Fake identity & fake activity detection in online social networks based on transfer learning

Koosha Zarei

INSTITUT POLYTECHNIQUE DE PARIS

NNT : 2022IPPAS008

Thèse de doctorat

Fake Identity & Fake Activity Detection in Online Social Networks based on Transfer Learning

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

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

Thèse présentée et soutenue à Évry, le 12 juillet 2022, par

KOOSHA ZAREI

Composition du Jury :

Luis Muñoz Gutiérrez Professor, University of Cantabria - Spain	Examinateur/Président
Xiaoming Fu Professor, University of Göttingen - Germany	Rapporteur
Elena Cabrio Professor, Université Côte d'Azur - France	Rapporteur
Bruce MacDowell Maggs Professor, Duke University - USA	Examinateur
Albert Meroño Peñuela Assistant Professor, King's College London - UK	Examinateur
Christophe Cerisara Researcher, CNRS - LORIA laboratory - France	Examinateur
Noel Crespi Professor, IP-Paris, Telecom SudParis - France	Directeur de thèse
Reza Farahbakhsh Adjunct Assistant Professor, IP-Paris, Telecom SudParis - France	Co-superviseur

Doctor of Philosophy (PhD) Thesis
Institut Polytechnique de Paris (IP-Paris)

Specialization

Computer Science
Artificial Intelligence

presented by

Koosha Zarei

Fake Identity & Fake Activity Detection in Online Social Networks based on Transfer Learning

Committee:

Xiaoming Fu	Reviewer	Professor, University of Göttingen - Germany
Elena Cabrio	Reviewer	Professor, Université Côte d'Azur - France
Bruce MacDowell Maggs	Examiner	Professor, Duke University - USA
Albert Meroño Peñuela	Examiner	Assistant Professor, King's College London - UK
Christophe Cerisara	Examiner	Researcher, CNRS - LORIA laboratory - France
Luis Muñoz Gutiérrez	Examiner	Professor, University of Cantabria - Spain
Noel Crespi	Advisor	Professor, IP-Paris, Telecom SudParis - France
Reza Farahbakhsh	Co-supervisor	Assistant Professor, IP-Paris, Telecom SudParis - France

Dedication

To
all my family, the symbol of love and giving.

Acknowledgements

First, I would like to express my deepest appreciation 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 honour for me.

I would like to thank my co-supervisor, Dr. Reza Farahbakhsh, for his technical support, motivation, friendship, and productive discussions 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 guidance in every situation.

I would like to extend my sincere thanks to my thesis reviewers, Prof. Xiaoming Fu and Prof. Elena Cabrio who patiently read this dissertation and provided invaluable comments and suggestions. A special thanks to Prof. Bruce Maggs, Dr. Albert Meroño Peñuela, Dr. Christophe Cerisara, and Prof. Luis Muñoz Gutiérrez for being part of my jury as examiners for my thesis defence.

My special thanks to all the lovely team members of the Data Intelligence and Communication Engineering Lab at TSP, especially Praboda, Samin, Faraz, and Yasir for the wonderful times we shared. You were always there with a word of encouragement. I also had the great pleasure of working with Prof. Roberto Minerva from whom I learned new ways of thinking and how to best contribute to a project. He is a great friend. Special thanks go to the wonderful administrative staff in TSP, Valerie Mateus and Veronique Guy.

My profound love, respect and thank go to my family whom I owe a great deal. I deeply thank my parents, Mahmood and Shohreh, for their unrequited love, unconditional trust, timely encouragement, and endless patience. It was their love that empowered me to break my limits and experience life freely and fearlessly. I would like to thank other family members, Mani, Nima, and Shabnam for their generous and endless support. I could not be able to finish this work without your support and I am so happy to have you. My biggest thanks to my wife, Nafiseh, whom I am so lucky to have in my life. When I struggled during my research, she always encouraged me to keep trying. Nafiseh, you are my best friend, my confidant, and my support, and there are no words to express my gratitude for having you in my life.

Koosha Zarei
28^{th} May 2022

Abstract

While Social Media has connected more people around the world and has increased the ease of access to free content, but is dealing with critical phenomena such as fake content, fake identities, and fake activities. Fake content detection on social media has recently become emerging research that is attracting tremendous attention. In this area, fake identities are playing an important role in the production and propagation of fake content on Online Social Networks such as Meta (Facebook), Twitter, and Instagram. The main reason behind this is that social media encourages impersonators, malicious accounts, trolls, and social bots to produce content and interact with humans or other bots without considering the credibility of the content and entice users to click and share them.

In this thesis, I primarily concentrate on impersonators as one of the concerning varieties of fake identities. These entities are nefarious fake accounts that intend to disguise a legitimate account by making similar profiles and then striking social media with fake content, which makes it considerably harder to understand which posts are genuinely produced. The recent advancements in Natural Language Processing (NLP), and Transformer-based Language Models (LM) can be adapted to develop automatic methods for many related NLP downstream tasks in this area. Language Models and their flexibility to cope with any corpus delivering great results has made this approach very popular. The fake content classification can be handled using Pretrained Language Models (PLM) and accurate deep learning models.

The aim of this thesis is to investigate the problem of fake identities, fake activities, and their generated ingenuine content in social media and propose algorithms to classify fake content. We define fake content as verifiably false pieces of information shared intentionally to mislead the readers. I propose different approaches in which I adapt advanced Transfer Learning (TL) models and NLP techniques to detect fake identities and classify fake content automatically.

In this thesis, (1) first, I assemble several novel datasets containing the content and activities of fake and genuine identities in several communities on Instagram and Twitter. A dedicated crawler has been developed in order to receive publicly available data concerning GDPR regulations. I use these datasets for various research in line with the subject of this thesis. In addition, some datasets have been published for the research community.

Next, (2) I present a practical approach to detect impersonators as fake identities and cluster their generated content based on profile characteristics and user behaviours. Meanwhile, I propose a Deep Neural Network architecture in order to detect impersonator-generated posts and genuine content on Social Media. Next, I investigate the content, behaviours, and activities of impersonators.

Eventually, (3) I leveraged RoBERTa to propose a pretrained transformer-based language model, called FakeRoBERTaSM, which is pretrained from scratch and optimized for social media textual data to overcome "*informal English-language textual*" challenges. Meanwhile, to handle "*unknown tokens*" on daily conversations on social media, I use the

Character CNN model which is a character-level tokenization technique. Next, I propose a fine-tuned and multi-domain deep learning architecture that is optimized for fake content classification on social media. The experimental results show that the deep model architecture trained with FakeRoBERTaSM embedding performed better than the remaining baseline models considered in my analyses.

Keywords

Fake Identities, Fake Content, Impersonators, Transformers, Pretrained Language Model, BERT, RoBERTa, NLP, Text Classification, Deep Learning, Social Media, Instagram

Résumé

Les médias sociaux ont permis de connecter un plus grand nombre de personnes dans le monde entier et d'accroître la facilité d'accès au contenu gratuit. d'accès à des contenus gratuits, mais ils sont confrontés à des phénomènes critiques tels que les faux contenus, les fausses identités et les fausses activités. La détection de faux contenus sur les médias sociaux est récemment devenue une recherche émergente qui attire une attention considérable. une recherche émergente qui suscite une attention considérable. Dans ce domaine, les fausses identités jouent un rôle important dans la production et la propagation de faux contenus dans les réseaux sociaux en ligne réseaux sociaux en ligne tels que Meta (Facebook), Twitter et Instagram. La principale raison de ce phénomène est que les médias sociaux encouragent les usurpateurs d'identité, les comptes malveillants, les trolls et les robots sociaux àproduire du contenu et interagir avec des humains ou d'autres robots sans tenir compte de la crédibilité du contenu. La détection de faux contenus sur les médias sociaux est récemment devenue une recherche émergente qui attire une attention considérable. une recherche émergente qui suscite une attention considérable. Dans ce domaine, les fausses identités jouent un rôle important dans la production et la propagation de faux contenus dans les réseaux sociaux en ligne réseaux sociaux en ligne tels que Meta (Facebook), Twitter et Instagram. La principale raison de ce phénomène est que les médias sociaux encouragent les usurpateurs d'identité, les comptes malveillants, les trolls et les robots sociaux à produire du contenu et interagir avec des humains ou d'autres bots sans tenir compte de la crédibilité du contenu et inciter les utilisateurs à cliquer et à les partager. L'objectif de cette thèse est d'étudier le problème des fausses identités, des fausses activités et du contenu authentique qu'elles génèrent dans les médias sociaux et de proposer des algorithmes pour classifier le contenu factice. Nous définissons le faux contenu comme un élément d'information vérifiable et faux partagé intentionnellement pour tromper les lecteurs. Je propose différentes approches dans lesquelles j'adapte des modèles avancés de Transfer Learning (TL) et des techniques NLP pour détecter les fausses identités et classer le faux contenu automatiquement.

