each dataset (hate speech or offensive language) into three meta-datasets: 1) a training set L_{train} comprising of training languages to train MAML; 2) a validation set L_{val} consisting of validation languages to tune MAML hyper-parameters; and 3) a test set L_{test} consisting of test (or target) language to evaluate generalization of the model on an unseen target language. Therefore, using labeled data in L_{train} the model is trained. Then, by using samples from L_{val}, we tune the hyper-parameters and set early stopping condition. As we consider a few-shot setting, we do not rely on a large validation set and use a held-out validation set of a specific language in validation set L_{val}. For evaluating the method on L_{test}, at first, we fine-tune the model using a sample of k-shot training data (k samples per label in target language 's train set) and then test the model on the entire test set of target language. Therefore, target language will be unknown during both training and model selection. All the settings in MAML and Proto-MAML are the same.

Baselines We create two transfer learning baselines (based on XLM-RoBERTa model) to evaluate the ability of these approaches as well as our proposed model for cross-lingual

Table 5.7 – Dataset description for hate speech and offensive language detection tasks. Class 0 and Class 1 represent normal and hate/offensive labels in the datasets, respectively.

Datasets	Class 0	Class 1	Total
Hate Speech			
English	66,205	13,143	79,348
Davidson [3]	4,163	1,430	5,593
Basile [77]	7,530	5,470	13,000
Founta [67]	53,851	4,965	58,816
Ousidhoum [174]	661	1,278	1,939
Arabic	4,565	1,223	5,788
Ousidhoum [174]	915	755	1,670
Mulki [190]	3,650	468	4,118
Spanish	8,294	4,306	12,600
Basile [77]	3,861	2,739	6,600
Pereira [191]	4,433	1,567	6,000
German	4,441	206	4,648
Ross [66]	315	54	369
Mandl [150]	4,126	152	4,279
Indonesian	8,061	5,821	13,882
Ibrohim [192]	7,608	5,561	13,169
Alfina [193]	453	260	713
Italian			
Bosco [149]	2,704	1,296	4,000
French			
Ousidhoum [174]	821	399	1,220
Portuguese			
Fortuna [194]	3,882	1,788	5,670
Offensive Language			
English [11]	9,460	4,640	14,100
Arabic [138]	8,085	1,915	10,000
Danish [195]	3,159	441	3,600
Turkish [139]	28,464	6,847	35,284
Greek [136]	7,376	2,911	10,287
Persian	4,376	1,624	6,000
TOTAL	159,893	46,560	206,453

few-shot hate speech and offensive language detection tasks. The baselines are as follows:

- **XLM-R** Aluru et al. [178] have recently proposed a multilingual BERT-based model for multilingual hate speech detection in which all samples in different languages except a target language l_{tgt} (test language) are used as training data and then the model is further fine-tuned with a portion of training data of l_{tgt} and evaluated in a held-out test set of l_{tgt}. Inspiring this study, we create a baseline for our few-shot cross-lingual model where we use the pre-trained model XLM-R with a two-step fine-tuning method. During the fine-tuning, first, the model is trained on all languages except l_{tgt} and the best model is selected according to the held-out validation set of l_{tgt}. Then the selected model is fine-tuned with only k samples (per class) in l_{tgt}. At the end, the model is evaluated on the test set of l_{tgt}. Samples from different languages in training, validation, and test steps of this model are considered as L_{train}, L_{val}, and L_{test}. We note that according to [178], this model uses target language for both model selection and test step. Therefore, the target language will be unknown only during training phase.

- **Non-episodic** To measure the exact impact of meta learning on the performance of model versus standard supervised learning, we use a non-episodic approach to train a model in which support and query sets of training languages in L_{train} are merged, and by using a mini-batch gradient descent with cross-entropy loss function the model is trained. In the test step, first, the trained model is fine-tuned on k-shot training data of L_{test}, and then is evaluated on test set of L_{test}. The target language will be unknown during both training and model selection.

5.3.5.2 Training Setup and Implementation

Training Setup We consider hate speech and offensive language detection as two separate tasks in which a binary classification is trained based on transfer learning or meta learning approaches. To create and initialize each model, we use the configuration, tokenizer, and pre-trained weights of the XLM-R (xlm-roberta-base) model from publicly available Transformers[10] library for Pytorch (pytorch-transformers). Then, each model will be fine-tuned on the downstream task by adding a classification head on top of the pre-trained XLM-R encoder. As hate speech and offensive language detection are binary classification tasks, we directly modify and fine-tune the classification class of the XLM-R model (*XLMRobertaForSequenceClassifcation*).

For MAML-based meta leaning models, we consider 50 epochs and sample 100 training episodes per epoch to perform meta training. The learning rate of inner loop α (adaptation

[10] `https://github.com/huggingface/transformers`

stage) and learning rate of outer loop β are set initially to 3e-5 and 6e-5, respectively. We use Adam optimizer to update the parameters. The number of update steps in the inner-loop is set to 10. During the first 30 epochs, we calculate the first-order derivatives and in the rest of training process we calculate the second-order derivatives in MAML. We perform evaluation on the samples in L_{val} set with 5 different seeds after each epoch, and to avoid overfitting, we apply early stopping when the validation accuracy failed to decrease for 5 epochs. In the few-shot setting, we chose $k \in \{4, 8, 16\}$ to evaluate how models generalize to new target language with a limited labeled data k per class.

For the XLM-R baseline, the maximum sequence length of the input sentences is set to 256 and in case the input length is shorter or longer, it will be padded with zero values or truncated to the maximum length, respectively. The model is fine-tuned with a batch size of 16 for 5 epochs. An Adam optimizer with a learning rate of 2e-5 is used to minimize the Cross-Entropy loss function. For non-episodic baseline, the model is trained for 5 epochs on L_{train} and is evaluated after each epoch on L_{val} set.

Implementation As the implementation and execution environment, we use Lab-IA[11] platform provided by The French National Centre for Scientific Research[12] (CNRS) with a NVIDIA Tesla V100 GPU with 32 GiB of RAM (NVLink).

5.3.5.3 Results and discussions

In this section, we evaluate the training models on hate speech and offensive language detection tasks with different languages.

Hate speech detection In this task, we combine all datasets in each language as reported in Table 5.7. Due to the lack of a held-out benchmark test set for each dataset, after combining all datasets in each language, we select 20% of samples in each language as test set by performing a stratified sampling. To have a verity of tasks during the meta-training step, we leverage different languages with different hateful content where all languages except two are selected as training set. For example, to evaluate meta-learning models on Arabic as a target language with k labeled samples per class, we consider one language for validation and the rest of languages for training, where $L_{train} = \{English, French, German, Indonesian, Spanish, Portuguese\}$, $L_{val} = \{Italian\}$, and $L_{test} = \{Arabic\}$. According to the literature in low-resource NLP classification tasks [196], it can be unreasonable to assume that we have a large validation set; thus we consider only one language in L_{val} set for all experiments. Performing initial experiments leads us to choose Italian as validation language. Therefore, in all experiments we set $L_{val} = \{Italian\}$ except when Italian is used as a target language at which we set $L_{val} = \{Spanish\}$. The ratio

[11] `https://doc.lab-ia.fr/`
[12] `http://www.cnrs.fr/`

of validation samples is set to 20% original dataset. As English has been frequently used in hate speech detection tasks with a large labeled data, we consider it as a high-resource language and fix it in L_{train} during all experiments.

Offensive language detection In this task, there exists one dataset per language that has a specific held-out test set, provided by OffensEval 2020, and we use this test set for evaluation. Only for Persian, which is provided by us, we sample a ratio of 20% of the data as test set. Similar to the hate speech dataset, in each experiment we consider all languages except two as training set, where English is always included. We set $L_{val} = \{Turkish\}$ except when Turkish is used as a target language at which we set $L_{val} = \{Arabic\}$. The ratio of validation samples is set to 10% original dataset.

Towards a faithful evaluation amongst all models, we keep the same train, validation, and test samples in all experiments. In our few-shot setting, we evaluate the models on $k \in \{4, 8, 16\}$ and due to the sensitivity of models to the k samples chosen from the target language in test set, we perform each experiment based on 10 testing episodes (for each k) and report the average performance in terms of macro F1-measure over 5 different random seeds. For the XLM-R and non-episodic baselines, we select 10 different random sets in size k and report the average performance.

Tables 5.8 and 5.9 present the results for k-shot hate speech and offensive language detection datasets, respectively. The performance of each model for each k-shot setting is displayed in terms of macro-averaged F1-measure along with the standard deviations. Each column corresponds to an unseen target language and the last column shows the average performance of each model on all target languages, for the sake of comparison. The values in bold indicate the best performing model in each k-shot setting.

Generally, the results clearly demonstrate that meta learning-based models, MAML and Proto-MAML, outperform other models in most cases, and Proto-MAML achieves the best performance across two datasets in the majority of settings. Regarding the last columns in both tables, when comparing against MAML, Proto-MAML improves notably by 6.7%, 20%, and 35% on average in 4-,8-, and 16-shot classification for hate speech dataset and by 5.2%, 8.3%, and 24.3% on average in 4-,8-, and 16-shot classification for offensive language dataset, respectively. Therefore, this specifies the high ability of Proto-MAML in generalizing to the new language given a few samples.

Considering two baselines XLM-R and non-episodic, we observe that in most settings, XLM-R achieves better results. Since the non-episodic baseline trains in a non-episodic fashion and concatenates the samples of all training languages during training, it performs the training process the same as XLM-R baseline. However, the main difference between these two baselines is in the choice of validation language to select the best model; where XLM-R uses the target language for validation, whereas non-episodic uses two different

Table 5.8 – Results of k-shot classification on the unseen target languages of hate speech dataset in terms of macro F1-measure with standard deviation. The values in bold indicate the best performing model in each k-shot setting. The last column corresponds to the average F1-measure across all target languages.

Models	**k-shot**	**Target Languages**							
		ar	**de**	**es**	**fr**	**id**	**it**	**pt**	**avg**
XLM-R [178]	4	42.32 ±0.91	37.90 ±1.41	46.06 ±1.37	46.23 ±0.62	45.74 ±1.65	37.22 ±2.11	41.67 ±0.73	42.44
	8	38.78 ±2.31	**46.13** ±0.81	39.77 ±1.78	**46.44** ±1.78	46.98 ±2.78	39.02 ±0.58	43.50±1.08	42.94
	16	43.01 ±1.32	50.23 ±2.01	45.64 ±0.82	52.32 ±2.24	49.86 ±0.87	45.22 ±1.41	51.03 ±2.82	48.18
Non-episodic	4	41.07 ±2.91	36.29 ±0.36	45.33 ±2.41	45.77 ±1.61	44.41 ±2.47	37.05 ±0.55	40.31±0.89	41.46
	8	35.84 ±0.59	42.22 ±0.90	38.49 ±2.51	45.24 ±2.18	34.41 ±0.10	37.89 ±0.78	**45.30** ±1.83	39.91
	16	39.00 ±0.46	37.04 ±1.23	45.29 ±0.69	35.55 ±1.25	41.67 ±0.47	34.38 ±0.62	50.20 ±0.57	40.44
MAML	4	**45.62** ±1.90	40.06 ±0.84	**49.97** ±1.90	44.93 ±2.01	45.15 ±0.93	36.96 ±0.55	**47.97** ±1.24	44.38
	8	36.99 ±0.75	45.77 ±2.35	34.30 ±1.44	44.16 ±2.16	36.97 ±0.33	35.85 ±1.82	31.27 ±0.57	37.90
	16	**51.48** ±1.76	39.87 ±1.40	42.44 ±0.81	40.87 ±0.78	37.53 ±0.41	35.90 ±0.39	35.39 ±0.55	40.49
Proto-MAML	4	44.31 ±1.80	**45.23** ±2.75	45.47 ±3.10	**48.16** ±3.51	**60.99** ±2.41	**45.34** ±2.52	43.62 ±3.61	**47.58**
	8	**46.85** ±5.23	44.48 ±3.71	**44.92** ±2.21	42.93 ±2.40	**60.51** ±0.43	**49.93** ±1.26	44.43 ±3.08	**47.72**
	16	44.42 ± 2.91	**61.94** ±0.74	**61.24** ±2.91	**67.36** ±1.54	**64.41** ±1.14	**70.64** ±0.06	**68.80** ±1.08	**62.68**

languages in the validation and test steps. Although the results show that using the same language for the best model selection (validation step) and test step yields better performance, it is not aligned with our assumption in cross-lingual few-shot setting in which test language remains unseen during training and validation.

An interesting observation is that although all models are initialized and fine-tuned with pre-trained language model XLM-RoBERTa, the cross-lingual knowledge in hate and offensive contexts is not transferred by baselines across languages well. Whereas, Proto-MAML leverages the cross-lingual class prototypes along with initial parameters performing equally well across languages in meta-training step to benefit from XLM-RoBERTa embeddings.

Regarding Table 5.8, we perceive that hateful content in different languages are more transferable through meta learning-based models (MAML and Proto-MAML) in comparison with transfer learning-based models (XLM-R and non-episodic); where, Proto-MAML and MAML achieve the best performances with different k values, except when German, French, and Portuguese are target languages with $k = 8$. Results indicate that increasing the number of labeled data per class (k) does not necessarily lead to better performance incrementally, however $k = 16$ is a stable number for Proto-MAML to perform well on different languages. An interesting observation is that although we have a heterogeneous set of languages in training, where Arabic and Indonesian are from different language families

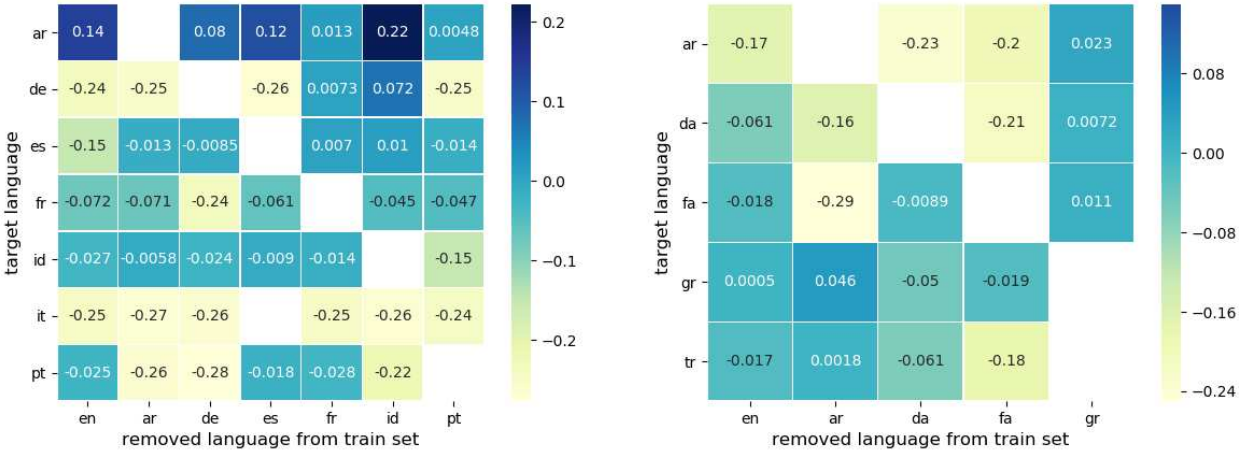
Table 5.9 – Results of k-shot classification on the unseen target languages of offensive language dataset in terms of macro F1-measure with standard deviation. The values in bold indicate the best performing model in each k-shot setting. The last column corresponds to the average F1-measure across all target languages.

Models	**k-shot**	**Target Languages**					
		ar	**da**	**fa**	**gr**	**tr**	**avg**
XLM-R [178]	4	33.76 ±0.92	40.26 ±2.55	**43.47** ±1.16	32.17 ±0.56	38.76 ±2.32	37.68
	8	37.60 ±2.42	39.60 ±3.05	45.04 ±2.38	38.87 ±0.68	**46.62** ±1.30	41.54
	16	40.32 ±1.62	42.09 ±2.15	45.76 ±1.08	39.26 ±0.78	46.95 ±1.18	42.87
Non-episodic	4	30.67 ±0.93	35.27 ±1.84	30.08 ±1.34	31.36 ±1.12	36.29 ±0.83	32.73
	8	47.69 ±1.13	32.02 ±1.14	43.60 ±1.26	34.32 ±1.24	39.62 ±0.68	39.45
	16	48.67 ±1.02	40.83 ±2.12	44.36 ±0.72	31.72 ±3.05	49.16 ±1.21	42.94
MAML	4	**51.12** ±1.11	46.64 ±1.66	30.90 ±0.58	**41.14** ±1.75	**42.82** ±4.22	42.52
	8	54.04 ±0.90	45.51 ±1.51	**54.68** ±1.81	51.52 ±1.98	40.38 ±0.44	49.22
	16	48.89 ±1.12	47.81 ±2.08	46.35 ±0.49	41.21 ±2.75	56.55 ±0.59	48.16
Proto-MAML	4	41.05 ±1.32	**57.84** ±2.60	43.21 ±0.91	40.50 ±1.21	41.70 ±2.01	**44.86**
	8	**58.65** ±2.06	**57.73** ±3.12	45.98 ±2.24	**59.70** ±2.50	46.25 ±3.52	**53.66**
	16	**64.16** ±3.14	**59.80** ±2.71	**72.92** ±4.77	**60.55** ±2.25	**60.64** ±2.36	**63.61**

with low typological commonalities with other languages, meta learning-based models can generalize to these languages with better performance quickly; which is very practical in real applications.

Regarding Table 5.9, offensive content is well generalized across languages where Proto-MAML is the best-performed model in all target languages with $k = 16$. Hate speech and offensive language are subjective and contextual-based phenomenon and the substantial improvements for languages such as Arabic, Persian, and Turkish indicate that meta learning is most beneficial when we have tasks with heterogeneous languages. More precisely, in hate speech and offensive language we are facing with a domain drift problem in which some context cannot be captured across different languages easily, such as cultural differences. However, our results show that meta learning can alleviate this problem.

Ablation Study To analyze the contributions of different training languages on performance of the meta-training process in Proto-MAML model, as the best-performing model, we conduct an ablation study. To that end, we repeat the experiments with training Proto-MAML model with $k = 16$ while removing each language in training set one by one and calculating the performance differences compared with original results (which is reported in Tables 5.8 and 5.9 for Proto-MAML/$k = 16$), in terms of F1-measure. Figure 5.6 shows the relative change in performance when each training language is held out from original train set of hate speech and offensive language detection datasets; where rows indicate target



(a) Hate speech detection (b) Offensive language detection

Figure 5.6 – Differences in the performance of Proto-MAML after removing each training language from the train set, in terms of F1-measure. Rows correspond to target languages and columns correspond to the removed language from the original train set. Each cell reports performance differences between training on the original train set and the train set without a specific training language.

languages and each column corresponds to an held-out training language. Positive values show an improvement in performance after removing a specific training language while the negative values indicate a reduction in the performance. It is noted that, in hate speech, choosing *es* as validation language when the target language is *it* causes an empty cell regarding the case in which the target language and the removed one are *it* and *es*, respectively. Furthermore, in offensive language, choosing *gr* as validation language when the target language is *tr* causes an empty cell regarding the case in which the target language and the removed one are *tr* and *gr*, respectively.

The results specify the effectiveness of each language in the generalization of the model where removing each of them results in a reduction of performance, in major cases (where the performance differences are negative). For hate speech detection task, as shown in Figure 5.6a, removing the training languages $\{en, ar, de, es, pt\}$ gives rise to a performance reduction except when the target language is *ar*, which indicates a positive contribution of each language in the model' s ability to generalize to the target language with a few labeled samples. A small improvement is observed in the performance of the model for target languages *de* and *es*, when we remove *fr* or *id* from training set. In addition, surprisingly for target language *ar*, we observe that removing each language during meta-training leads to the performance improvement where removing *id* has the largest impact. We hypothesize

that the large distance between *ar* and other languages is the cause of this observation; where *ar* belongs to Afro-Asiatic, *id* belongs to Austronesian, and the rest of languages belong to Indo-European language families. Therefore, in this situation, the choice of training languages has different implications for an unseen target language, and has a crucial impact in the ability of meta learning model to adapt to the new language. The relationship between the training languages and an unseen target language in terms of typology and distance must be investigated further in the future.

For offensive language detection task, as shown in Figure 5.6b, we observe that removing the training languages $\{en, ar, da, fa\}$ results in a performance reduction in all cases except when *gr* is a target language; at which there exists a small improvement by removing *ar* and *en*. On the other hand, removing *gr* from training set causes an improvement in the performance for all target languages. This indicates that *gr* language has a negative impact across different target languages in meta-training process. However, the diversity of other training languages has positive effect in the performance of the model.

5.3.6 Conclusion

Although pre-trained transformer models have yielded promising results in hate speech detection tasks, they require a large amount of labeled data in a specific language; which is not always feasible for low-resource languages. In this section, we studied the problem of few-shot learning in cross-lingual hate speech and offensive language detection tasks by exploring the feasibility of meta learning approach as a potential solution for the first time, to our knowledge. To that end, we collected a diverse set of publicly available datasets containing hateful and offensive content from different languages to create two benchmark datasets for cross-lingual hate speech and offensive language classification tasks. We employed a meta learning approach based on optimization-based and metric-based methods (MAML and Proto-MAML) to train a model being able to generalize quickly to a new language with a few labeled data (k samples per classes). The experiments demonstrate that meta learning-based models outperform transfer learning-based models in a majority of cases, and Proto-MAML is the best performing model where it can quickly generalize and adapt to new languages with a few labeled data (mainly 16 sample per class yields an effective performance) to identify hateful or offensive content. In addition, MAML also performs strongly, however transfer learning-based baselines notably presents the lowest results. The results indicate that Proto-MAML is the best performing model where it can quickly generalize and adapt to new languages with a few labeled data (mainly 16 sample per class yields an effective performance) to identify hateful or offensive content. In addition, MAML also performs strongly, however non-episodic baseline notably presents the lowest results. Our future work will extend this study to investigate different sampling strategies for training tasks

and see how different languages in training set affect on the performance of meta learning models for an unseen target language. We will also perform a typological analysis to study the relationships between different language families and the performance of meta learning in cross-lingual hate speech detection task.

5.4 Summary and Discussion

To summarize and conclude, this chapter presented a general overview of the third contribution related to low-resource languages in hate speech and offensive language detection into two parts.

In the first part, we studied offensive language in Persian where we provided the first dataset for Persian offensive language. Using two random-based and lexicon-based sampling strategies, we collected a corpus in size 520k from Twitter and sampled 6k tweets out of it to be annotated with three volunteers who were native Persian speakers. We used a three-level annotation schema to labeled data as offensive/non-offensive, targeted/untargeted, and targeted towards individual/group/others. We created a variety of traditional machine learning, deep learning, and monolingual and multilingual pre-trained language models to investigate the performance of different models in Persian offensive language detection. We fine-tuned different monolingual (e.g., ParsBERT and ALBERTPersian) and multilingual (e.g., mBERT and XLM-RoBERTa) pre-trained language models on the labeled data, and the results show the priority of the ParsBERT model in identification of offensive language and its type and target. In addition, to boost the performance of the classification task, we proposed an ensemble stacking model which outperforms single classifiers.

In the second part, we investigated the problem of few-shot learning in cross-lingual hate speech and offensive language detection, where there exists only k labeled data per class, and exploited a cross-lingual approach based on meta learning methods to address the problem. We performed simple but effective modifications in two meta learning methods MAML and Proto-MAML, where we used a combination of first-order and second-order derivatives in computing gradient descent during training, and dynamically learned step size in inner loop and outer loop of meta-training process. Evaluation results on two hate speech and offensive language data, including 21 publicly available datasets from 12 languages, indicate the priority of meta learning-based models in our few-shot cross-lingual classification tasks. Since the application of meta learning approach in few-shot cross-lingual hate speech detection is very new, we believe that this study could institute the first step towards exploiting different meta learning methods in few-shot hate speech classification in the future.

In the next chapter, we move towards the conclusion of this thesis and some future works for the extension of this thesis.

Chapter 6

Conclusion and Future Work

Contents

6.1 Conclusion **146**
6.1.1 Summary and Insights of Contributions 146
6.2 Future Work and Challenges **149**

6.1 Conclusion

An exponential increase in the utilization of social media for generating and propagating different types of content makes these online platforms as a potential place for expressing hatred, offense, and harassment. This thesis makes timely and constructive contributions to social media content analysis in terms of hate speech and offensive language detection and alleviation by proposing different automatic models based on transfer learning approaches. To that end, we first have analyzed user-generated content in social media to understand the type of content and the user intention behind generating that, in terms of toxicity analysis. This gave us a hint to the inevitability of hate speech detection in social media. Therefore, we have studied the problem of hate speech detection in social media using a transfer learning approach in both monolingual (where only English data is considered) and multilingual (where low-resource languages are considered) settings. We have adapted the advanced pre-trained language models on the hate speech detection by proposing new fine-tuning strategies on the downstream task. In addition, we have proposed a bias mitigation mechanism to alleviate the racial bias in our model. At the end, we have addressed the problem of low-resource languages in this domain by providing a Persian offensive language dataset and proposing a meta learning-based model for cross-lingual hate speech detection.

In this final chapter, we recapitulate the proposed methods, summarize our findings, and provide an outlook into the future.

6.1.1 Summary and Insights of Contributions

In this section, we provide the summary of each contribution, as well as the insights gained from each contribution.