Mots-clés

Fausses identités, Fausses données, imposteurs, transformateurs, modèle de langage préentraîné, BERT, RoBERTa, NLP, Classification de texte, Deep Learning, Réseaux sociaux, Instagram

Table of contents

1 Introduction **17**
- 1.1 Motivation . . . 18
  - 1.1.1 Fake Content on Social Media . . . 18
  - 1.1.2 Fake Identities & Impersonation on Social Media . . . 18
  - 1.1.3 Fake Content Language Modeling . . . 19
  - 1.1.4 Social Media Aware Language Modeling . . . 19
- 1.2 Objectives and Contributions of the Thesis . . . 20
- 1.3 Publications List . . . 21
- 1.4 Relationship of Publications with Contributions . . . 22
- 1.5 Outline of the Thesis . . . 23
- 1.6 Ethical Considerations . . . 23

2 Background and Related Technologies **25**
- 2.1 Overview . . . 26
- 2.2 Fake Content on Social Media . . . 26
  - 2.2.1 Fake Accounts . . . 27
  - 2.2.2 Fake Engagement . . . 27
  - 2.2.3 User Behaviour . . . 27
- 2.3 Fake Content detection on Social Media . . . 27
- 2.4 Transfer Learning for fake Content detection . . . 28
- 2.5 Contextual Language Modeling . . . 29
- 2.6 Summary and Conclusion . . . 29

3 Social Media Data Collection & Analysis **31**
- 3.1 Overview . . . 33
- 3.2 Crawler . . . 33
  - 3.2.1 Architecture . . . 33
- 3.3 Impersonators dataset . . . 34
  - 3.3.1 Data Collection . . . 34
  - 3.3.2 Data Validation . . . 35
  - 3.3.3 Data Pre-processing . . . 36

3.3.4 Challenges & Limitations . . . 36
3.3.5 Dataset Usage . . . 36
3.3.6 Ethics . . . 37
3.4 Influencer dataset . . . 37
3.4.1 Data Collection . . . 37
3.4.2 Data Validation . . . 38
3.4.3 Characterising Influencers . . . 38
3.4.4 Characterising Reactions . . . 40
3.4.5 How often do influencers post? . . . 43
3.4.6 What do influencers promote? . . . 44
3.4.7 Dataset Usage . . . 45
3.4.8 Ethics . . . 45
3.5 COVID_19 dataset . . . 45
3.5.1 Data Collection . . . 46
3.5.2 Limitations . . . 47
3.5.3 Data Summary . . . 47
3.5.4 Characterising Publishers . . . 49
3.5.5 Characterising Hashtags . . . 53
3.5.6 Dataset Usage . . . 55
3.5.7 Access To Dataset . . . 55
3.5.8 Ethics . . . 55
3.6 Conclusion . . . 55

4 Impersonator: Fake identities & Ingenuine Content in Social Media 57
4.1 Overview . . . 59
4.2 Related Work . . . 59
4.2.1 User Behaviour . . . 60
4.2.2 Fake account . . . 60
4.2.3 Bot generated content . . . 61
4.3 Definition . . . 61
4.3.1 Bots . . . 61
4.3.2 Political Bots . . . 61
4.3.3 Impersonator or Imposter . . . 61
4.3.4 Impersonation and Social Media Profile Theft (SMPT) . . . 61
4.3.5 Profile Similarity . . . 62
4.3.6 Types of Impersonators . . . 63
4.4 Case Study Accounts . . . 64
4.5 Data Collection & Data Pre-Processing . . . 64
4.6 Identification of Impersonating Accounts . . . 64
4.6.1 Identifying Impersonators . . . 64
4.6.2 Primary Account Analysis . . . 66
4.6.3 Clustering . . . 66
4.6.3.1 Impersonator Bots - Cluster 0 . . . 68
4.6.3.2 Impersonator Fan Pages - Cluster 1 . . . 68

4.6.4 Manual inspection for validation. . . . 68
4.7 A Deep Neural Approach . . . 68
4.7.1 Dataset Overview . . . 69
4.7.2 Over-Sampling . . . 69
4.7.3 Feature Engineering . . . 69
4.7.4 Proposed DNN Architecture . . . 70
4.7.5 Feature Analysis . . . 72
4.8 Assessing Published Content . . . 72
4.8.1 Politicians . . . 74
4.8.2 Sports Players. . . . 74
4.8.3 Musicians . . . 75
4.9 Conclusion . . . 76

5 Multi-Domain & Social Media Aware Language Modeling for Fake Content 79
5.1 Overview . . . 81
5.2 Background & Related Work . . . 82
5.2.1 Multi-domain learning . . . 82
5.2.2 Transfer Learning for fake Content detection . . . 83
5.2.3 Contextual Language Modeling . . . 84
5.2.3.1 Google BERT . . . 84
5.2.3.2 ROBERTA . . . 85
5.2.4 Multi-Domain Adaptation and Language Modeling . . . 85
5.2.5 Social Media Context Aware Model . . . 86
5.3 Fake RoBERTa for Social Media . . . 87
5.3.1 Architecture . . . 87
5.3.2 Character-Level Tokenization . . . 89
5.3.3 Data Corpus & preparation . . . 90
5.4 Fake Content Classification using FakeRoBERTaSM . . . 92
5.4.1 Methodology . . . 92
5.4.2 PreTraining Procedure . . . 93
5.5 Evaluation Setup . . . 95
5.5.1 List of Models . . . 95
5.5.2 Model Configuration . . . 96
5.5.3 Evaluation Metrics . . . 96
5.6 Experiments & Results . . . 96
5.6.1 Evaluation Datasets . . . 96
5.6.2 Performance Analysis . . . 97
5.7 Discussion & Conclusion . . . 98

6 Conclusion and Future Work 99
6.1 Conclusion . . . 100
6.1.1 Summary and Insights of Contributions . . . 100
6.1.1.1 Social Media Content Analysis & Datasets . . . 100

6.1.1.2 Impersonation on Social Media 101
6.1.1.3 Social Media Aware Language Modeling 102
6.2 Future Work and Challenges 102

References **103**

List of figures **113**

List of tables **115**

Chapter 1

Introduction

Contents

1.1 Motivation **18**
1.1.1 Fake Content on Social Media 18
1.1.2 Fake Identities & Impersonation on Social Media 18
1.1.3 Fake Content Language Modeling 19
1.1.4 Social Media Aware Language Modeling 19
1.2 Objectives and Contributions of the Thesis **20**
1.3 Publications List **21**
1.4 Relationship of Publications with Contributions **22**
1.5 Outline of the Thesis **23**
1.6 Ethical Considerations **23**

1.1 Motivation

1.1.1 Fake Content on Social Media

Fake content can be defined as a verifiably false piece of information shared intentionally to mislead readers [1] and has been used to create a political, social, and economic bias in the minds of people for personal gains. One main reason of disseminating many fake content on social media is that they often encourage impersonators, malicious accounts, trolls, and social bots to produce information [2] [3] without considering the credibility of the content as an attempt to entice users to read them [4].

Compared with traditional news, fake news attract readers and get rapid dissemination causing large-scale negative effects. The best example for this is that within the first three months of the USA presidential election 2016, fake news generated to favor both nominees was believed and shared by almost 37 million social media users [5]. Since social media content is relayed among users without filtering, editorial judgment, or fact-checking, it is required to introduce highly efficient models to detect fake content with high accuracy to control the spread of fake content on internet platforms. Due to the above reasons, Fake content detection on social media has recently become an active area of research.

However, Fake content detection on social media is really challenging as they are inherently multilingual and in multiple forms such as textual, visual, and auditory forms. The lack of labeled data is another major challenge in exploring fake content on social media especially when using traditional machine learning-based models and algorithms. In addition, social media platforms have their own characteristics in terms of data types, user relations, user behaviors, and linguistic differences and which require special attention when handled at once. Furthermore, social media permit users to share information on a variety of topics such as memes, events, politics, health, and celebrities.

1.1.2 Fake Identities & Impersonation on Social Media

Currently, social networking websites do not provide any notification to their users about profile authenticity [6]. Many threats, such as cloning of profile information and monitoring of the user's activity and others, also increase—privacy of the users' data being a sensitive issue as it introduces many cybercrimes. In the past few years, researchers have developed many models to address these issues, but, still, the problem remains open.

Impersonators are playing an important role in the production and propagation of the content on Online Social Networks, notably on Instagram. These entities are nefarious fake accounts that intend to disguise a legitimate account by making similar profiles and then striking social media by fake content, which makes it considerably harder to understand which posts are genuinely produced.

Although many impersonators may be innocuous, there also exists malicious fake accounts. These often have clear plans, where they make accounts appear more popular than they are, produce pre-planned untrustworthy content, perform brand abuse or generate fake engagement [7]. Therefore several lawsuits have taken place in the United State (along with other countries), where criminal impersonation is a crime. It involves assuming a false identity with the intent to defraud another or pretending to be a representative of another person or organisation [8].

However, identifying such activities is often slow and laborious — hence, developing techniques for automated detection would have real value to social media companies.

1.1.3 Fake Content Language Modeling

The key step in detecting fake content on social media is to understand the textual data. In this area, Language Modeling (LM) is considered a central task of language understanding and language processing [9]. In contrast to traditional context-free text embedding techniques, transformer-based Pretrained Language Models (PLMs) use much deeper network architectures [10], and are pre-trained on much larger text corpora to learn contextual text representations. So, textual data becomes more meaningful through a deeper understanding of its context, which in turn facilitates text analysis and mining.