- **Social media content analysis:** This contribution aims at providing an insight into the user-generated content in social media focusing on posts and comments of a public news agency page in Facebook. We collected posts and associated comments of BBC News page using Facebook Graph API, and annotated a portion of data based on a well-defined set of clues for identifying related and unrelated comments in terms of their similarity to topic of the actual post. Then, we proposed a framework to measure the similarity of given comments to a post in terms of the content and distinguish the related and unrelated written comments to a post by leveraging a novel feature engineering comprised of a lexical, topical, and semantical set of features. The results indicate that our model yields in average the precision of 86% in identifying related and unrelated comments with an improvement of 9.6% in comparison with previous work. We applied the trained classifier on the whole dataset, and the results show that almost 60% of the written comments are not related to the actual posts' content in

terms of the discussed topics. In addition, we used a publicly available tool to measure the toxicity of comments and results declare the frequent occurrence of different types of hate speech (e.g., thread, insult, and toxicity) within comments that it is worth further investigation. More details about this research work can be found from our published paper [197].

- **BERT-Based transfer learning approach for hate speech detection:** This contribution aims to deal with the limitations of existing algorithms for hate speech detection where we have the lack of a sufficient amount of labeled data. The emergence of BERT model, in 2018, was a major breakthrough which has transformed the NLP landscape and achieved significant results for many NLP tasks. Therefore, for the first time, we adapted BERT to the hate speech detection task to not only improve the performance, but also alleviate the lack of enough labeled data by leveraging the pre-trained knowledge of BERT. To that end, we investigated the ability of BERT at capturing hateful context within social media content by proposing new fine-tuning strategies that employed contextual information embedded in different encoder layers of the BERT. We used two publicly available datasets annotated as racism, sexism, hate, or offensive content on Twitter. The results show that our solution obtains a considerable improvement in performance on these datasets in terms of F1-score in comparison to the existing approaches; where we integrated a CNN layer with BERT. Furthermore, investigating the performance of model regarding different portions of training data indicates a significant improvement in comparison to the baselines that did not rely on transfer learning approach.

- **Racial bias mitigation in hate speech detection algorithm:** This contribution aims to tackle the problem of existing bias in benchmark datasets for hate speech and abusive language, where these data are containing oddities that result in a high preference for trained classifiers to classify some samples to a specific class. We explored two publicly available benchmark datasets in terms of these oddities and our investigation revealed that these oddities were mainly associated with a high correlation between some specific n-grams in training set and a specific negative class. Therefore, we introduced a bias alleviation mechanism to mitigate the effect of bias in training set during the fine-tuning of our pre-trained BERT-based model (proposed in previous contribution) for hate speech detection. Toward that end, we used a regularization method to reweight input samples, thereby decreasing the effects of high correlated training set' s n-grams ($n = 2$) with class labels, and then fine-tuned our pre-trained BERT-based model with the new re-weighted samples. Then, we used a new dataset from Twitter containing AAE-aligned and White-aligned groups, which

indicated tweets written in African American English (AAE) and Standard American English (SAE), respectively, to evaluate the bias alleviation mechanism. The results show the existence of systematic racial bias in trained classifiers, as they tend to assign tweets written in AAE from AAE-aligned group to negative classes such as racism, sexism, hate, and offensive more often than tweets written in SAE from White-aligned group. However, the racial bias in our classifiers reduces significantly after that our bias alleviation mechanism is incorporated. We also analyzed the performance of model after incorporating bias mitigation mechanism on the same datasets that it was trained on and we observed a reduction in the performance which will be part of a future investigation.

- **Offensive language detection in Persian:** This contribution aims to provide the first offensive language dataset in Persian, as a low-resource language, where there has not been any labeled data for this specific task. Therefore, we crawled a corpus of size $520k$ from Twitter using both random and lexicon-based sampling techniques and then we selected $6k$ samples out of it to be annotated with three volunteers who were native Persian speakers. By using a three-level annotation process, we categorized the offense content as well as its type and target. Then, we performed extensive experiments using a number of classical machine learning, deep learning, and transformer-based neural networks including monolingual and multilingual pre-trained language models, to investigate the ability of pre-trained language models in Persian offensive language detection. The results show that pre-trained language model ParsBERT yields the best performance among all pre-trained language models, however, it performed comparable to word and character n-grams based models in some cases. Another interesting result is that monolingual pre-trained language models (ParsBERT and mBERT) outperform multilingual pre-trained language models (mBERT and XLM-R) where XLM-R has the lowest performance in identification of offensive content which indicates the low quality of representation for Persian language in this cross-lingual pre-trained model. Furthermore, we proposed a stacking ensemble model that outperforms the single models by a substantial margin, obtaining 5% respective macro F1-score improvement for the three levels of annotation.

- **Cross-lingual hate speech detection using meta learning:** This contribution aims to study a few-shot cross-lingual hate speech detection task by employing a meta learning approach based on different metric-based and optimization-based methods (MAML and ProtoMAML) to use trained knowledge from different source languages for generalizing to the target language with few labeled data efficiently. To best of our knowledge, we performed the first analysis of few-shot cross-lingual hate speech and

offensive language detection by exploiting meta learning-based models where we have two diverse collections of different publicly available corpora comprising 15 datasets across 8 languages for hate speech and 6 datasets across 6 languages for offensive language. Our experiments show that meta learning-based models are able to quickly generalize to new languages where mainly 16 sample per class yields an effective performance.

6.2 Future Work and Challenges

This section summarizes some perspectives on the future work to extend the work in this thesis.

In the first study, we have done an analysis to gain insight into the user-generated content in social media and the level of toxicity in comments generated by users on Facebook. This study can be expanded into future interests by: i) analyzing comments across different categories such as politicians, celebrities, and companies in Facebook and examining the proposed model to filter the most similar comments into the posts in terms of the topic; and ii) inspecting the correlation between the topic of posts and the level of toxicity in comment section.

In the second study, we have proposed a hate speech detection model based on advanced pre-trained language model BERT with promising results. This study can be expanded to cover other concerns: i) although we have achieved the promising results using different fine-tuning strategies for BERT, we should focus on making our model more robust by evaluating the model on out-of-domain data. Therefore, we need to have a dedicated out-of-domain test set for our hate speech detection task; ii) adding a lexical knowledge from different hate or abusive lexicons can inject more knowledge about hateful content into our BERT-based model and achieve better results; and iii) exploring other pre-trained language models (e.g., ALBERT, GPT, etc.) in hate speech detection task rather than BERT.

In the third study, we have proposed a bias mitigation mechanism using a regularization method for our BERT-based hate speech detection model. This study can be expanded into future interests: i) although our proposed method is able to alleviate racial bias on a cross-domain data, it underperforms on the dataset that it has been trained on, after applying the regularization mechanism during fine-tuning process. It is interesting to see how we can mitigate data-driven and algorithm-driven bias while we preserve the performance of our model on the main training data.

In the forth study, we have addressed the problem of offensive language in low-resource language Persian, where there has not been any labeled data so far. This study can be expanded to future interests: i) in this study we have used a specific definition of offense

content for annotating the data which may not capture other concepts related to hate speech (e.g., sexism, racism, etc.). Therefore, providing new datasets for different concepts of hate and offense will be a constructive contribution in this research direction; and ii) relatively low performance level in multilingual pre-trained language models (i. e., XLM-RoBERTa) in our downstream task provokes us to study other transfer learning approaches such as meta learning.

In our ongoing work, we are working on a meta learning-based approach to cover some aforementioned challenges. In this work, we have proposed a few-shot cross-lingual hate speech detection model based on meta learning approach. This study can be expanded into future interests: i) since few-shot learning has raised significant attention to the NLP research community and meta learning is a powerful solution for this problem, it is worth exploring the effect of choosing different cross-lingual base learners rather than XLM-R in the performance of the model, and studying linguistic features in different language families to select training languages resulting in better generalization in meta-learning process; and ii) the lack of held-out test set in some benchmark datasets has made it difficult to evaluate the performance of different proposed algorithms. Therefore, providing benchmark datasets in hate, offense, toxicity, etc, similar to other benchmark datasets in NLP tasks (e.g., GLUE) makes the training, evaluating, and analyzing of different models fair and easy.

References

[1] K. Müller and C. Schwarz, "Fanning the Flames of Hate: Social Media and Hate Crime," *Journal of the European Economic Association*, 10 2020, jvaa045. [Online]. Available: https://doi.org/10.1093/jeea/jvaa045

[2] Z. Waseem and D. Hovy, "Hateful symbols or hateful people? predictive features for hate speech detection on Twitter," in *Proceedings of the NAACL Student Research Workshop*. San Diego, California: Association for Computational Linguistics, June 2016, pp. 88–93. [Online]. Available: https://www.aclweb.org/anthology/N16-2013

[3] T. Davidson, D. Warmsley, M. W. Macy, and I. Weber, "Automated hate speech detection and the problem of offensive language," *CoRR*, vol. abs/1703.04009, 2017. [Online]. Available: http://arxiv.org/abs/1703.04009

[4] Z. Zhang, D. Robinson, and J. Tepper, "Detecting hate speech on twitter using a convolution-gru based deep neural network," in *The Semantic Web*, A. Gangemi, R. Navigli, M.-E. Vidal, P. Hitzler, R. Troncy, L. Hollink, A. Tordai, and M. Alam, Eds. Cham: Springer International Publishing, 2018, pp. 745–760.

[5] J. H. Park and P. Fung, "One-step and two-step classification for abusive language detection on twitter," *CoRR*, vol. abs/1706.01206, 2017. [Online]. Available: http://arxiv.org/abs/1706.01206

[6] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *CoRR*, vol. abs/1810.04805, 2018. [Online]. Available: http://arxiv.org/abs/1810.04805

[7] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. International Convention Centre, Sydney, Australia: PMLR, 06–11 Aug 2017, pp. 1126–1135. [Online]. Available: http://proceedings.mlr.press/v70/finn17a.html

[8] A. Guterres, "United nations strategy and plan of action on hate speech," accessed: 10-4-2021. [Online]. Available: https://www.un.org/en/genocideprevention/documents/advising-and-mobilizing/Action_plan_on_hate_speech_EN.pdf

[9] I. Gagliardone, D. Gal, T. Alves, and M. Gabriela, *Countering online hate speech*, ser. UNESCO series of Internet freedom. United Nations Educational, Scientific and Cultural Organization, 2015.

[10] M. Zampieri, P. Nakov, S. Rosenthal, P. Atanasova, G. Karadzhov, H. Mubarak, L. Derczynski, Z. Pitenis, and c. Çöltekin, "SemEval-2020 Task 12: Multilingual Offensive Language Identification in Social Media (OffensEval 2020)," in *Proceedings of SemEval*, 2020.

[11] M. Zampieri, S. Malmasi, P. Nakov, S. Rosenthal, N. Farra, and R. Kumar, "Predicting the type and target of offensive posts in social media," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 1415–1420. [Online]. Available: https://www.aclweb.org/anthology/N19-1144

[12] J. M. Struš, M. Siegel, J. Ruppenhofer, M. Wiegand, and M. Klenner, "Overview of germeval task 2, 2019 shared task on the identification of offensive language," ser. Preliminary proceedings of the 15th Conference on Natural Language Processing (KONVENS 2019), October 9 – 11, 2019 at Friedrich-Alexander-Universität Erlangen-Nürnberg. München [u.a.]: German Society for Computational Linguistics & Language Technology und Friedrich-Alexander-Universität Erlangen-Nürnberg, 2019, pp. 352 – 363. [Online]. Available: http://nbn-resolving.de/urn:nbn:de:bsz:mh39-93197

[13] K. Wang, Y. Cui, J. Hu, W. Zhao, L. Feng, and Y. Zhang, "Cyberbullying detection, based on the fasttext and word similarity schemes," *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*.

[14] R. Kumar, A. K. Ojha, B. Lahiri, M. Zampieri, S. Malmasi, V. Murdock, and D. Kadar, Eds., *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*. Marseille, France: European Language Resources Association (ELRA), May 2020. [Online]. Available: https://www.aclweb.org/anthology/2020.trac-1.0

[15] R. Kumar, A. K. Ojha, S. Malmasi, and M. Zampieri, "Benchmarking aggression identification in social media," in *TRAC@COLING 2018*, 2018.

[16] J. Risch and R. Krestel, "Bagging BERT models for robust aggression identification," in *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*. Marseille, France: European Language Resources Association (ELRA), May 2020, pp. 55–61. [Online]. Available: https://www.aclweb.org/anthology/2020.trac-1.9

[17] H. Liu, P. Burnap, W. Alorainy, and M. Williams, "Scmhl5 at trac-2 shared task on aggression identification: Bert based ensemble learning approach," in *TRAC@LREC*, 2020.

[18] J. Pavlopoulos, J. Sorensen, L. Dixon, N. Thain, and I. Androutsopoulos, "Toxicity detection: Does context really matter?" in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 4296–4305. [Online]. Available: https://www.aclweb.org/anthology/2020.acl-main.396

[19] C. Nobata, J. R. Tetreault, A. O. Thomas, Y. Mehdad, and Y. Chang, "Abusive language detection in online user content," in *Proceedings of the 25th International Conference on World Wide Web*, ser. WWW '16. Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee, 2016, pp. 145–153.

[20] C. D. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*. Cambridge, MA: MIT Press, 1999.

[21] A. M. TURING, "I.—COMPUTING MACHINERY AND INTELLIGENCE," *Mind*, vol. LIX, no. 236, pp. 433–460, 10 1950. [Online]. Available: https://doi.org/10.1093/mind/LIX.236.433

[22] N. Chomsky, *Syntactic Structures*. The Hague: Mouton and Co., 1957.

[23] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *J. Mach. Learn. Res.*, vol. 12, no. null, p. 2493–2537, Nov. 2011.

[24] Y. Goldberg, *Neural Network Methods for Natural Language Processing*, 2017, vol. 10, no. 1. [Online]. Available: https://doi.org/10.2200/S00762ED1V01Y201703HLT037

[25] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *J. Mach. Learn. Res.*, vol. 3, no. null, p. 1137–1155, Mar. 2003.

[26] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proceedings of the 25th International Conference on Machine Learning*, ser. ICML '08. New York, NY, USA: Association for Computing Machinery, 2008, p. 160–167. [Online]. Available: https://doi.org/10.1145/1390156.1390177

[27] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur, "Recurrent neural network based language model," in *INTERSPEECH*, 2010.

[28] A. Graves, "Generating sequences with recurrent neural networks." *CoRR*, vol. abs/1308.0850, 2013. [Online]. Available: http://dblp.uni-trier.de/db/journals/corr/corr1308.html#Graves13

[29] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Baltimore, Maryland: Association for Computational Linguistics, June 2014, pp. 655–665. [Online]. Available: https://www.aclweb.org/anthology/P14-1062

[30] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'13. Curran Associates Inc., 2013, pp. 3111–3119.

[31] T. Mikolov, K. Chen, G. S. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013. [Online]. Available: http://arxiv.org/abs/1301.3781

[32] D. Mimno and L. Thompson, "The strange geometry of skip-gram with negative sampling," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, Sept. 2017, pp. 2873–2878. [Online]. Available: https://www.aclweb.org/anthology/D17-1308

[33] L. Wendlandt, J. K. Kummerfeld, and R. Mihalcea, "Factors influencing the surprising instability of word embeddings," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 2092–2102. [Online]. Available: https://www.aclweb.org/anthology/N18-1190

[34] T. Bolukbasi, K.-W. Chang, J. Zou, V. Saligrama, and A. Kalai, "Man is to computer programmer as woman is to homemaker? debiasing word embeddings," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, ser. NIPS'16. Red Hook, NY, USA: Curran Associates Inc., 2016, p. 4356–4364.

[35] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proceedings of the 31st International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, E. P. Xing and T. Jebara, Eds., vol. 32, no. 2. Bejing, China: PMLR, 22–24 Jun 2014, pp. 1188–1196. [Online]. Available: http://proceedings.mlr.press/v32/le14.html

[36] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1532–1543.

[37] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," *CoRR*, vol. abs/1802.05365, 2018. [Online]. Available: http://arxiv.org/abs/1802.05365

[38] J. Howard and S. Ruder, "Fine-tuned language models for text classification," *CoRR*, vol. abs/1801.06146, 2018. [Online]. Available: http://arxiv.org/abs/1801.06146

[39] A. Radford and K. Narasimhan, "Improving language understanding by generative pre-training," 2018.

[40] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, Eds., vol. 27. Curran Associates, Inc., 2014. [Online]. Available: https://proceedings.neurips.cc/paper/2014/file/a14ac55a4f27472c5d894ec1c3c743d2-Paper.pdf

[41] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, L. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean, "Google's neural machine translation system: Bridging the gap between human and machine translation," *CoRR*, vol. abs/1609.08144, 2016. [Online]. Available: http://arxiv.org/abs/1609.08144

[42] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. PMLR, 06–11 Aug 2017, pp. 1243–1252. [Online]. Available: http://proceedings.mlr.press/v70/gehring17a.html

[43] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *CoRR*, vol. abs/1409.0473, 2015.

[44] W. Yin, H. Schütze, B. Xiang, and B. Zhou, "Abcnn: Attention-based convolutional neural network for modeling sentence pairs," *Transactions of the Association for Computational Linguistics*, vol. 4, pp. 259–272, 2016.

[45] Z. Lin, M. Feng, C. D. Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio, "A structured self-attentive sentence embedding," *ArXiv*, vol. abs/1703.03130, 2017.

[46] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[47] D. Otter, J. R. Medina, and J. Kalita, "A survey of the usages of deep learning for natural language processing," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, pp. 604–624, 2021.

[48] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[49] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

[50] B. McCann, J. Bradbury, C. Xiong, and R. Socher, "Learned in translation: Contextualized word vectors," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/20c86a628232a67e7bd46f76fba7ce12-Paper.pdf

[51] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 328–339. [Online]. Available: https://www.aclweb.org/anthology/P18-1031

[52] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "Albert: A lite bert for self-supervised learning of language representations," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=H1eA7AEtvS

[53] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A Robustly Optimized BERT Pretraining Approach," *arXiv e-prints*, p. arXiv:1907.11692, July 2019.

[54] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020. [Online]. Available: http://jmlr.org/papers/v21/20-074.html

[55] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," 2019.

[56] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 1877–1901. [Online]. Available: https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

[57] S. Ruder, I. Vulić, and A. Søgaard, "A survey of cross-lingual word embedding models," *J. Artif. Int. Res.*, vol. 65, no. 1, p. 569–630, May 2019. [Online]. Available: https://doi.org/10.1613/jair.1.11640

[58] T. Mikolov, Q. V. Le, and I. Sutskever, "Exploiting similarities among languages for machine translation," *CoRR*, vol. abs/1309.4168, 2013. [Online]. Available: http://arxiv.org/abs/1309.4168

[59] M. Artetxe, G. Labaka, and E. Agirre, "A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 789–798. [Online]. Available: https://www.aclweb.org/anthology/P18-1073

[60] X. Chen and C. Cardie, "Unsupervised multilingual word embeddings," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 261–270. [Online]. Available: https://www.aclweb.org/anthology/D18-1024

[61] M. Artetxe and H. Schwenk, "Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond," *Transactions of the Association for Computational Linguistics*, vol. 7, pp. 597–610, 09 2019. [Online]. Available: https://doi.org/10.1162/tacl_a_00288

[62] G. Lample and A. Conneau, "Cross-lingual language model pretraining," *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[63] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, "Unsupervised cross-lingual representation learning at scale," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 8440–8451. [Online]. Available: https://www.aclweb.org/anthology/2020.acl-main.747

[64] Z. Waseem, T. Davidson, D. Warmsley, and I. Weber, "Understanding abuse: A typology of abusive language detection subtasks," in *Proceedings of the First Workshop on Abusive Language Online*. Vancouver, BC, Canada: Association for Computational Linguistics, Aug. 2017, pp. 78–84. [Online]. Available: https://www.aclweb.org/anthology/W17-3012

[65] Z. Waseem, "Are you a racist or am I seeing things? annotator influence on hate speech detection on Twitter," in *Proceedings of the First Workshop on NLP and Computational Social Science*. Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 138–142. [Online]. Available: https://www.aclweb.org/anthology/W16-5618

[66] B. Ross, M. Rist, G. Carbonell, B. Cabrera, N. Kurowsky, and M. M. Wojatzki, "Measuring the reliability of hate speech annotations: The case of the european refugee crisis," Nov. 2016, originally published in Bochumer Linguistische Arbeitsberichte 17, NLP4CMC III: 3rd Workshop on Natural Language Processing for Computer-Mediated Communication, by Michael Beißwenger, Michael Wojatzki and Torsten Zesch (Eds.), 22 September 2016 (ISSN 2190-0949). [Online]. Available: https://duepublico2.uni-due.de/receive/duepublico_mods_00042132

[67] A.-M. Founta, C. Djouvas, D. Chatzakou, I. Leontiadis, J. Blackburn, G. Stringhini, A. Vakali, M. Sirivianos, and N. Kourtellis, "Large scale crowdsourcing and characterization of twitter abusive behavior," in *11th International Conference on Web and Social Media, ICWSM 2018*. AAAI Press, 2018.

[68] S. Malmasi and M. Zampieri, "Challenges in discriminating profanity from hate speech," *CoRR*, vol. abs/1803.05495, 2018. [Online]. Available: http://arxiv.org/abs/1803.05495

[69] S. MacAvaney, H.-R. Yao, E. Yang, K. Russell, N. Goharian, and O. Frieder, "Hate speech detection: Challenges and solutions," *PLOS ONE*, vol. 14, no. 8, pp. 1–16, 08 2019. [Online]. Available: https://doi.org/10.1371/journal.pone.0221152

[70] P. Badjatiya, S. Gupta, M. Gupta, and V. Varma, "Deep learning for hate speech detection in tweets," in *Proceedings of the 26th International Conference on World Wide Web Companion*, ser. WWW '17 Companion. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2017, p. 759–760. [Online]. Available: https://doi.org/10.1145/3041021.3054223

[71] A. Arango, J. Pérez, and B. Poblete, "Hate speech detection is not as easy as you may think: A closer look at model validation," in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR'19. New York, NY, USA: Association for Computing Machinery, 2019, p. 45–54. [Online]. Available: https://doi.org/10.1145/3331184.3331262

[72] M. Mozafari, R. Farahbakhsh, and N. Crespi, "Hate speech detection and racial bias mitigation in social media based on bert model," *PLOS ONE*, vol. 15, no. 8, pp. 1–26, 08 2020. [Online]. Available: https://doi.org/10.1371/journal.pone.0237861

[73] M. Wich, J. Bauer, and G. Groh, "Impact of politically biased data on hate speech classification," in *Proceedings of the Fourth Workshop on Online Abuse and Harms*. Online: Association for Computational Linguistics, Nov. 2020, pp. 54–64. [Online]. Available: https://www.aclweb.org/anthology/2020.alw-1.7

[74] L. Dixon, J. Li, J. Sorensen, N. Thain, and L. Vasserman, "Measuring and mitigating unintended bias in text classification," in *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, ser. AIES '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 67–73. [Online]. Available: https://doi.org/10.1145/3278721.3278729

[75] J. H. Park, J. Shin, and P. Fung, "Reducing gender bias in abusive language detection," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 2799–2804. [Online]. Available: https://www.aclweb.org/anthology/D18-1302

[76] S. Kiritchenko, I. Nejadgholi, and K. C. Fraser, "Confronting abusive language online: A survey from the ethical and human rights perspective," 2020.

[77] V. Basile, C. Bosco, E. Fersini, D. Nozza, V. Patti, F. M. Rangel Pardo, P. Rosso, and M. Sanguinetti, "SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter," in *Proceedings of the 13th International Workshop on Semantic Evaluation*. Minneapolis, Minnesota, USA: Association for Computational Linguistics, June 2019, pp. 54–63. [Online]. Available: https://www.aclweb.org/anthology/S19-2007

[78] O. de Gibert, N. Pérez, A. García-Pablos, and M. Cuadros, "Hate speech dataset from a white supremacy forum," in *ALW*, 2018.

[79] Y. Mehdad and J. Tetreault, "Do characters abuse more than words?" in *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Los Angeles: Association for Computational Linguistics, Sept. 2016, pp. 299–303. [Online]. Available: https://www.aclweb.org/anthology/W16-3638

[80] B. Pete and W. M. L., "Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making," *Policy and Internet*, vol. 7, no. 2, p. 223–242, 2015.

[81] A. Founta, D. Chatzakou, N. Kourtellis, J. Blackburn, A. Vakali, and I. Leontiadis, "A unified deep learning architecture for abuse detection," in *Proceedings of the 10th ACM Conference on Web Science*, ser. WebSci '19. New York, NY, USA: ACM, 2019, pp. 105–114.

[82] J. H. Park and P. Fung, "One-step and two-step classification for abusive language detection on Twitter," in *Proceedings of the First Workshop on Abusive Language Online*. Vancouver, BC, Canada: Association for Computational Linguistics, Aug. 2017, pp. 41–45. [Online]. Available: https://www.aclweb.org/anthology/W17-3006

[83] L. Gao and R. Huang, "Detecting online hate speech using context aware models," in *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*. Varna, Bulgaria: INCOMA Ltd., Sept. 2017, pp. 260–266. [Online]. Available: https://doi.org/10.26615/978-954-452-049-6_036

[84] B. Gambäck and U. K. Sikdar, "Using convolutional neural networks to classify hate-speech," in *Proceedings of the First Workshop on Abusive Language Online*. Vancouver, BC, Canada: Association for Computational Linguistics, Aug. 2017, pp. 85–90.

[85] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *CoRR*, vol. abs/1607.04606, 2016. [Online]. Available: http://arxiv.org/abs/1607.04606

[86] N. Djuric, J. Zhou, R. Morris, M. Grbovic, V. Radosavljevic, and N. Bhamidipati, "Hate speech detection with comment embeddings," in *Proceedings of the 24th International Conference on World Wide Web*, ser. WWW '15 Companion. New York, NY, USA: ACM, 2015, pp. 29–30.

[87] T. Davidson, D. Bhattacharya, and I. Weber, "Racial bias in hate speech and abusive language detection datasets," *CoRR*, vol. abs/1905.12516, 2019. [Online]. Available: http://arxiv.org/abs/1905.12516

[88] M. Wiegand, J. Ruppenhofer, and T. Kleinbauer, "Detection of Abusive Language: the Problem of Biased Datasets," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 602–608.