Bidirectional Encoder Representations from Transformers (BERT) is one of the first tranformer-based PLMs that has achieved state-of-the-art results in a broad range of NLP tasks [11]. BERT and other BERT-based transformers (*e.g.* RoBERTa, DistilBERT) are designed to pretrain deep bidirectional representations from unlabeled text and, then, be fine-tuned for downstream tasks [12].

So, proper language modeling for textual data on social media could significantly increase the accuracy of fake content detection.

1.1.4 Social Media Aware Language Modeling

Nowadays, we see a considerable linguistic differences between the language spoken on social media (*e.g.* daily conversation, Tweets, comments) and formal corpora (*e.g.* books, Wikipedia). Misspelling, new vocabularies, abbreviations, slang, *etc.* are some examples that could impose an impact on downstream NLP tasks. A major problem with statistical language models was the inability to deal well with synonyms or Out-of-Vocabulary (OOV) words that were not present in the training corpus [13]. However, proper word representation in transformers is an important step in order to process textual data. While various distributed word representations exist, few are capable of handling OOVs especially in daily conversations in social media.

In addition, most PLMs are trained on general-domain text corpora (*e.g.* Wikipedia, Books) [12]. If the target domain is completely different from the general domain, the final task result could be poor. In this situation, we might consider adapting the PLM using domain-specific data. However, PLMs can be pretrained with multi-domain topics to increase the target task accuracy. For example, to address COVID-19-related textual challenges, a PLM can be adapted on medical literature.

The language model that considers low-level features of the informal daily conversation on social media, could increase the accuracy of fake content detection.

1.2 Objectives and Contributions of the Thesis

In this section, I outline the main objectives of this thesis in which each objective is represented as one contribution. The aim of this thesis is to investigate the problem of fake identities, fake activities, and their generated content in social media and propose algorithms to classify fake content using transfer learning. The main objectives to achieve this aim are as follows:

- To assemble novel datasets containing the content and activities of fake identities and fake content in various communities in social media.
- To present a practical approach to identify impersonators as one of the important type of the fake identities in social media.
- To cluster impersonator accounts based on profile characteristics and user behaviours, and present a full investigation of the ingenuine content generated by impersonators.
- To propose a pretrained Transformer-based Language Model which is pretrained from scratch and is optimized for social media textual content.
- To present a fine-tuned and multi-domain Deep Learning neural network model that is optimized for fake content classification on social media.

My approach to achieve the above research objectives is organized into several contributions as follows:

C1: As the main objective of this thesis is about fake content and fake identities in social media platforms, I set the first contribution to (1) first develop a dedicated crawler in order to collect data with respect to GDPR from social media, and (2) provide the proper datasets for further analysis. The implemented crawler is used in order to gather data on the following domains from Instagram: Impersonators and their

activities, Influencers and distributed sponsored content, and the COVID_19 related content. Some of the mentioned datasets have been released for research purposes.

C2: The second contribution is about detecting impersonators as an important type of fake identities in social media platforms. This contribution aims to (1) detect impersonators, (2) analyse their behaviours in different communities, and (3) develop a machine learning model to identify ingenuine content automatically (impersonator-generated content). Impersonation is where (sometimes malicious) users create social media accounts mimicking a legitimate account. Such impersonators are found on all major social media platforms such as Facebook, Twitter, Instagram, YouTube and LinkedIn. Among these platforms, Instagram is widely used by celebrities, influencers, businesses, and public figures with different levels of popularity.

C3: The third contribution focuses on the context-aware language modeling of the textual content on social media platforms. The fake content classification task is one of the greatest challenges of researchers on Online Social Networks that could be addressed using Pretrained Language Models (PLM). This contribution aims to: (1) propose a pretrained transformer-based language model, called FakeRoBERTaSM, which is pretrained from scratch and optimized for social media textual data to overcome "*informal English-language textual*" challenges. (2) Address "*unknown tokens*" challenge on daily conversations on social media by using Character CNN which is a character-level tokenization technique. And (3) propose a fine-tuned multi-domain deep learning architecture that is optimized for fake content classification on social media. The experimental results shows that the deep model architecture trained with FakeRoBERTaSM embedding performed better than the remaining baseline models considered.

1.3 Publications List

Journal Papers

- K. Zarei, R. Farahbakhsh, N. Crespi and G. Tyson, "Dataset of Coronavirus Content From Instagram With an Exploratory Analysis," in IEEE Access, vol. 9, pp. 157192-157202, 2021, doi: 10.1109/ACCESS.2021.3126552. [URL]

Conference Papers

- Koosha Zarei, D. Ibosiola, R. Farahbakhsh, Z. Gilani, K. Garimella, N. Crespi, G. Tyson. Characterising and Detecting Sponsored Influencer Posts on Instagram. In 2020 ACM/IEEE ASONAM, 2020. [URL]

- Koosha. Zarei, R. Farahbakhsh, N. Crespi and G. Tyson. Impersonation on Social Media: A Deep Neural Approach to Identify Ingenuine Content. In 2020 ACM/IEEE ASONAM, 2020. [URL]
- Koosha. Zarei, R. Farahbakhsh, and N. Crespi. How impersonators exploit instagram to generate fake engagement? In ICC 2020, pages 1–6, 2020. [URL]
- Koosha. Zarei, R. Farahbakhsh, and N. Crespi. Typification of impersonated accounts on instagram. In 2019 IEEE 38th IPCCC, pages 1–6, 2019. [URL]
- Koosha. Zarei, R. Farahbakhsh, and N. Crespi. Deep dive on politician impersonating accounts in social media. In 2019 ISCC, pages 1–6, 2019. [URL]

Dataset

1. Koosha Zarei, R. Farahbakhsh, N. Crespi, G. Tyson. A First Instagram Dataset on COVID-19. arXiv, 2020. [URL]

Under Review

- Koosha Zarei, P. Rajapaksha, R. Farahbakhsh, N. Crespi, G. Tyson, "Multi-Domain and Social Media Aware Language Model Adaptation for Fake Content", IEEE Access 2022.

1.4 Relationship of Publications with Contributions

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

- The publications 'How impersonators exploit instagram to generate fake engagement', 'A First Instagram Dataset on COVID-19', 'Characterising and Detecting Sponsored Influencer Posts on Instagram', and 'Dataset of Coronavirus Content From Instagram With an Exploratory Analysis' correspond to Contribution C1 in Chapter 3.
- The publications 'Impersonation on Social Media: A Deep Neural Approach to Identify Ingenuine Conte', 'Deep dive on politician impersonating accounts in social media', 'Typification of impersonated accounts on instagram', and 'How impersonators exploit instagram to generate fake engagement?' correspond to Contributions C2 in Chapter 4.
- The submitted paper 'Multi-Domain and Social Media Aware Language Model Adaptation for Fake Content' and 'How impersonators exploit Instagram to generate fake engagement?' correspond to Contributions C3 in Chapter 5.

1.5 Outline of the Thesis

The thesis is structured into following 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., fake content definition and detection, natural language processing techniques, transfer learning, language modeling, and multi-domain language modeling.
- Chapter 3 presents the process of data collection and the architecture of the implemented crawler. In addition, three different datasets are discussed in detail. These datasets are used in this thesis.
- Chapter 4 presents the fake identity detection methodology on social media. In particular, the challenge of impersonation is discussed in detail. Several well-known politician, sports stars, and celebrities in different leading communities of Instagram are selected to be analysed.
- Chapter 5 presents the pretrained transormer-based language model, called FakeRoBERTaSM, which is pretrained from scratch and optimized for social media textual data to overcome "*informal English-language textual*" challenges. This chapter is divided into: i) multi-domain language modeling based on social media textual data, and ii) social media aware language modeling for fake content detection.
- Chapter 6 summarizes the thesis and provides an outlook into the future.

1.6 Ethical Considerations

Regarding General Data Protection Regulation (GDPR) compliance, to respect privacy and ethical aspects of users on social media, I did not collect any sensitive and personal information of users from social media platforms. I only collected publicly available data from Twitter and Instagram 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 **26**
2.2 Fake Content on Social Media **26**
2.2.1 Fake Accounts 27
2.2.2 Fake Engagement 27
2.2.3 User Behaviour 27
2.3 Fake Content detection on Social Media **27**
2.4 Transfer Learning for fake Content detection **28**
2.5 Contextual Language Modeling **29**
2.6 Summary and Conclusion **29**

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 Fake Content on Social Media

Fake content can be defined as a verifiably false piece of information shared intentionally to mislead the readers [1]. Fake content detection on social media has recently become emerging research that is attracting tremendous attention. The main reason behind this is that social media encourages impersonators, malicious accounts, trolls, and social bots to produce content and interact with human or other bots in social media [2] [3] without considering the credibility of the content and entice users to click and share them [4].