[89] J. B. Houston, G. J. Hansen, and G. S. Nisbett, "Influence of user comments on perceptions of media bias and third-person effect in online news," *Electronic News*, vol. 5, no. 2, pp. 79–92, 2011.

[90] S. Xie, J. Wang, M. S. Amin, B. Yan, A. Bhasin, C. Yu, and P. S. Yu, "A context-aware approach to detection of short irrelevant texts," in *IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, Oct. 2015, pp. 1–10.

[91] A. Zhang, B. Culbertson, and P. Paritosh, "Characterizing online discussion using coarse discourse sequences," 2017.

[92] N. C. Dang, F. De la Prieta, J. M. Corchado, and M. N. Moreno, "Framework for retrieving relevant contents related to fashion from online social network data," in *Trends in Practical Applications of Scalable Multi-Agent Systems, the PAAMS Collection*. Springer International Publishing, 2016, pp. 335–347.

[93] S. A. Salloum, M. Al-emran, A. A. Monem, and K. Shaalan, "A Survey of Text Mining in Social Media : Facebook and Twitter Perspectives," *Advances in Science, Technology and Engineering Systems Journal*, vol. 2, no. 1, pp. 127–133, 2017.

[94] A. Kothari, W. Magdy, K. Darwish, A. Mourad, and A. Taei, "Detecting comments on news articles in microblogs," in *ICWSM*, 2013.

[95] A. Sureka, "Mining user comment activity for detecting forum spammers in youtube," *CoRR*, vol. abs/1103.5044, 2011. [Online]. Available: http://arxiv.org/abs/1103.5044

[96] A. H. Wang, "Detecting spam bots in online social networking sites: A machine learning approach," in *Data and Applications Security and Privacy XXIV*, S. Foresti and S. Jajodia, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 335–342.

[97] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003. [Online]. Available: http://dl.acm.org/citation.cfm?id=944919.944937

[98] H. Liu, J. Han, and H. Motoda, "Uncovering deception in social media," *Social Network Analysis and Mining*, vol. 4, no. 1, p. 162, Feb. 2014.

[99] A. Suarez, D. Albakour, D. Corney, M. Martinez, and J. Esquivel, "A data collection for evaluating the retrieval of related tweets to news articles," in *Advances in Information Retrieval*, G. Pasi, B. Piwowarski, L. Azzopardi, and A. Hanbury, Eds. Springer International Publishing, 2018, pp. 780–786.

[100] S. Shehnepoor, M. Salehi, R. Farahbakhsh, and N. Crespi, "Netspam: A network-based spam detection framework for reviews in online social media," *Trans. Info. For. Sec.*, vol. 12, no. 7, pp. 1585–1595, July 2017. [Online]. Available: https://doi.org/10.1109/TIFS.2017.2675361

[101] A. Mukherjee, A. Kumar, B. Liu, J. Wang, M. Hsu, M. Castellanos, and R. Ghosh, "Spotting opinion spammers using behavioral footprints," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD, 2013, pp. 632–640.

[102] J. Wang, C. T. Yu, P. S. Yu, B. Liu, and W. Meng, "Diversionary comments under blog posts," *ACM Trans. Web*, vol. 9, no. 4, pp. 18:1–18:34, Sept. 2015. [Online]. Available: http://doi.acm.org/10.1145/2789211

[103] G. Fei, A. Mukherjee, B. Liu, M. Hsu, and M. C. et al, "Exploiting burstiness in reviews for review spammer detection," in *ICWSM*, 2013.

[104] G. Mishne, D. Carmel, and R. Lempel, "Blocking blog spam with language model disagreement," in *AIRWeb*, 2005.

[105] A. Bhattarai, V. Rus, and D. Dasgupta, "Characterizing comment spam in the blogosphere through content analysis," in *2009 IEEE Symposium on Computational Intelligence in Cyber Security*, Mar. 2009.

[106] J. Zhu, K. Wang, Y. Wu, Z. Hu, and H. Wang, "Mining user-aware rare sequential topic patterns in document streams," *IEEE Transactions on Knowledge & Data Engineering*, vol. 28, no. 07, July 2016.

[107] T. Landauer, P. Foltz, and D. Laham, "An introduction to latent semantic analysis," *Discourse processes*, vol. 25, pp. 259–284, 1998.

[108] D. Q. Nguyen, R. Billingsley, L. Du, and M. Johnson, "Improving topic models with latent feature word representations," *CoRR*, vol. abs/1810.06306, 2018. [Online]. Available: http://arxiv.org/abs/1810.06306

[109] X.-H. Phan, L.-M. Nguyen, and S. Horiguchi, "Learning to classify short and sparse text & web with hidden topics from large-scale data collections," in *Proceedings of the 17th International Conference on World Wide Web*, ser. WWW '08. ACM, 2008, pp. 91–100. [Online]. Available: http://doi.acm.org/10.1145/1367497.1367510

[110] X. H. Phan, C.-T. Nguyen, D.-T. Le, M. L. Nguyen, S. Horiguchi, and Q.-T. Ha, "A hidden topic-based framework toward building applications with short web documents," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, pp. 961–976, 2011.

[111] D. M. Blei, "Probabilistic topic models," *Commun. ACM*, vol. 55, no. 4, pp. 77–84, Apr. 2012. [Online]. Available: http://doi.acm.org/10.1145/2133806.2133826

[112] G. Heinrich, "Parameter estimation for text analysis," Tech. Rep., 2004.

[113] Z. Bouraoui, S. Jameel, and S. Schockaert, "Relation induction in word embeddings revisited," in *Proceedings of the 27th International Conference on Computational Linguistics*. Association for Computational Linguistics, 2018, pp. 1627–1637.

[114] T. Pedersen, S. Patwardhan, and J. Michelizzi, "Wordnet::similarity - measuring the relatedness of concepts," 04 2004.

[115] R. Řehůřek and P. Sojka, "Software Framework for Topic Modelling with Large Corpora," in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Valletta, Malta: ELRA, May 2010, pp. 45–50, http://is.muni.cz/publication/884893/en.

[116] W. G. Cochran, *Sampling Techniques, 3rd Edition*. John Wiley, 1977.

[117] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*. O'Reilly Media Inc, 2009.

[118] L. Y.-F. Su, M. A. Xenos, K. M. Rose, C. Wirz, D. A. Scheufele, and D. Brossard, "Uncivil and personal? comparing patterns of incivility in comments on the facebook pages of news outlets," *New Media & Society*, vol. 20, no. 10, pp. 3678–3699, 2018. [Online]. Available: https://doi.org/10.1177/1461444818757205

[119] M. Sap, D. Card, S. Gabriel, Y. Choi, and N. A. Smith, "The risk of racial bias in hate speech detection," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 1668–1678.

[120] P. Fortuna and S. Nunes, "A survey on automatic detection of hate speech in text," *ACM Comput. Surv.*, vol. 51, no. 4, pp. 85:1–85:30, July 2018.

[121] Z. Waseem, J. Thorne, and J. Bingel, *Bridging the Gaps: Multi Task Learning for Domain Transfer of Hate Speech Detection*. Cham: Springer International Publishing, 2018, pp. 29–55.

[122] A. Olteanu, C. Castillo, J. Boy, and K. R. Varshney, "The effect of extremist violence on hateful speech online," *CoRR*, vol. abs/1804.05704, 2018. [Online]. Available: http://arxiv.org/abs/1804.05704

[123] A. Mittos, S. Zannettou, J. Blackburn, and E. D. Cristofaro, ""And We Will Fight For Our Race!" A measurement Study of Genetic Testing Conversations on Reddit and 4chan," *CoRR*, vol. abs/1901.09735, 2019. [Online]. Available: http://arxiv.org/abs/1901.09735

[124] R. Ottoni, E. Cunha, G. Magno, P. Bernardina, W. M. Jr., and V. A. F. Almeida, "Analyzing right-wing youtube channels: Hate, violence and discrimination," in *Proceedings of the 10th ACM Conference on Web Science*, ser. WebSci '18. New York, NY, USA: ACM, 2018, pp. 323–332.

[125] M. Rizoiu, T. Wang, G. Ferraro, and H. Suominen, "Transfer learning for hate speech detection in social media," *CoRR*, vol. abs/1906.03829, 2019. [Online]. Available: http://arxiv.org/abs/1906.03829

[126] Y. Kim, "Convolutional neural networks for sentence classification," *CoRR*, vol. abs/1408.5882, 2014. [Online]. Available: http://arxiv.org/abs/1408.5882

[127] B. Vidgen and T. Yasseri, "Detecting weak and strong islamophobic hate speech on social media," *Journal of Information Technology & Politics*, vol. 17, no. 1, pp. 66–78, 2020. [Online]. Available: https://doi.org/10.1080/19331681.2019.1702607

[128] P. Badjatiya, M. Gupta, and V. Varma, "Stereotypical bias removal for hate speech detection task using knowledge-based generalizations," in *The World Wide Web Conference*, ser. WWW '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 49–59. [Online]. Available: https://doi.org/10.1145/3308558.3313504

[129] T. Schuste, D. J. Shah, Y. J. S. Yeo, D. Filizzola, E. Santus, and R. Barzilay, "Towards debiasing fact verification models." EMNLP/IJCNLP, 2019, pp. 3417–3423.

[130] S. Evert, *The Statistics of Word Cooccurrences: Word Pairs and Collocations*, 2005. [Online]. Available: https://books.google.fr/books?id=Uof3tgAACAAJ

[131] S. L. Blodgett, L. Green, and B. O'Connor, "Demographic dialectal variation in social media: A case study of African-American English," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 1119–1130. [Online]. Available: https://www.aclweb.org/anthology/D16-1120

[132] M. Mozafari, R. Farahbakhsh, and N. Crespi, "A bert-based transfer learning approach for hate speech detection in online social media," in *Complex Networks and Their Applications VIII*, H. Cherifi, S. Gaito, J. F. Mendes, E. Moro, and L. M. Rocha, Eds. Cham: Springer International Publishing, 2020, pp. 928–940.

[133] E. Fersini, D. Nozza, and P. Rosso, "Overview of the evalita 2018 task on automatic misogyny identification (ami)," in *Proceedings of the 6th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA'18)*, T. Caselli, N. Novielli, V. Patti, and P. Rosso, Eds. Turin, Italy: CEUR.org, 2018.

[134] E. Fersini, P. Rosso, and M. Anzovino, "Overview of the task on automatic misogyny identification at ibereval 2018," in *IberEval@SEPLN*, 2018.

[135] M. Guzman-Silverio, A. Balderas-Paredes, and A. P. Lopez-Monroy, "Transformers and data augmentation for aggressiveness detection in mexican spanish," in *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2020), co-located with 36th Conference of the Spanish Society for Natural Language Processing (SEPLN 2020)*, Sept. 2020.

[136] Z. Pitenis, M. Zampieri, and T. Ranasinghe, "Offensive language identification in greek," in *LREC*, 2020.

[137] H. Mubarak, K. Darwish, and W. Magdy, "Abusive language detection on arabic social media," in *ALW@ACL*, 2017.

[138] H. Mubarak, A. Rashed, K. Darwish, Y. Samih, and A. Abdelali, "Arabic offensive language on twitter: Analysis and experiments," 2021.

[139] c. Çöltekin, "A corpus of turkish offensive language on social media," in *Proceedings of The 12th Language Resources and Evaluation Conference*, Marseille, France, 2020, pp. 6174–6184. [Online]. Available: https://www.aclweb.org/anthology/2020.lrec-1.758

[140] M. Farahani, M. Gharachorloo, M. Farahani, and M. Manthouri, "Parsbert: Transformer-based model for persian language understanding," *ArXiv*, vol. abs/2005.12515, 2020.

[141] M. Farahani, "Albert-persian: A lite bert for self-supervised learning of language representations for the persian language," *GitHub repository*, 2020.

[142] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, "Unsupervised cross-lingual representation learning at scale," in *ACL*, 2020.

[143] M. Zampieri, S. Malmasi, P. Nakov, S. Rosenthal, N. Farra, and R. Kumar, "Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval)," in *Proceedings of the 13th International Workshop on Semantic Evaluation*, 2019, pp. 75–86.

[144] J. Ruppenhofer, M. Siegel, and M. Wiegand, Eds., *Proceedings of GermEval 2018, 14th Conference on Natural Language Processing (KONVENS 2018), Vienna, Austria - September 21, 2018*. Vienna, Austria: Austrian Academy of Sciences, 2019. [Online]. Available: http://nbn-resolving.de/urn:nbn:de:bsz:mh39-84901

[145] R. Kumar, A. K. Ojha, M. Zampieri, and S. Malmasi, Eds., *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, Aug. 2018. [Online]. Available: https://www.aclweb.org/anthology/W18-4400

[146] Z. Waseem, W. H. K. Chung, D. Hovy, and J. Tetreault, Eds., *Proceedings of the First Workshop on Abusive Language Online*. Vancouver, BC, Canada: Association for Computational Linguistics, Aug. 2017. [Online]. Available: https://www.aclweb.org/anthology/W17-3000

[147] D. Fišer, R. Huang, V. Prabhakaran, R. Voigt, Z. Waseem, and J. Wernimont, Eds., *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018. [Online]. Available: https://www.aclweb.org/anthology/W18-5100

[148] S. T. Roberts, J. Tetreault, V. Prabhakaran, and Z. Waseem, Eds., *Proceedings of the Third Workshop on Abusive Language Online*. Florence, Italy: Association for Computational Linguistics, Aug. 2019. [Online]. Available: https://www.aclweb.org/anthology/W19-3500

[149] C. Bosco, F. Dell'Orletta, F. Poletto, M. Sanguinetti, and M. Tesconi, "Overview of the EVALITA 2018 hate speech detection task," in *Proceedings of the Sixth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2018) co-located with the Fifth Italian Conference on Computational Linguistics (CLiC-it 2018), Turin, Italy, December 12-13, 2018*, ser. CEUR Workshop Proceedings, T. Caselli, N. Novielli, V. Patti, and P. Rosso, Eds., vol. 2263. CEUR-WS.org, 2018. [Online]. Available: http://ceur-ws.org/Vol-2263/paper010.pdf

[150] T. Mandl, S. Modha, P. Majumder, D. Patel, M. Dave, C. Mandalia, and A. Patel, "Overview of the hasoc track at fire 2019: Hate speech and offensive content identification in indo-european languages," in *FIRE '19*, 2019.

[151] H. Mubarak, K. Darwish, W. Magdy, T. Elsayed, and H. Al-Khalifa, "Overview of OSACT4 Arabic offensive language detection shared task," in *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*. Marseille, France: European Language Resource Association, May 2020, pp. 48–52. [Online]. Available: https://www.aclweb.org/anthology/2020.osact-1.7

[152] G. Imane, A. Ahsan, A. Faical, C. Sara, M. Hanene, and H. Thinhinane, "Detecting hate speech against politicians in arabic community on social media," *International Journal of Web Information Systems*, vol. 16, no. 3, pp. 295–313, Jan. 2020. [Online]. Available: https://doi.org/10.1108/IJWIS-08-2019-0036

[153] S. Chopra, R. Sawhney, P. Mathur, and R. Shah, "Hindi-english hate speech detection: Author profiling, debiasing, and practical perspectives," in *AAAI*, 2020.

[154] M. P. Akhter, Z. Jiangbin, I. R. Naqvi, M. Abdelmajeed, and M. T. Sadiq, "Automatic detection of offensive language for urdu and roman urdu," *IEEE Access*, vol. 8, pp. 91 213–91 226, 2020.

[155] M. M. Khan, K. Shahzad, and M. K. Malik, "Hate speech detection in roman urdu," *ACM Transactions on Asian and Low-Resource Language Information Processing*, Accepted July 2020.

[156] A. Schmidt and M. Wiegand, "A survey on hate speech detection using natural language processing," in *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*. Valencia, Spain: Association for Computational Linguistics, Apr. 2017, pp. 1–10. [Online]. Available: https://www.aclweb.org/anthology/W17-1101

[157] E. Bassignana, V. Basile, and V. Patti, "Hurtlex: A multilingual lexicon of words to hurt," in *Proceedings of CLiC-it*. Turin: CEUR, Dec. 2018.

[158] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, and T. Mikolov, "Learning word vectors for 157 languages," in *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[159] S. Mohtaj, B. Roshanfekr, A. Zafarian, and H. Asghari, "Parsivar: A language processing toolkit for Persian," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA), May 2018. [Online]. Available: https://www.aclweb.org/anthology/L18-1179

[160] B. Gambäck and U. K. Sikdar, "Using convolutional neural networks to classify hate-speech," in *ALW@ACL*, 2017.

[161] Z. Zhang, D. Robinson, and J. Tepper, "Hate speech detection using a convolution-lstm based deep neural network," in *ESWC 2018: The semantic web*, 2018.

[162] D. Opitz and R. Maclin, "Popular ensemble methods: An empirical study," *J. Artif. Int. Res.*, vol. 11, no. 1, p. 169–198, July 1999.

[163] N. C. Oza and K. Tumer, "Classifier ensembles: Select real-world applications," *Information Fusion*, vol. 9, no. 1, pp. 4–20, 2008, special Issue on Applications of Ensemble Methods. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1566253507000620

[164] J. Risch and R. Krestel, "Aggression identification using deep learning and data augmentation," in *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, Aug. 2018, pp. 150–158. [Online]. Available: https://www.aclweb.org/anthology/W18-4418

[165] Y. Li, J. Gao, Q. Li, and W. Fan, "Ensemble learning," in *Data Classification: Algorithms and Applications*, C. C. Aggarwal, Ed. CRC Press, 2014, ch. 19, pp. 443–504. [Online]. Available: http://www.crcnetbase.com/doi/abs/10.1201/b17320-7

[166] L. Zintgraf, K. Shiarli, V. Kurin, K. Hofmann, and S. Whiteson, "Fast context adaptation via meta-learning," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 09–15 Jun 2019, pp. 7693–7702. [Online]. Available: http://proceedings.mlr.press/v97/zintgraf19a.html

[167] J. Gu, Y. Wang, Y. Chen, V. O. K. Li, and K. Cho, "Meta-learning for low-resource neural machine translation," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 3622–3631. [Online]. Available: https://www.aclweb.org/anthology/D18-1398

[168] T. Bansal, R. Jha, and A. McCallum, "Learning to few-shot learn across diverse natural language classification tasks," in *Proceedings of the 28th International Conference on Computational Linguistics*. Barcelona, Spain (Online): International Committee on Computational Linguistics, Dec. 2020, pp. 5108–5123. [Online]. Available: https://www.aclweb.org/anthology/2020.coling-main.448

[169] E. Triantafillou, T. Zhu, V. Dumoulin, P. Lamblin, K. Xu, R. Goroshin, C. Gelada, K. Swersky, P. Manzagol, and H. Larochelle, "Meta-dataset: A dataset of datasets for learning to learn from few examples," *CoRR*, vol. abs/1903.03096, 2019. [Online]. Available: http://arxiv.org/abs/1903.03096

[170] T. Ranasinghe and M. Zampieri, "Multilingual offensive language identification with cross-lingual embeddings," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 5838–5844. [Online]. Available: https://www.aclweb.org/anthology/2020.emnlp-main.470

[171] M. Corazza, S. Menini, E. Cabrio, S. Tonelli, and S. Villata, "A multilingual evaluation for online hate speech detection," *ACM Trans. Internet Technol.*, vol. 20, no. 2, Mar. 2020. [Online]. Available: https://doi.org/10.1145/3377323

[172] N. Vashistha and A. Zubiaga, "Online multilingual hate speech detection: Experimenting with hindi and english social media," *Information*, vol. 12, no. 1, 2021. [Online]. Available: https://www.mdpi.com/2078-2489/12/1/5

[173] M. O. Ibrohim and I. Budi, "Translated vs non-translated method for multilingual hate speech identification in twitter," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 9, no. 4, pp. 1116–1123, 2019. [Online]. Available: http://ijaseit.insightsociety.org/index.php?option=com_content&view=article&id=9&Itemid=1&article_id=8123

[174] N. Ousidhoum, Z. Lin, H. Zhang, Y. Song, and D.-Y. Yeung, "Multilingual and multi-aspect hate speech analysis," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 4675–4684. [Online]. Available: https://www.aclweb.org/anthology/D19-1474

[175] S. Wang, J. Liu, X. Ouyang, and Y. Sun, "Galileo at semeval-2020 task 12: Multi-lingual learning for offensive language identification using pre-trained language models," *ArXiv*, vol. abs/2010.03542, 2020.

[176] G. Wiedemann, S. M. Yimam, and C. Biemann, "Uhh-lt & lt2 at semeval-2020 task 12: Fine-tuning of pre-trained transformer networks for offensive language detection," *ArXiv*, vol. abs/2004.11493, 2020.

[177] L. Stappen, F. Brunn, and B. Schuller, "Cross-lingual zero- and few-shot hate speech detection utilising frozen transformer language models and axel," *ArXiv*, vol. abs/2004.13850, 2020.

[178] S. S. Aluru, B. Mathew, P. Saha, and A. Mukherjee, "Deep learning models for multilingual hate speech detection," *CoRR*, vol. abs/2004.06465, 2020. [Online]. Available: https://arxiv.org/abs/2004.06465

[179] E. W. Pamungkas, V. Basile, and V. Patti, "A joint learning approach with knowledge injection for zero-shot cross-lingual hate speech detection," *Information Processing & Management*, vol. 58, no. 4, p. 102544, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306457321000510

[180] Z.-Y. Dou, K. Yu, and A. Anastasopoulos, "Investigating meta-learning algorithms for low-resource natural language understanding tasks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 1192–1197. [Online]. Available: https://www.aclweb.org/anthology/D19-1112

[181] F. Nooralahzadeh, G. Bekoulis, J. Bjerva, and I. Augenstein, "Zero-shot cross-lingual transfer with meta learning," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 4547–4562. [Online]. Available: https://www.aclweb.org/anthology/2020.emnlp-main.368

[182] I. Tarunesh, S. Khyalia, V. Kumar, G. Ramakrishnan, and P. Jyothi, "Meta-learning for effective multi-task and multilingual modelling," 2021.

[183] S. Thrun and L. Pratt, *Learning to Learn: Introduction and Overview*. Boston, MA: Springer US, 1998, pp. 3–17. [Online]. Available: https://doi.org/10.1007/978-1-4615-5529-2_1

[184] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," 2015.

[185] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/cb8da6767461f2812ae4290eac7cbc42-Paper.pdf

[186] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, "Meta-learning with memory-augmented neural networks," in *Proceedings of The 33rd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. F. Balcan and K. Q. Weinberger, Eds., vol. 48. New York, New York, USA: PMLR, 20–22 Jun 2016, pp. 1842–1850. [Online]. Available: http://proceedings.mlr.press/v48/santoro16.html

[187] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. [Online]. Available: https://openreview.net/forum?id=rJY0-Kcll

[188] A. Antoniou, H. Edwards, and A. Storkey, "How to train your MAML," in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=HJGven05Y7

[189] I. Loshchilov and F. Hutter, "Sgdr: Stochastic gradient descent with warm restarts," in *International Conference on Learning Representations (ICLR) 2017 Conference Track*, Apr. 2017.

[190] H. Mulki, H. Haddad, C. Bechikh Ali, and H. Alshabani, "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, Aug. 2019, pp. 111–118. [Online]. Available: https://www.aclweb.org/anthology/W19-3512

[191] J. C. Pereira-Kohatsu, L. Quijano-Sánchez, F. Liberatore, and M. Camacho-Collados, "Detecting and monitoring hate speech in twitter," *Sensors*, vol. 19, no. 21, 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/21/4654

[192] M. O. Ibrohim and I. Budi, "Multi-label hate speech and abusive language detection in Indonesian Twitter," in *Proceedings of the Third Workshop on Abusive Language Online*. Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 46–57. [Online]. Available: https://www.aclweb.org/anthology/W19-3506

[193] I. Alfina, R. Mulia, M. I. Fanany, and Y. Ekanata, "Hate speech detection in the indonesian language: A dataset and preliminary study," in *2017 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, 2017, pp. 233–238.

[194] P. Fortuna, J. Rocha da Silva, J. Soler-Company, L. Wanner, and S. Nunes, "A hierarchically-labeled Portuguese hate speech dataset," in *Proceedings of the Third Workshop on Abusive Language Online*. Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 94–104. [Online]. Available: https://www.aclweb.org/anthology/W19-3510

[195] G. Sigurbergsson and L. Derczynski, "Offensive language and hate speech detection for danish," in *Proceedings of the International Conference on Language Resources and Evaluation*. European Language Resources Association, May 2020, p. 3498–3508.

[196] K. Kann, K. Cho, and S. R. Bowman, "Towards realistic practices in low-resource natural language processing: The development set," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 3342–3349. [Online]. Available: https://www.aclweb.org/anthology/D19-1329

[197] M. Mozafari, R. Farahbakhsh, and N. Crespi, "Content similarity analysis of written comments under posts in social media," in *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, 2019, pp. 158–165.