Social media users quickly believe in fake news content due to some psychological factors such as consensus (user believes if others also believe in), consistency (if the content favors his own beliefs), and popularity (users tend to trust more popular content, but its popularity might be driven by social bots) [14]. Therefore, fake content detection on social media is really challenging and presents unique characteristics that make traditional machine learning based models and algorithms ineffective. Apart from that, social media has its challenges in terms of data types, user relations, user behaviours, linguistic differences, and *etc.* which requires special attention. In addition, lack of labelled data is another major challenge in exploring fake content on social media mainly for the traditional machine learning models.

Fake content has been used to create a political, social, and economic bias in the minds of people for personal gains. It aims at exploiting people by creating fake content that sounds legit [15]. Thus, it is extremely important to detect and control the spread of fake content on internet platforms. Compared with traditional news, fake news attract readers and get rapid dissemination causing large-scale negative effects. The best example for this is that within the first three months of the USA presidential election 2016, fake news generated to favor both nominees were believed and shared by almost 37 million social media users [5]. Since social media content relayed among users without filtering, editorial judgement or fact-checking, it is required to introduce highly efficient models to detect fake content with high accuracy.

2.2.1 Fake Accounts

Several studies tried to shed light on this direction by profiling users based on their activities and reactions. This work [16] presents a novel technique to discriminate real accounts on social networks from fake ones. The writers from this [17] study provide a review of existing and state-of-the-art Sybil detection methods with an introductory approach and present some of the emerging open issues for Sybil detection in Online Social Networks.

On the other hand, the huge existence of Bots can alter the perception of social media influence, artificially enlarging the audience of some people, or they can impact the reputation of a company. The problem of rising social bots are discussed in [18]. There are various strategies to tackle the problem of bot detection. [19] suggested a profile-based approach and [20] proposed a novel framework on detecting spam content. Also, [21] presented a machine learning pipeline for detecting fake accounts and authors in [22, 23] present a method to classify bots and understand their behaviour in scale.

2.2.2 Fake Engagement

From this viewpoint, Authors in [24], focus on the social site of YouTube and the problem of identifying bad actors posting inorganic contents and inflating the count of social engagement metrics. They propose an effective method and show how fake engagement activities on YouTube can be tracked over time. Likewise, another study, [25], enumerate the potential factors which contribute towards a genuine like on Instagram. Based on analysis of liking behaviour, they build an automated mechanism to detect fake likes on Instagram which achieves a high precision of 83.5

2.2.3 User Behaviour

On another line of research, the authors in [26] [27] look at the profile and behavioural patterns of a user and discussed existing challenges on different OSNs. By integrating semantic similarity and existing relationships between users, it is possible to match profiles across various OSNs [28] [29]. Also, [30] conducted a detailed investigation of user profiles and proposed a matching scheme. On Instagram, for the sake of mitigating impersonation attack, [25] explored fake behaviours and built an automated mechanism to detect fake activities.

2.3 Fake Content detection on Social Media

Previous research tried to understand fake news on social media through general lexical features such as lexicon, syntax, discourse, semantic, POS tags and probabilistic context-

free grammar [31] [32]. These research works mainly used text embedding methods at the word level, sentence level, and document level to represent news items in vector formats in order to feed them to traditional machine learning algorithms [33]. Later, researchers experimented with deep learning models such as Neural Networks and Transformers to extract latent textual content from fake news [34] [35] [36]. For example, Wani, et al. look at automated techniques for fake news detection from a data mining perspective [15]. They evaluate the importance of unsupervised learning in the form of language model pre-training and distributed word representations using unlabelled Covid-19 tweets corpus.

2.4 Transfer Learning for fake Content detection

There have been several attempts to apply transfer learning to fake news detection. Santiago González-Carvajal et al. studied the general comparison between BERT against traditional machine learning classification in [37]. They differentiate types of approaching NLP problems into two categories: a linguistic approach that generally uses different features of the text, and a machine/deep learning approach. From tokenizing perspective, [38] highlighted various challenges in the BERT model which if solved could significantly boost the model's accuracy, especially in domain-specific applications. With the advancement of Transformers in NLP tasks as reviewed in the above-mentioned research works, several recent works have used Transfer Learning for fake news detection.

Liu et al. [39] proposed a BERT-based method for fake news detection. They treated fake news detection as a fine-grained multiple-classification task and their model exhibited superior performance to the baselines and other competitive approaches. A data-driven BERT-based automatic fake news detection method was proposed by Heejung et al. [40]. This model analyzed the relationship between the headline and the body text of news. CT-BERT model was introduced in [41] which proposed an approach using the BERT-based ensemble model focused on COVID-19 fake news detection. Xiangyang et al. [42] have also proposed ensemble method of different pre-trained language models such as BERT, Roberta and Ernie, targeting COVID-19 fake news detection. FakeBERT [43] is a BERT-based deep learning approach that integrates a deep convolution neural network having different kernel sizes and filters with the BERT. Their proposed model outperformed many other existing models with 98.9% accuracy. Khan et al. [44] conducted a benchmark study to assess performance 19 different machine learning models for fake news detection. Their experimental results show that BERT-based models have achieved better performance than all other models across datasets. For interested readers, Rogers et al. [45] reviewed how BERT works, what kind of information it learns and how it is represented, common modifications to its training objectives and architecture.

As BERT-based deep learning models perform better than many baseline models on fake news classification, in this thesis, I enhance Transfer Learning-based models on both pre-training and fine-tuning to improve fake content classification metrics.

2.5 Contextual Language Modeling

Language modeling is the task of predicting the next word or character in a document. A wide range of natural language processing tasks uses language models for text generation, text classification, and question answering. Transformer based models such as GPT3 and BERT that are pre-trained on a large corpus of dataset are popular and notable state-of-the-art language models. In general, pre-trained representations can either be context-free or contextual. Traditional context-free models such as TF-IDF [37, 46] and word2vec [47, 48] generate a single word embedding representation for each word in the vocabulary. For example, the word "bank" would have the same context-free representation in "bank account" and "bank of the river." Contextual models instead generate a representation of each word that is based on the other words in the sentence. As classification gives more promising results incorporating contextual information, this thesis mainly focuses on context aware language models.

Since 2018 [12], we have seen the rise of a set of large-scale Transformer-based Pre-trained Language Models (PLMs) in the domain of NLP. Transformer-based models use deeper network architectures (e.g., 48-layer Transformers [144]), and are pre-trained on much larger text corpora to learn contextual text [11]. Contextual representations can further be unidirectional or bidirectional. For example, in the sentence "I accessed the bank account," a unidirectional contextual model would represent "bank" based on "I accessed the" but not "account." However, BERT represents "bank" using both its previous and next context — "I accessed the ... account" — starting from the very bottom of a deep neural network, making it deeply bidirectional.

Transformer-based PLMs such as BERT, RoBERTa, or DistilBERT are pre-trained on large corpora and can be fine-tuned to solve many NLP tasks [12]. During pre-training, the model is trained on unlabeled textual data which is an unsupervised (or self-supervised) task, and in the fine-tuning part, the pre-trained parameters are fine-tuned using labeled data (supervised task). With this technique, I get the word relationships in different context and use their weight to solve the downstream task. Below I describe base PLMs:

2.6 Summary and Conclusion

This chapter presented a general overview of the major topics relevant to this thesis. To summarize, it covered some major areas: First, it discussed the problem of Fake Content in

social media and provided its definition and the major ideas behind the solutions proposed. The activity of Fake Identities and their variants are also covered. Next, it discussed Transfer Learning and its progress in NLP and contextual language modelling with baseline variants. Next, multi-domain adaptation and social media context modelling is discussed in detail. I use these definitions, backgrounds, and technologies in this thesis in different research directions.

Chapter 3

Social Media Data Collection & Analysis

Contents

3.1 Overview **33**
3.2 Crawler **33**
3.2.1 Architecture 33
3.3 Impersonators dataset **34**
3.3.1 Data Collection 34
3.3.2 Data Validation 35
3.3.3 Data Pre-processing 36
3.3.4 Challenges & Limitations 36
3.3.5 Dataset Usage 36
3.3.6 Ethics 37
3.4 Influencer dataset **37**
3.4.1 Data Collection 37
3.4.2 Data Validation 38
3.4.3 Characterising Influencers 38
3.4.4 Characterising Reactions 40
3.4.5 How often do influencers post? 43
3.4.6 What do influencers promote? 44
3.4.7 Dataset Usage 45
3.4.8 Ethics 45
3.5 COVID_19 dataset **45**
3.5.1 Data Collection 46
3.5.2 Limitations 47

3.5.3 Data Summary 47
3.5.4 Characterising Publishers 49
3.5.5 Characterising Hashtags 53
3.5.6 Dataset Usage 55
3.5.7 Access To Dataset 55
3.5.8 Ethics 55
3.6 Conclusion 55

3.1 Overview

In this chapter, first, I present the process of data crawling from social media and the architecture of dedicated tool that I have implemented. I use this crawler in order to collect information from social media platforms in line with my research questions. Next, I describe three different datasets in detail: (*i*) Impersonator accounts, (*ii*) Influencer activities on Instagram, and (*iii*) COVID-19 content during the first lockdown. Each one is used in a separated research paper.