[198] F. Del Vigna12, A. Cimino23, F. Dell'Orletta, M. Petrocchi, and M. Tesconi, "Hate me, hate me not: Hate speech detection on facebook," in *Proceedings of the First Italian Conference on Cybersecurity (ITASEC17)*, 2017, pp. 86–95. [Online]. Available: http://ceur-ws.org/Vol-1816/paper-09.pdf

[199] M. Sanguinetti, F. Poletto, C. Bosco, V. Patti, and M. Stranisci, "An Italian Twitter corpus of hate speech against immigrants," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA), May 2018. [Online]. Available: https://www.aclweb.org/anthology/L18-1443

List of figures

2.1 Different learning processes between traditional machine learning and transfer learning [48]. 34

3.1 Schema of the proposed framework. 48

3.2 WordCloud of related comments following the sampled posts; the more important a word makes the larger its size. 60

3.3 WordCloud of unrelated comments following the sampled posts; the more important a word makes the larger its size. 61

3.4 Distribution of related/unrelated comments following all the posts within a period of 24 hours. 62

3.5 The portion of related/unrelated comments written under 4 sampled posts within the first 12 hours. 63

3.6 Distribution of toxicity scores for different attributes derived from Perspective API. 64

4.1 The proposed framework for hate speech detection and bias mitigation tasks. It consists of two different modules: Hate Speech Detection and Bias Mitigation with different inputs as a result of different pre-processing approaches. The pre-trained $\text{BERT}_{\text{base}}$ is a common component between two modules that is fine-tuned differently in respect of each module's goal. 70

4.2 Fine-tuning strategies 74

4.3 The performances of hate speech detection models trained with a variation of training sets on Davidson and Waseem datasets. The x-axis is the portion of the training and validation sets used for training our BERT-based model and the baselines, the y-axis shows the F1-measure. 80

4.4 Waseem-samples' embeddings analysis before and after fine-tuning. To investigate the impact of information included in different layers of BERT, sentence embeddings are extracted from all the layers of the pre-trained BERT model fine-tuning, using the bert-as-service tool. Embedding vectors of size 768 are visualized to a two-dimensional visualization of the space of all Waseem-dataset samples using PCA method. For sake of clarity, we just include visualization of the first 4 layers (1-4), which are close to the training output, and the last 4 layers (9-12), which are close to the word embedding, of the pre-trained BERT model before and after fine-tuning. . . 83

4.5 Davidson-samples' embeddings alanysis before and after fine-tuning. To investigate the impact of information included in different layers of BERT, sentence embeddings are extracted from all the layers of the pre-trained BERT model before (a) and after (b) fine-tuning, using the bert-as-service tool. Embedding vectors of size 768 are visualized to a two-dimensional visualization of the space of all Davidson-dataset samples using PCA method. For sake of clarity, we just include visualization of the first 4 layers (1-4), which are close to the training output, and the last 4 layers (9-12), which are close to the word embedding, of the pre-trained BERT model before and after fine-tuning. 84

4.6 Evaluation results by confusion matrix. 85

4.7 The top 20 LMI-ranked n-grams ($n = 2$) that are highly correlated with the negative classes of Waseem-dataset (Racism and Sexism) in the training and test sets. nan value denotes computationally infeasible, as the occurrence is zero in the test set. 91

4.8 The top 20 LMI-ranked n-grams ($n = 2$) that are highly correlated with the negative classes of Davidson dataset (Hate and Offensive) in the training and test sets. nan value denotes computationally infeasible, as the occurrence is zero in the test set. 92

5.1 The Proposed workflow of the offensive language detection methodology in Persian language. 103

5.2 Tweet samples (original and translated) from the annotated data with their categories for each level of the annotation schema. 110

5.3 Pairwise Pearson Correlation Coefficient between the predicted probabilities of different single classifiers on out-of-fold test set. First level (a) shows the correlation between the output predictions of classifiers trained on offensive vs non-offensive annotated data. Second level (b) shows the correlation between the output predictions of classifiers trained on targeted vs untargeted samples. Third level (c) shows the correlation between the output predictions of classifiers trained on targeted offensive towards individual or group. . . . 122

5.4 Offensive language identification performance among all models in three levels of annotation. First level (a), Second level (b), and Third level (c) indicate performance of selected base-level classifiers accompanying stacking ensemble classifier in identification of offensive vs non-offensive, targeted vs untargeted offensive content, and the target of offensive language towards individual or group, respectively. 123
5.5 An overview of the cross-lingual meta learning-based framework for few-shot hate speech classification task. 130
5.6 Differences in the performance of Proto-MAML after removing each training language from the train set, in terms of F1-measure. Rows correspond to target languages and columns correspond to the removed language from the original train set. Each cell reports performance differences between training on the original train set and the train set without a specific training language. 141

List of tables

2.1 Hate speech definition from different sources. 29

3.1 Feature sets of the proposed framework 49
3.2 Data sampling 54
3.3 Performance of different feature combinations. 56
3.4 Impact of combining a topical approach with word embeddings on identifying related/unrelated contents 57
3.5 Impact of pre-trained word embeddings models on identifying related/unrelated contents 57
3.6 Performance metrics evaluation in different approaches 58
3.7 Four sampled posts from BBC news agency page on Facebook. 59
3.8 Definition of toxicity attributes from perspective API. 64

4.1 Datasets description. The columns show the total number of tweets, the different categories and the percentage of tweets belong to each one in the datasets, respectively. 76
4.2 Dataset statistics for Waseem-dataset and Davidson-dataset. Splits are produced using stratified sampling to select 0.8, 0.1, and 0.1 portions of tweets from each class (racism/sexism/neither or hate/offensive/neither) for train, validation, and test samples, respectively. 78
4.3 The performance of different trained classifiers on Waseem-dataset in terms of F1-measure. 78
4.4 The performance of different trained classifiers on Davidson-dataset in terms of F1-measure. 78
4.5 Misclassified samples from Waseem-dataset. 86
4.6 Misclassified samples from Davidson-dataset. 86
4.7 Racial bias analysis before and after reweighting the training data. To quantify the impact of the re-weighting mechanism in alleviating the racial bias propagated through trained classifiers, we examine our BERT-based classifiers trained on Davidson and Waseem datasets with and without re-weighting mechanism on AAE-aligned and SAE-aligned samples. 96

4.8 Performance evaluation after applying the re-weighting mechanism. To quantify the impact of the re-weighting mechanism in the performance of our pre-trained BERT-based model (with $\text{BERT}_{\text{base}}$ strategy for fine-tuning), we examine the classifier trained on Waseem and Davidson datasets with and without re-weighting mechanism on the training set in terms of macro precision, recall, and F1-measure. 97
4.9 Top 20 unigrams and 2-grams highly correlated with AAE and SAE languages. 98

5.1 Shared tasks in identification of abusive language in different types and languages. 105
5.2 Distribution of annotated data in three levels of annotation schema. A set of 6k out of 520k sampled data is randomly selected for annotation process. 110
5.3 Description of the transformer-based neural network models used in identification of offensive language in Persian. 114
5.4 Results of offensive language identification (first level). The bold and underline numbers represent the first and second best scores, respectively, in each category: classical ML, DL, and transformer-based neural networks. 119
5.5 Results of targeted offensive language identification (second level). The bold and underline numbers represent the first and second best scores, respectively, in each category: classical ML, DL, and transformer-based neural networks. 119
5.6 Results of target type of offensive language identification (third level). The bold and underline numbers represent the first and second best scores, respectively, in each category: classical ML, DL, and transformer-based neural networks. 120
5.7 Dataset description for hate speech and offensive language detection tasks. Class 0 and Class 1 represent normal and hate/offensive labels in the datasets, respectively. 135
5.8 Results of k-shot classification on the unseen target languages of hate speech dataset in terms of macro F1-measure with standard deviation. The values in bold indicate the best performing model in each k-shot setting. The last column corresponds to the average F1-measure across all target languages. . 139
5.9 Results of k-shot classification on the unseen target languages of offensive language dataset in terms of macro F1-measure with standard deviation. The values in bold indicate the best performing model in each k-shot setting. The last column corresponds to the average F1-measure across all target languages. 140

Appendix A

Appendix

A.1 Hate Speech Datasets

We use 15 publicly available sources in 8 languages provided by research community as follows:

Arabic This category consists of two hateful datasets in Arabic, explained in the following:

- Mulki et al. [190] introduced the first Levantine hate speech and abusive Twitter dataset in size of 5,846. Levantine is one of the Arabic dialects used on Twitter. The dataset was collected based on different strategies including: 1) querying for tweets containing the potential entities that are usually targeted by hate or abusive language, and 2) using user timelines belonging to certain politicians, social/political activists and TV anchors with high probability of receiving hate content regarding their tweets and tweets' replies. Three Levantine native speakers annotated the data as *hate*, *abusive*, or *normal*. Here, we only select the tweets labeled as hate or normal.

- Ousidhoum et al. [174] built a dataset containing 13,014 tweets in English (5,647), French (4,014), and Arabic (3,353) from Twitter. Here, we just select tweets that have hateful or normal sense in their annotation labels from Arabic samples (3,353).

English This category consists of four different hateful datasets in English, explained in the following:

- Basile et al. [77] introduced a multilingual hate speech dataset in English and Spanish for HatEval 2019, a shared task at SemEval 2019, which focuses on identification of multilingual hate speech against immigrants and women in Twitter. The dataset was collected by employing different approaches such as monitoring potential victims of hate accounts, using a set of keywords to filter tweets, and downloading the history of identified haters, and resulted in a composition of 19,600 tweets for English (13,000) and Spanish (6,600). Authors used the crowdsourcing platform Figure Eight (F8) to annotate the data in three categories including: 1) hate speech (*hate speech* or *not hate speech* towards immigrant or

women, 2) target range (*generic* or *individual*), and 3) aggressiveness (*aggressive* or *not aggressive*). Here, we only select the first category of annotation for English, in which each tweet is labeled as hate speech or not hate speech.

- Davidson et al. [3] built a dataset by crawling and annotating 24,783 tweets in English with using the Twitter API. This dataset was collected using a hate speech lexicon containing words and phrases issued by Hatebase[1] dictionary, and was annotated using the crowdsourcing platform CrowdFlower[2]. Each tweet was labeled as *hate speech*, *offensive*, or *neither*. Here, we only select tweets that are labeled as hate speech or neither.

- Founta et al. [67] proposed a methodology for annotating a large-scale dataset that were randomly sampled from Twitter utilizing the Twitter Stream API. The randomly sampled data, in size of 32 million tweets, was boosted with tweets that are likely to belong into the minority classes (containing inappropriate speech) and resulted in 80K tweets. The dataset was annotated to four classes: *hate speech*, *abusive*, *spam*, and *normal* by using a crowdsourcing platform CrowdFlower. Here, we only select tweets marked as either hate or normal.

- Ousidhoum et al. [174] built a dataset containing 13,014 tweets in English (5,647), French (4,014), and Arabic (3,353) from Twitter. The authors proposed a multi-aspect annotation schema to annotate the dataset as *offensive*, *disrespectful*, *hateful*, *fearful*, *abusive*, or *normal* using a crowdsourcing mechanism with the Amazon Mechanical Turk[3] platform. They also considered directness and target of hatred and the sentiment of the annotator in their annotation process. Here, we only select tweets in English that have hateful or normal sense in their annotation label.

French We use the dataset introduced in [174], containing 13,014 tweets in English (5,647), French (4,014), and Arabic (3,353) from Twitter. Here, we just select tweets that have hateful or normal sense in their annotation label that results in 1,220 samples in French.

German This category consists of two different hateful datasets in German, explained in the following:

- Mandl et al. [150] created a corpus of size 17,657 in three languages English (7,005), Hindi (5,983), and German (4,669) from Twitter and Facebook, which was introduced in the first edition of HASOC track (Hate Speech and Offensive Content Identification in Indo-European Languages shared task in FIRE 2019). The dataset was collected using a set of hashtags and keywords containing offensive content and users' timelines with potential hateful content. The dataset was annotated in a three-layer annotation schema as: 1) identification of *Hate and Offensive* or *Non Hate-Offensive*, 2) identifying the type of hate as *Hate speech*, *Offensive*, *Profane*, and 3) identifying whether a post is containing *Targeted Insult* or *Untargeted*. Here, we only select samples from the first and second layers of annotation labeled as hate speech or not hate speech.

[1] `https://hatebase.org`
[2] Now the name of platform is changed to Appen: `https://appen.com/`
[3] `https://www.mturk.com/`

- Ross et al. [66] introduced the first hate speech corpus, consisting of 469 tweets, for the refugee crisis in German language. The aim of the study was to measure the reliability of hate speech annotations. To collect the dataset, they used a list of hashtags with potential insulting or offensive meaning towards refugees. Two experts annotated the corpus as *hate speech* or*not hate speech*. In addition, the offensiveness of each tweet was rated from 1 (Not offensive at all) to 6 (Very offensive). Here, we select tweets according to a complete agreement between annotators, which results in 369 tweets.

Indonesian We use two following datasets proposed for hate speech detection in Indonesian [192, 193].

- Alfina et al. [193] introduced a dataset for hate speech detection in Indonesian containing 713 tweets and was collected from Twitter based on a set of hashtags related to the political events. The dataset was annotated as *hate speech* or *not hate speech* by a group of 30 college students as annotators.

- Ibrohim et al. [192] built an Indonesian Twitter corpus in size of 13,169 to detect hate speech and abusive language along with the target, category, and level of hate. The dataset contains a combination of existing datasets and new dataset collected from Twitter using Twitter Search API for a duration of 7 months. The dataset was annotated by a large group of annotators using crowdsourcing mechanism and resulted in a multi-label hate speech and abusive language dataset. Here, we only select tweets that are labeled as *hate speech* or *not hate speech*.