3.2 Crawler

In order to collect the Instagram public content, I develop a crawler that is able to handle various tasks simultaneously. This crawler connects to Instagram via multiple channels, downloads public data content concurrently, performs some NLP based pre-processing steps, and finally stores them in a NoSQL format database. In Instagram, a reaction to a post can be active (comment) or passive (like). The crawler relies on the official Instagram APIs described in [49]. To get the public content that is tagged with a specific hashtag or keyword, I use the Instagram Hashtag Engine which is available in [50]. This API returns public posts that have been tagged with particular hashtags. my crawler runs on several virtual machines in parallel 24/7. Note, that I do not manually filter any posts and therefore I gather all posts containing the hashtags, regardless of the specific topics discussed within.

3.2.1 Architecture

Figure 3.1 shows the complete architecture design of my crawler, which contains four different major parts to handle data crawling: (*i*) API Connection Layer, (*ii*) Proxy Layer, (*iii*) Main Body, and (*iv*) Database Layer. The process of receiving data is as follows:

1. The API Connection Layer (Block 1 in Figure 3.1) connects to the official Instagram platform [49], which is currently using the Graph API. The crawler is registered as an application to be able to perform user authentication [51]. Note, there are certain rate limitations for requesting information per hour [49].

2. Between the Connection Layer and the Main Module, the Proxy Layer (Block 2 in Figure 3.1) is responsible for handling multiple proxy IP addresses and creating multiple connection layers. This helps us to receive data at a faster rate from various IP addresses. Thus, these layers are working concurrently.

3. The main body of the crawler (Block 3 in Figure 3.1) contains several inner modules such as Post, Reaction, Profile, Social Connection, and Story or Live modules. These

are responsible for getting parts that are associated with their names. For example, the Post module is programmed to get Instagram posts and metadata. These modules are directly connected to a scheduler that handles time management. For example, checking daily stories, updating reactions, looking for new posts, checking highlights, and revising new social connections. Last, the *Pre-Processing layer* is used to perform some basic pre-processing steps such as text cleaning, data management, language extraction, *etc.*

4. In the Database Layer (Block 4 in Figure 3.1), I store my data. I use MongoDB as the primary database and I keep each module in a separate corresponding collection. For example, post content is stored in the *post collection*.

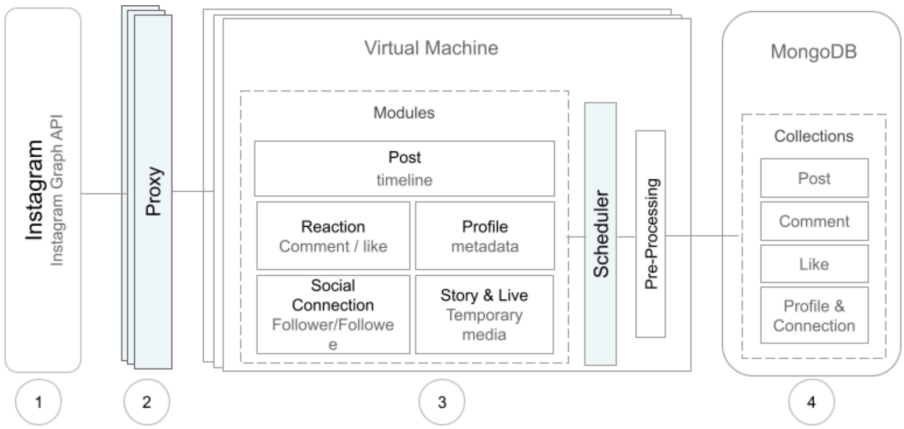


Figure 3.1: The architecture of the Instagram crawler.

3.3 Impersonators dataset

In this part, I target the problem of impersonation in Social Media and I crawl various information from Instagram.

3.3.1 Data Collection

The data collection involves several key steps as follow:

Genuine Accounts: As the first step, I collect posts of the 15 genuine case studies (Table 3.1) which are published between October 2018 and January 2020. Posts contain publicly available information including caption, hashtags, image/video, number of likes, number of comments, location, time, and tagged list. 1.3K posts across the three communities has been collected during the campaign. I use the crawler presented in [52].

Table 3.1: Use Cases and Corresponding Hashtags on Instagram

Politician		Sports Stars		Musician	
D. Trump	*#donaldtrump*	L. Messi	*#leomessi*	L. Gaga	*#ladygaga*
B. Obama	*#barackobama*	C. Ronaldo	*#cristianoronaldo*	Beyonce	*#beyonce*
E. Macron	*#emmanuelmacron*	R. Federer	*#rogerfederer*	T. Swift	*#taylorswift*
B. Johnson	*#borisjohnson*	R. Nadal	*#rafaelnadal*	Madonna	*#madonna*
T. May	*#theresamay*	N. Djokovic	*#novakdjokovic*	Adele	*#adele*

Identifying Impersonators: To obtain a set of impersonators, I configure a crawler to collect public posts that contain associated hashtag with the name of each account (Table 3.1) between September 2019 and January 2020. For example, in Trump, I gather posts include the *#donaldtrump* tag. Next, based on the methodology that I presented in [52], I measure the profile similarity of the publishers to identify impersonators across case studies. The process of identification of impersonators is later described in Section 4.6.1. In total, I discover 1.6K impersonators with different levels of similarity.

Followers/Followees: I next crawl the follower and followee list of each impersonator from the previous phase (from October 2018 to January 2020). As it is infeasible to collect *all* followers/followees, I define a limitation of 1K for followers and 500 for followees. At the same time, I examined the profile similarity of them to see if they are impersonator or not. Finally, the main list of impersonators extended (by recently detected ones) to 2.3K.

Posts: I crawled the 50 most recent posts published by the impersonator. Furthermore, I gather impersonators' (*i*) profile information, (*ii*) number of comments received on posts, and (*iii*) number of likes attracted on posts. This task was running simultaneously between October 2018 and January 2020.

Table 3.2: Summary of Dataset

Community	**Imposter**	**post**	**comment**	**like**
Politician	36%	30%	36%	35%
Sport player	34%	30%	34%	40%
Musician	30%	40%	30%	25%
Total	**2.2K**	**10K**	**68K**	**90K**

3.3.2 Data Validation

I finally manually inspect the profiles of the impersonators to confirm they are impersonators. I filter any incorrectly identified impersonators alongside their posts. 36 Impersonators were identified incorrectly (1.5% of the total population), and 42 accounts (1.8%) change the application of the page or sell their account at some point during the measurement period. In total, I obtain nearly 68K comments and 90K likes from 10K posts of 2.2K

impersonators. Table 3.2 summarized the entire dataset.

3.3.3 Data Pre-processing

Some features require pre-processing: (*i*) For caption and Profile Biography, I remove all punctuation marks, stopwords and convert them to lowercase characters. I then filter words that contain fewer than three characters, and words are stemmed to reduce to their root forms. (*ii*) I then remove and covert all emojis and emoticons to word format. Then I replace URLs with 'website', emails with 'email', new lines with 'line', and phone numbers with 'phones'. (*iii*) I break down each Hashtag and Username into its constituent words, *e.g.* "*makeamericagreatagain*" contains 4 meaningful words: "make", "america", "great", and "again" [53]. (*iv*) From posts and profile biographies, I extract hashtags (#) and mentions (@) into separated lists. (*v*) Wherever possible, I extract the text from post image thumbnail using Tesseract OCR [54] and apply text pre-processing steps. In Instagram, to get viewer attention, publishers sometimes prefer to put text on images/videos rather than writing a caption. The spaCy [55] is used for French Language Modeling.

3.3.4 Challenges & Limitations

To be able to perform analysis, for each use case, I randomly selected 500k unique users and crawled their profiles which are shown in the last part of the table. These users might be engaged in one or both reactions (like and comment). As a result, the total population contains nearly 1,5M profiles. As the process of crawling profiles is a time-consuming task, a proper pool of them is assessed for this thesis.

3.3.5 Dataset Usage

This dataset has been used in several research papers:

- Koosha. Zarei, R. Farahbakhsh, N. Crespi and G. Tyson. Impersonation on Social Media: A Deep Neural Approach to Identify Ingenuine Content. In 2020 ACM/IEEE ASONAM, 2020. [URL]
- Koosha. Zarei, R. Farahbakhsh, and N. Crespi. How impersonators exploit instagram to generate fake engagement? In ICC 2020, pages 1–6, 2020. [URL]
- Koosha. Zarei, R. Farahbakhsh, and N. Crespi. Typification of impersonated accounts on instagram. In 2019 IEEE 38th IPCCC, pages 1–6, 2019. [URL]
- Koosha. Zarei, R. Farahbakhsh, and N. Crespi. Deep dive on politician impersonating accounts in social media. In 2019 ISCC, pages 1–6, 2019. [URL]

- Koosha Zarei, P. Rajapaksha, R. Farahbakhsh, N. Crespi, G. Tyson, "Multi-Domain and Social Media Aware Language Model Adaptation for Fake Content", IEEE Access 2022.

3.3.6 Ethics

In line with Instagram policies and ethical consideration on user privacy defined by the community, I only collect publicly available data through public API excluding any potentially sensitive data.

3.4 Influencer dataset

Influencers are hugely active on Instagram. In this dataset, I focus on different types of influencers to understand and analyse their behaviour in terms of publishing sponsored content.