Italian Bosco et al. [149] used two Italian corpus form Twitter and Facebook for the Hate Speech Detection (HaSpeeDe) task at EVALITA 2018. The first dataset is a collection of 4,000 Facebook posts provided by [198], and the second dataset is a collection of 4,000 tweets from Twitter built by [199]. To keep platform consistency across different datasets, we only use the Twitter dataset here. The Twitter dataset was collected by considering three potential hate speech targets in the Italian context: immigrants, Muslims, and Roma and with employing a set of neutral keywords associated with these groups. Using a combination of experts and crowdsourcing annotators, the dataset was annotated as *hate speech*, *aggressiveness*, *offensiveness*, *irony*, *stereotype*, and *intensity*. Here, we only select tweets labeled as hate speech or not hate speech.

Portuguese This dataset composes of 5,668 tweets in Portuguese [194]. Tweets were collected using a set of hate-related keywords and hate-related profiles. The authors used two annotation schemas: 1) binary annotation (*hate* vs. *no-hate*) relying on non-expert annotators and 2) multi-label hate speech hierarchical annotation (including 81 hate categories) relying on an expert annotator (a researcher in hate speech domain who was trained in social psychology). Here, we only select tweets annotated with binary annotation schema as hate or no-hate.

Spanish This category consists of two hateful datasets in Spanish, explained in the following:

- Basile et al. [77] introduced a multilingual hate speech dataset in English and Spanish for HatEval 2019, a shared task at SemEval 2019, which focuses on identification of multilingual hate speech against immigrants and women in Twitter. The dataset was composed of 19,600 tweets for English (13,000) and Spanish (6,600). Here, we only select the first category of annotation for Spanish (6,600), in which each tweet is labeled as hate speech or not hate speech.

- Pereira et al. [191] introduced a dataset on hate speech in Spanish consisting of 6,000 tweets filtered from a corpus of two million tweets, sampled from Twitter using the Twitter Rest API. The filtering process was based on different dictionaries containing absolute hate or relative hate with generic insults. Using expert annotators, the dataset was labeled as *hate speech* or *not hate speech*. Here, we use all samples in the dataset.

A.2 Offensive Language Datasets

We use the multilingual offensive language dataset provided in OffensEval-2020 along with our Persian offensive language dataset, as follows:

Arabic This dataset contains 10,000 tweets in Arabic collected from Twitter and annotated by an experienced annotator who is a native Arabic speaker and familiar with several Arabic dialects [138]. The authors considered a specific pattern in tweets to increase the chance of having offensive content, so that an initial collection of 660K tweets having at least two vocative particles ("yA" in Arabic - meaning "O") were collected. The intuition was that the vocative particle ("yA") is mainly used in directing the speech to a specific person or group and this vocative is widely observed in all Arabic dialects containing offensive language. Then 10K out of the initial corpus was selected and annotated as offensive or clean. If a tweet is offensive, then annotator searched for any potential vulgar or hate speech content. Therefore, each tweet is given one or more labels: *offensive*, *vulgar*, *hate speech*, or *clean*. In this study, we consider tweets annotated as offensive or clean.

Danish This dataset contains 3,600 comments collected from different three popular social media platforms among Danish speakers: Twitter, Facebook, and Reddit. An initial platform-specific lexicon containing abusive terms in Danish collected through a crowdsourcing mechanism in Reddit was used in data collection process [195]. The annotation process followed the three-layer annotation scheme proposed in [11], for English, to identify the type and the target of offense. Here, we just used the first level of annotation where each comment is annotated as offensive or non-offensive.

English The Offensive Language Identification Dataset (OLID) containing over 14,000 English tweets, is introduced at SemEval-2019 for identification of offensive language, the type of offensive content, and the target of offensive in English [11]. The OLID targeted different kinds of offensive content and was annotated using a fine-grained three-layer annotation scheme to identify the type and the target of offense as well. In the first level of

annotation, tweets are annotated as *offensive* or *non-offensive*. In the second level, offensive tweets are annotated as *targeted insult* or *untargeted*, and in the third level, targeted offensive tweets are annotated as *individual*, *group*, or *other*. Here, we just used the first level of annotation where each tweet is annotated as offensive or non-offensive.

Greek The first version of this dataset, named Offensive Greek Tweet Dataset (OGTD), contains 4,779 posts from Twitter collected between May and June 2019 [136]. Different sampling strategies were used in collecting data including: 1) using popular and trending hashtags in Greek attributed to the television programs, reality and entertainment shows and political tweets, querying for tweets containing keywords usually found in offensive content such as curse words, expletives and their plural forms, and searching for tweets containing (eisai, "you are") as a keyword. Following the same annotation guidelines proposed in [11], the dataset was annotated as *offensive*, *not offensive* and *spam*, by a group of three volunteers annotators through the LightTag[4] platform. The spam tweets were filtered out from the dataset. To enrich the corpus for OffensEval 2020, the second version of the dataset, in size 5,508, was collected and annotated in November 2019 with the same approach used in the first version. The combination of two versions results in 10,287 tweet samples that we use in this study.

Turkish This dataset contains over 35,000 tweets extracted from Twitter using Twitter streaming API, from March 2018 to September 2019 [139]. Although a list of frequent words in Turkish tweets was used to filter Twitter streams, all the tweets were sampled uniformly without any strategy such as offensive keywords for extracting offensive content specifically. To annotate the corpus by volunteers, the annotation guidelines proposed in [11] with a small divergence was used; where at the top level, tweets were labeled as *offensive* or *non-offensive* and then offensive content were labeled as *targeted* or *profanity*. Similar to [11], the targeted offensive content were divided to *individual*, *group*, or *other*. Here, we just used the first level of annotation where each tweet is annotated as offensive or non-offensive.

Persian The details about this dataset are included in Section 5.2.3.

[4] `https://www.lighttag.io`

ECOLE DOCTORALE

Titre : Détection du Discours de Haine et du Langage Offensant utilisant des Approches de Transfer Learning

Mots clés : Discours de haine, Langage offensant, Apprentissage par transfert, BERT, XLM-RoBERTa, l'apprentissage en profondeur, Classification interlinguistique des textes, Few-shot learning, Meta learning, Réseaux sociaux, Twitter

Résumé : Une des promesses des plateformes de réseaux sociaux (comme Twitter et Facebook) est de fournir un endroit sûr pour que les utilisateurs puissent partager leurs opinions et des informations. Cependant, l'augmentation des comportements abusifs, comme le harcèlement en ligne ou la présence de discours de haine, est bien réelle. Dans cette thèse, nous nous concentrons sur le discours de haine, l'un des phénomènes les plus préoccupants concernant les réseaux sociaux.

Compte tenu de sa forte progression et de ses graves effets négatifs, les institutions, les plateformes de réseaux sociaux et les chercheurs ont tenté de réagir le plus rapidement possible. Les progrès récents des algorithmes de traitement automatique du langage naturel (NLP) et d'apprentissage automatique (ML) peuvent être adaptés pour développer des méthodes automatiques de détection des discours de haine dans ce domaine.

Le but de cette thèse est d'étudier le problème du discours de haine et de la détection des propos injurieux dans les réseaux sociaux. Nous proposons différentes approches dans lesquelles nous adaptons des modèles avancés d'apprentissage par transfert (TL) et des techniques de NLP pour détecter automatiquement les discours de haine et les contenus injurieux, de manière monolingue et multilingue.

La première contribution concerne uniquement la langue anglaise. Tout d'abord, nous analysons le contenu textuel généré par les utilisateurs sur Facebook en introduisant un nouveau cadre capable de catégoriser le contenu en termes de similarité basée sur différentes caractéristiques, à savoir les caractéristiques lexicales, topiques et sémantiques. En outre, en utilisant l'API Perspective de Google, nous mesurons et analysons la toxicité du contenu. Ensuite, nous proposons une approche TL pour l'identification des discours de haine en utilisant une combinaison du modèle non supervisé pré-entraîné BERT (Bidirectional Encoder Representations from Transformers) et de nouvelles stratégies supervisées de réglage fin. Enfin, nous étudions l'effet du biais involontaire dans notre modèle pré-entraîné BERT et proposons un nouveau mécanisme de généralisation dans les données d'entraînement en repondérant les échantillons puis en changeant les stratégies de réglage fin en termes de fonction de perte pour atténuer le biais racial propagé par le modèle. Pour évaluer les modèles proposés, nous utilisons trois datasets publics provenant de Twitter.

Dans la deuxième contribution, nous considérons un cadre multilingue où nous nous concentrons sur les langues à faibles ressources dans lesquelles il n'y a pas ou peu de données annotées disponibles. Tout d'abord, nous présentons le premier corpus de langage injurieux en persan, composé de 6 000 messages de micro-blogs provenant de Twitter, afin d'étudier la détection du langage injurieux. Après avoir annoté le corpus, nous réalisons étudions les performances des modèles de langages pré-entraînés monolingues et multilingues basés sur des transformeurs (par exemple, ParsBERT, mBERT, XLM-RoBERTa) dans la tâche en aval. De plus, nous proposons un modèle d'ensemble pour améliorer la performance de notre modèle. Enfin, nous étendons notre étude à un problème d'apprentissage multilingue de type few-shot, où nous disposons de quelques données annotées dans la langue cible. Nous adaptons une approche basée sur le méta-apprentissage pour étudier le problème de la détection des discours de haine et de langage offensant de type few-shot dans les langues à faibles ressources, qui permettra de prédire le contenu haineux ou offensant en n'observant que quelques éléments de données étiquetés dans une langue cible spécifique. Pour évaluer les modèles proposés, nous utilisons diverses collections de différents corpus accessibles au public, comprenant 15 datasets dans 8 langues pour le discours de haine et 6 datasets dans 6 langues pour langage offensant. Au meilleur de la connaissance de l'auteur, il y a eu un nombre insignifiant de tentatives d'utilisation des modèles de méta-apprentissage sur les tâches de détection des discours de haine.

Title : Hate Speech and Offensive Language Detection using Transfer Learning Approaches

Keywords : Hate Speech, Offensive Language, Transfer Learning, BERT, XLM-RoBERTa, Deep Learnig, Cross Lingual Text Classification, Few-shot Learning, Meta Learning, Social Media, Twitter

Abstract : The great promise of social media platforms (e.g., Twitter and Facebook) is to provide a safe place for users to communicate their opinions and share information. However, concerns are growing that they enable abusive behaviors, e.g., threatening or harassing other users, cyberbullying, hate speech, racial and sexual discrimination, as well. In this thesis, we focus on hate speech as one of the most concerning phenomena in online social media.

Given the high progression of online hate speech and its severe negative effects, institutions, social media platforms, and researchers have been trying to react as quickly as possible. The recent advancements in Natural Language Processing (NLP) and Machine Learning (ML) algorithms can be adapted to develop automatic methods for hate speech detection in this area.

The aim of this thesis is to investigate the problem of hate speech and offensive language detection in social media, where we define hate speech as any communication criticizing a person or a group based on some characteristics, e.g., gender, sexual orientation, nationality, religion, race. We propose different approaches in which we adapt advanced Transfer Learning (TL) models and NLP techniques to detect hate speech and offensive content automatically, in a monolingual and multilingual fashion.

In the first contribution, we only focus on English language. Firstly, we analyze user-generated textual content in Facebook to gain a brief insight into the type of content by introducing a new framework being able to categorize contents in terms of topical similarity based on different features, namely lexical, topical, and semantical features. Furthermore, using the Perspective API from Google, we measure and analyze the toxicity of the content. Secondly, we propose a TL approach for identification of hate speech by employing a combination of the unsupervised pre-trained model BERT (Bidirectional Encoder Representations from Transformers) and new supervised fine-tuning strategies. Finally, we investigate the effect of unintended bias in our pre-trained BERT-based model and propose a new generalization mechanism in training data by reweighting samples and then changing the fine-tuning strategies in terms of the loss function to mitigate the racial bias propagated through the model. To evaluate the proposed models, we use three publicly available datasets from Twitter.

In the second contribution, we consider a multilingual setting where we focus on low-resource languages in which there is no or few labeled data available. First, we present the first corpus of Persian offensive language consisting of 6 000 microblogs from Twitter to deal with offensive language detection in Persian as a low-resource language in this domain. After annotating the corpus, we perform extensive experiments to investigate the performance of transformer-based monolingual and multilingual pre-trained language models (e.g., ParsBERT, mBERT, XLM-RoBERTa) in the downstream task. Furthermore, we propose an ensemble model to boost the performance of our model. Then, we expand our study into a cross-lingual few-shot learning problem and we adapt a meta learning-based approach to study the problem of few-shot hate speech and offensive language detection in low-resource languages that will allow hateful or offensive content to be predicted by only observing a few labeled data items in a specific target language. To evaluate the proposed model, we use diverse collections of different publicly available corpora, comprising 15 datasets across 8 languages for hate speech and 6 datasets across 6 languages for offensive language. To the best of the author's knowledge, there has been an insignificant number of attempts to use meta learning approaches on hate speech detection tasks.

Institut Polytechnique de Paris
91120 Palaiseau, France