3.4.1 Data Collection

The data collection activities took place over four phases:

Phase 1: Hashtags. It is first necessary to obtain a large list of influencer accounts. One approach would be to manually curate this set, however, this would limit us to a small set of influencers, largely dominated by well known celebrities who are easy to identify. Hence, I compile the list by crawling all posts attached to a set of influencer-related hashtags. I turn to the UK's Advertising Standards Agency [56], which states that influencers should use the #ad, #advert or #sponsored hashtags in *any* posts that have been paid for. I expand this list with #advertising, #giveaway, #spon and #sponsor [57].

Phase 2: Post & Stories Collection. I then use the official Instagram API [49] to gather all posts and stories that include any of the above hashtags. Note that stories are similar to normal posts, yet they are automatically deleted after 24 hours (akin to Snapchat posts). Hashtag Engine is used [50] with a maximum of 30 unique hashtags. This API returns public photos and videos that have been tagged with specific hashtags. My crawl for posts and stories ran between Sep 2018 and April 2019. This process identifies 12K accounts that have posted using the previously mentioned hashtag.

Phase 3: Account Collection. Although the above yields a substantial body of posts and stories, I are primarily interested in gathering data on a per-influencer basis. Hence, I next extract all accounts identified from the Phase 2 dataset and begin dedicated monitoring for all posts and stories generated by those users. (*i.e.,* influencers). This covers all posts, reactions and stories from those accounts from July 2019 to August 2019. In this step,

I collect 19.7K posts, 63K stories, 3.1M comments, and 27M likes (generated by the 12K user accounts from Phase 2). Note that they contain a mix of both sponsored (16%) and non-sponsored (84%) entities. For each post, I collect the image, comments, likes and public profile information of the user, as well as any other users who reacted to the post. For each story, I collect the equivalent information, although I cannot collect likes (as these are not available in stories). Each sponsored post is also tagged with the product being advertised, and the category of advertiser. In total, I have 35K posts, 99K stories, 3.1M comments, and 27M likes generated by 12K users.

Phase 4: Categorization. Once I have collected the posts and stories, it is necessary to tag explicitly which are considered sponsored. I take a simple approach. If a post is tagged with one of the above hashtags, I assume that it is sponsored. In the case of Instagram stories, there is explicit metadata which tells us if it is sponsored (the Paid Partnership tag). Hence, for stories I rely on this metadata item (rather than hashtags). Note that this excludes posts that are sponsored, yet the user does not add the appropriate hashtag.

3.4.2 Data Validation

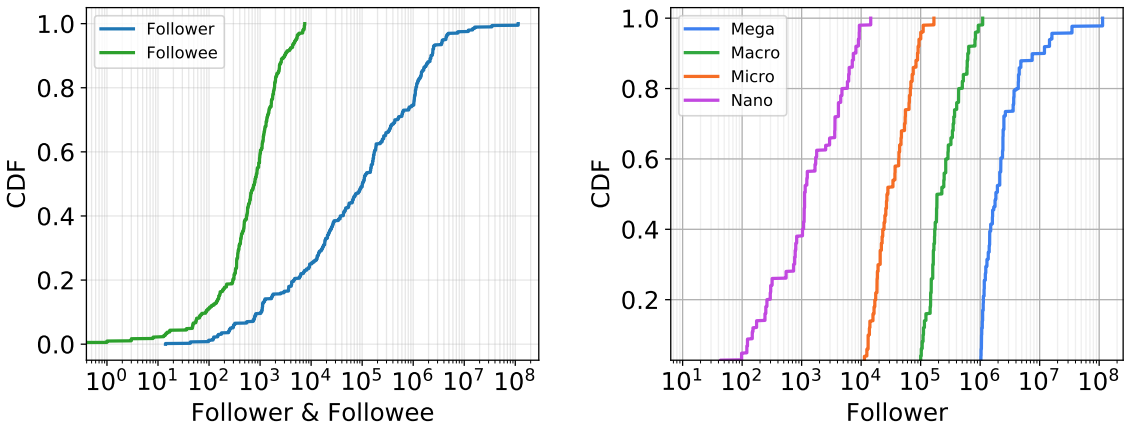
A natural risk is that a subset of the posts containing the curated hashtags may be generated by users who are not influencers. Although it is impossible to entirely discount this at scale, I further perform manual annotation to validate the general correctness of my data. To validate the dataset, I manually looked at the profiles of the influencers to verify if they were really promoting sponsored content.

All users with above 10K followers are checked, confirming that they were all correctly tagged as posting sponsored content. I further check 25% (2K) of all influencers with under 10K followers. I find that the above approach yields 97.6% accuracy: just 48 accounts were incorrectly classified as influencers. Note that the above only checks if a user account has one more truly sponsored posts. To provide further confidence I randomly select 500 influencers and check *all* of their posts. Around 80% of sponsored-post are correctly classified as the sponsored content (based on the hashtags previously mentioned). I filter any incorrectly identified influencers.

3.4.3 Characterising Influencers

I begin by exploring influencers' follower counts and engagement levels (comments and likes), before profiling the types of products promoted.

I take inspiration from past work [58], and begin the analysis of influence by looking at follower counts. This is a natural indicator of influence as it captures the upper-bound of people to whom posts are directly pushed. Figure 3.2(a) presents the cumulative distribution



(a) All dataset

(b) Follower per account

Figure 3.2: Follower counts: (a) CDF of followers and followees of all the influencers in the data. (b) CDF of number of followers per account separated in groups.

function (CDF) of the follower and followee counts of the influencers in my dataset.

Unsurprisingly, I see a sizeable fraction of extremely popular accounts. 35% of users have over 100K followers, and 17% posses over 1M. These conform to the common interpretation of influencers. More surprising, however, is the presence of a long tail: 37% of accounts have fewer than 10K influencers, with 15.5% even having below 1K. At first, I suspected that this may be caused by miscellaneous use of the advert-related hashtags. However, upon manual inspection, I confirm that these are indeed influencers. For instance, @lesya_9_9 with 500 followers has promoted over 10 times more regularly than @blogging_with_tiffany who has fewer than 100 followers. This reveals a growing set of small-scale influencers who promote products, despite their low follower counts. In other words, *influencers are not just celebrities*: they appear to encompass a morass of different account types. For context, I can contrast these results with the followee count (Figure 3.2(a)) which are, on average, far lower than follower numbers. Whereas the median follower count is 23.7K, it is just 770 for the number of followees.

Table 3.3: Influencer Profile Characteristics

Influencer Category	**#Followers Category**	**Avg. follower**	**Avg. followee**	**Avg. mediacount**	**% of verified**
Mega	$\geq 1M$	5.8m	845	9.1k	82%
Macro	$< 1M$ & $\geq 100K$	257k	1.3k	3.1k	22%
Micro	$< 100k$ & $> 10K$	32k	1.9k	1.8k	4%
Nano	$< 10K$	1.5k	0.9k	597	0.5%

The key profile characteristics of influencers (full timeline).

Based on the above findings, I categorise influencers into 4 distinct categories based on their reach (# followers). This taxonomy underpins my subsequent analysis. I term these nano, micro, macro and mega influencers. Table 3.3 presents a summary of these groups. I note that 80% of mega influencers are verified by Instagram, but in contrast, under 5% of nano and micro accounts have a blue verified icon.[1] Figure 3.2(b) presents the CDF of follower counts per-account, broken into these four groups. Naturally, the distributions reflect the split with nano influencers having the fewest followers. For context, a few examples of influencers (top three in terms of followers) from the each categories is shown in Table 3.4. As previously discussed, primarily the more "popular" influencers have verified accounts. These users tend to also have more posts on average.

Table 3.4: Examples of influencers

Category	*Username*	*Section*	*#post*	*#follower*	*#followee*	*#verified*	*#url*
Mega	@kendalljenner	Fashion	3K	116M	203	✓	-
	@vanessahudgens	Fashion	3.2K	36M	1.1K	✓	-
	@brentrivera	Lifestyle	1.8K	16M	395	✓	Youtube
Macro	@tonyamichelle26	Lifestyle	5.3K	937K	2.1K	-	Business Page
	@alice_gao	Design	4.1K	910K	500	✓	Business Page
	@lilleejean	Beauty	700	950K	650	✓	Youtube
Micro	@charisseo_	Fashion	1.7K	98K	600	-	Email
	@morgbullard	Lifestyle	1.9K	97.5K	1.3K	-	Business Page
	@ginascrocca	Fashion	246	96K	1K	-	Business Page
Nano	@jaimesays	Travel	600	9K	1.9K	-	Business Page
	@aberhalloooo	Food	1.1K	9K	7K	-	Email
	@lawrence.carlyFollow	Dance	393	8.1K	1.2K	-	Business Page

3.4.4 Characterising Reactions

Another way to measure "influence" is to inspect engagement levels on a users' posts, *e.g.* comments, likes and mentions. Previous studies [58] have argued that these levels can be a better proxy for influencer than simply inspecting follower counts. In this section, I directly contrast engagement levels for posts that are sponsored *vs.* not sponsored.

Active attention - Comments. Figure 3.3(a) presents the CDF of the received comments per-post for each influencer. I separate posts into sponsored and non-sponsored posts within the influencer timelines. In almost all the cases I observe that sponsored posts receive fewer comments from the users. This suggests that these sponsored posts are of less interest than other posts. This difference is even more significant in mega influencers, where sponsored posts gained 10 times fewer comments than their non-sponsored counterparts. I also observe that the number of comments are ranked in order of Mega, Macro,

[1] Users on Instagram can get verified badge as low as 500 followers. However, that account must represent a well-known, highly searched for person, brand or entity [59]

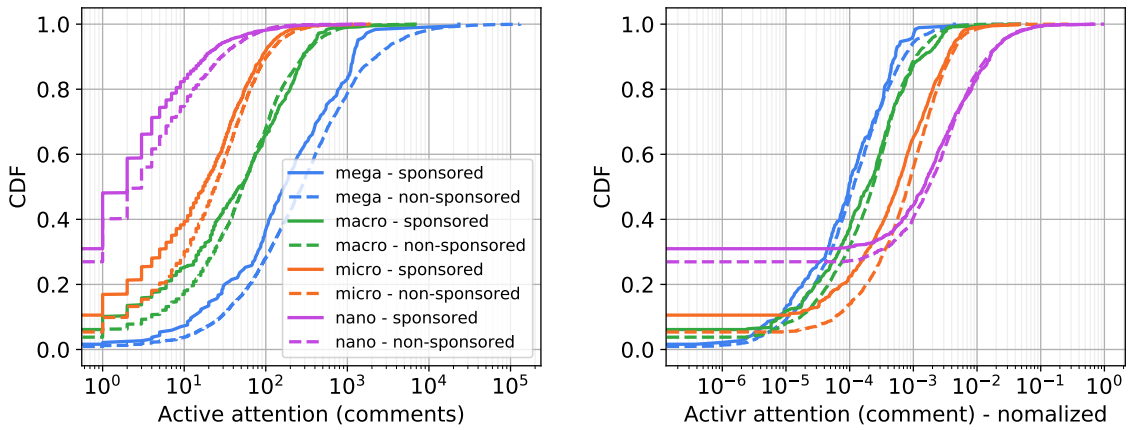
Micro and Nano influencers with, unsurprisingly, Mega influencers getting over 40 times more comments than Nano.

The above analysis of absolute counts may give a misleading perspective as influencers with high follower counts (*e.g.* Mega) will obviously obtain higher comment counts. Hence, I normalize the comment count as a fraction of the follower count, and plot the results in Figure 3.3(b). Here, I see rather different trends with nano influencers gaining the most engagement. In other words, even though popular accounts gain more comments, less popular accounts obtain engagement from a higher fraction of their follower-base. This perhaps sheds light on why nano-influencers have recently started to gain traction, with their ability to engage more targeted populations.

I also inspect the duration before a user adds a comment on a post. Intuitively, comments that are issued shortly after a post is created might be from more engaged users. This confirms similar results to Figure 3.3, with Nano influencers gaining posts most rapidly than their more popular counterparts. In all the cases, non-sponsored posts gain comments earlier; the most significant difference is for Mega influencers. The median comment age for a Mega influencer's non-sponsored posts is 328 *vs.* 366.5 for their sponsored posts. That said, I observe a subtle difference between the influencer groups. In the first hour <30% of comments of all influencers are issued. After the first 10 hours, non-sponsored posts of Macro influencers receive <70% of their total comments, but sponsored posts of mega influencers get only less than 50%. The density of users who reacted to non-sponsored posts is larger, and this difference is more significant in mega influencers and less for nano influencers (plot not shown due to space constraints).

Briefly, I also note that nano influencers tend to have more consistent engagement. Whereas more popular influencers, gain comments from many different users, nano influencers tend to get comments from the same users multiple times: 30% give more than one comment (Figure 3.4b).

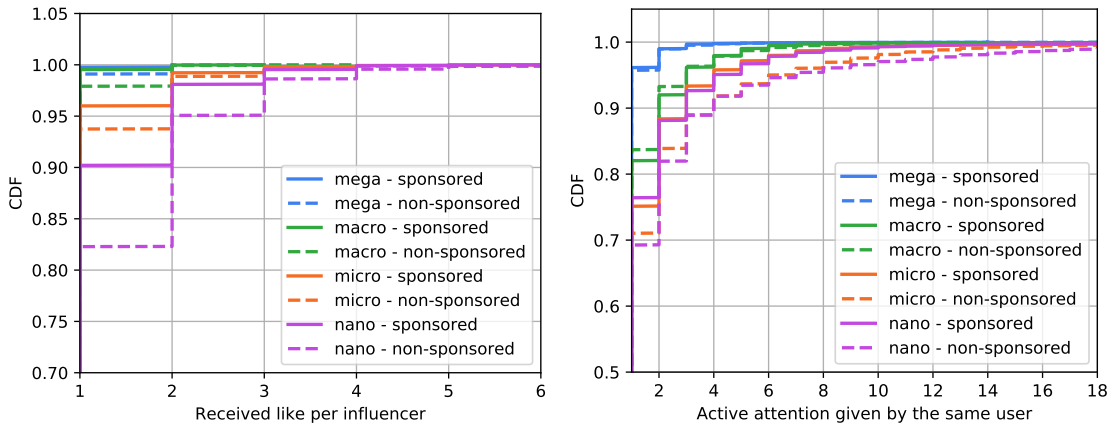
Passive attention - Likes . As comments are generally the most active form of engagement, I also inspect more passive engagement via likes. Figure 3.4a presents the CDF of the received likes for both sponsored and non-sponsored posts among each influencer sategory. For Mega influencers, I observe 28% more likes for non-sponsored *vs.* sponsored posts. This, however, is far less for the other categories (macro, micro, and nano), with an equivalent value of 6%. In fact, I observe that sponsored posts gain marginally more likes than non-sponsored posts for nano influencers (median 56 *vs.* 47). That said, the categories exhibit broadly similar patterns to that seen in comments (with Mega gaining the most and nano gaining the fewest likes in absolute terms). Turning my attention to the normalized like count, I see that again nano influencers get more likes than the other categories. However, I do see outliers: for instance, inside nano, nearly 8% of accounts receive a large number



(a) Absolute (b) Normalized by followers

Figure 3.3: CDF of number of comments received per-post: (a) absolute number; (b) normalized.

of likes, yet nearly no comments (all posts). To investigate this, I manually inspect these subsets of nano influencers. I find that there is a prevalence of fake profiles, apparently using bots to boost their impact.

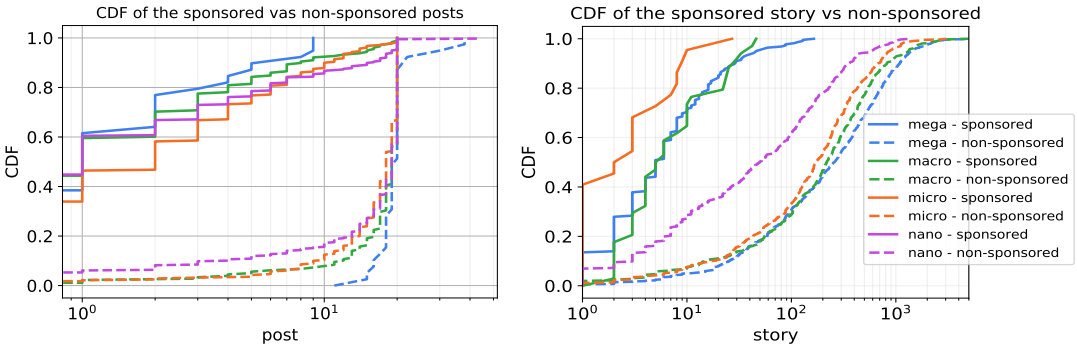


(a) Passive attention (like) (b) Active attention (comment)

Figure 3.4: (a) attracted attention CDF of number of likes received per-influencer; (b) number of comments performed by each user per-influencer

3.4.5 How often do influencers post?

I start by measuring the number of posts per influencer. Figure 3.5(a) presents a CDF plot of the number of sponsored *vs.* non-sponsored posts, whereas Figure 3.5(b) repeats the same for stories.



(a) Post (b) Story

Figure 3.5: CDF of the number of posts and stories published per influencer.

I observe distinct distributions, with most influencers publishing more non-sponsored posts. Only 8.3% of influencers distribute more sponsored posts compared to non-sponsored. On average, 16% of posts are sponsored with just 9.3% of influencers tagging over half of their posts as sponsored. This is anticipated as most influencer guides recommend that users keep the percentage of sponsored posts below 60%, to maintain audience engagement.

Subtle differences can also be observed between the different categories of influencer. For example, where the mega influencers on average post the most sponsored posts, they actually post the least non-sponsored posts. Of course, this might also be a product of how such influencers tag their posts. I find that no sponsored story is published by any nano influencers. This contrasts to what is observed with posts from influencers within the same nano category in Figure 3.5(a), where over 80% of these influencers publish $\leq$10 posts. This striking difference suggests that nano influencers are primarily using posts (rather than stories) to publish sponsored contents. In contrast, mega influencers tend to use stories to promote sponsored contents more regularly (compared to macro and micro influencers). For example, $\leq$21% of Macro and $\leq$3% of micro influencers publish more than 10 sponsored stories compared to over 30% for mega publishers. In general, I see that influencers across mega, macro and micro category favor the use of stories to promote sponsored content compare to post, possibly because it is cheaper to advertise via stories compared to feeds [60]. Another reason for using stories is the exclusivity *i.e.,* followers must stay engaged and must hurry to see the offer or discount code *etc.* while the story

lasts.

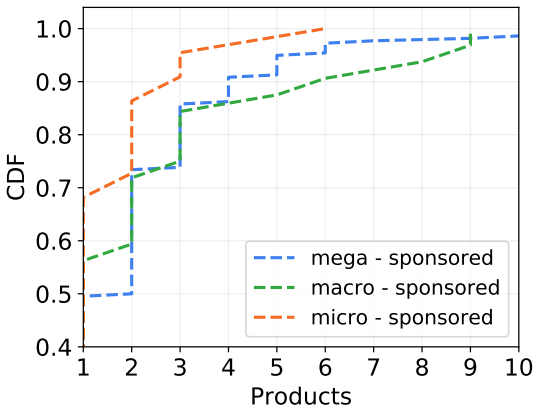


Figure 3.6: CDF of number of products promoted by influencers across categories. This only covers stories because equivalent metadata is not available for posts.

3.4.6 What do influencers promote?

Finally, I wish to inspect the types of products being promoted by influencers. This can be done via the Instagram stories dataset, as each sponsored item is optionally tagged with the category of the advertiser. This is taken from a control set of tags offered by Instagram. Note that this is *only* available for stories, and not regular posts. I find that this feature is not widely used by influencers, with only 3% stating their product.

Figure 3.6 presents a CDF showing the number of products promoted by influencers across categories. Most influencers only promote a single product, particularly in the case of micro. I observe that 50% *Mega*, 58% *Macro* and 70% *Micro* influencers promote just a single product. This suggests that influencers tend to focus on a particular product type, likely in their own specialist area.

I also inspect the type of products influencers use stories to promote. This metadata is captured by the Instagram API, although it only covers stories, as posts do not contain this explicit metadata. Figure 3.7 presents the top 20 product types influencers promote. The Y-axis counts the number of unique accounts promoting each type of product. I observe that most publishers tend to advertise products under *Health/Beauty* (14%), *Product/Service* (11%) and *Clothing (Brand)* (11%). Across products I observe that Mega influencers tend to dominate: they are the major publishers for 77% of product types I identify. This is in sharp contrast to Macro (14%) and for Micro (9%). These findings confirm the intuition that Instagram is dominated by promotions surrounding consumables such as food, retail and beauty. These cover 43% of all adverts.

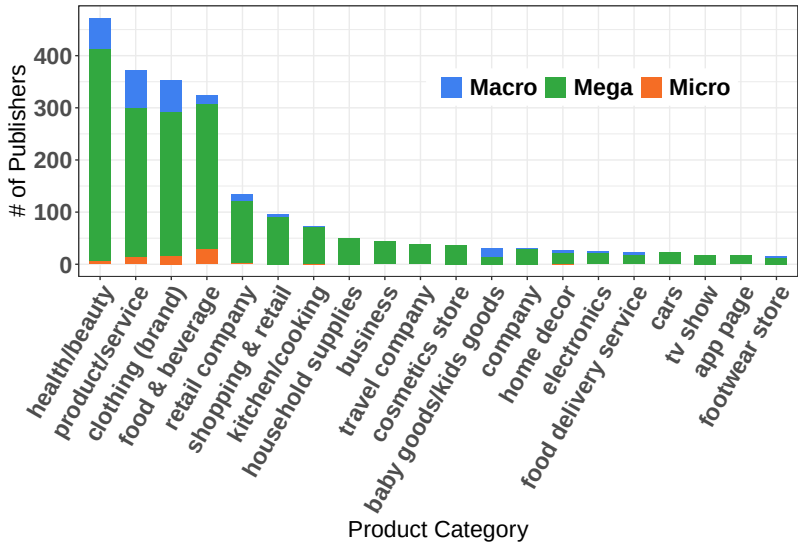


Figure 3.7: Number of products promoted in stories based on their type (identified via the Instagram API).

3.4.7 Dataset Usage

This dataset has been used in:

- Koosha Zarei, D. Ibosiola, R. Farahbakhsh, Z. Gilani, K. Garimella, N. Crespi, G. Tyson. Characterising and Detecting Sponsored Influencer Posts on Instagram. In 2020 ACM/IEEE ASONAM, 2020. URL
- Koosha Zarei, P. Rajapaksha, R. Farahbakhsh, N. Crespi, G. Tyson, "Multi-Domain and Social Media Aware Language Model Adaptation for Fake Content", IEEE Access 2022.

3.4.8 Ethics

In line with Instagram policies as well as user privacy, I only gather publicly available data that is obtainable from Instagram. Whereas I do analyse the content of influencer posts, I do not inspect the content of comments submitted by non-influencer users.

3.5 COVID_19 dataset

Collection COVID_19 information from social media would help to investigate many challenges such as fake news detection, hate speech detection, data distribution, user behaviour analysis, *etc.* In this part, I focus on Instagram published posts.

3.5.1 Data Collection

Collection. On 5 January 2020, I prepared an initial list with '*#coronavirus*', '*#covid19*', and '*#covid_19*' keywords. I list the complete tracked keywords/hashtags in Table 3.5. Whenever a new keyword appears, I add it to the watch list. I continuously check new hashtags from [61] and [62] sources. For example, on 19 January 2020, I added '*#corona*', and '*#stayhome*'. By the end of January and beginning the lockdown in Europe, I also began to track '*#quarantine*', and '*#covid*' tags.

Using the above crawler, I continuously iterate over this list to collect associated posts. If any of the keywords exist in a post's *caption, hashtags, tagged users, location, or mentions*, I consider that post as COVID-19 related. In order to get post reactions, I revisit posts for two weeks after the initial posting to gather comments and likes.

Table 3.5: Tracking Hashtags on Instagram

Hashtag	**Post**	**Publisher**	**Reaction**	**Crawled Since**
#coronavirus	12.7K	11K	7.3M	*January 5, 2020*
#covid19	8.0K	7K	6.5M	*January 5, 2020*
#covid_19	6.1K	5.9K	1.1M	*January 5, 2020*
#corona	2.9K	2.7K	1.9M	*January 19, 2020*
#stayhome	2.9K	2.7K	421K	*January 30, 2020*
#quarantine	2.3K	2.1K	322K	*January 30, 2020*
#covid	1.6K	1.4K	135K	*January 30, 2020*
#socialdistancing	0.7K	490	43.9K	*January 30, 2020*
#pandemic	0.7K	354	55K	*January 30, 2020*
#lockdown	1.3K	644	68K	*January 30, 2020*

Graphs. I later explore the relationships between hashtags. To achieve this, I induce a graph dataset whereby hashtags that appear in posts are nodes, and edges indicate that two hashtags have appears in the same post (at least once). I only consider hashtags between 3 to 25 characters. I set the node weight as the frequency that a tag is used. I later plot graphs using [63].

Bot Detection. In order to identify bots, I extract and use features from [64–66] studies. Features are a combination of post and publisher metrics: *"profile image (image), biography text (text), account url (text), full name (text), number of followers (numeric), number of followee (numeric), account age (numeric), number of posts (numeric), avg. received like (numeric), avg. received comments (numeric), number of posts (numeric), number of issued like (numeric), number of issued comments (numeric), following/followee ratio (numeric), followers/post ratio (numeric), biography emoji count (numeric), biography hashtag count (numeric), biography length (numeric), verified (numeric), duplicated comments (numeric),*

number of followers that are bots (numeric), number of followee that are bots (numeric), post caption (text)". To build a training set, I randomly select 6K posts and manually label the profiles. Based on mentioned metrics, I examine each profile by hand and annotate it as "*bot*" or "*not bot*" identity. Metrics include profile-level features (*"full name, profile image, number of follower, verified, account age,* etc. *"*) and post-level features (*"received like, received comments, post caption,* etc. *"*). In the training set, each class has 2.1K validated samples. For all text-based features such as "biography", I remove all punctuation marks, stopwords and convert them to lowercase characters. Words are stemmed to reduce to their root forms. Numerical metrics are min-max normalised. Next, I train a Contextual LSTM Neural Network classifier with the same model architecture reported in [67]. In this model, both text and metadata metrics from posts and profiles are considered. First, I tokenize text metrics (*e.g.* biography) using Keras Tokenizer Class [68] and then the result is fed to the LSTM layer which outputs a 64-dimension vector. I attach numerical metadata to this vector and pass it through 2 ReLU activated layers of sizes 128 and 64. Finally, it connects to an output layer that predicts the label. I use a random split of 80% (training set) and 20% (test set), and to avoid over-fitting I use 10-fold Cross-validation. The Contextual model achieved a final accuracy of 88%, precision of 87%, recall of 87%, and F1 of 88%.

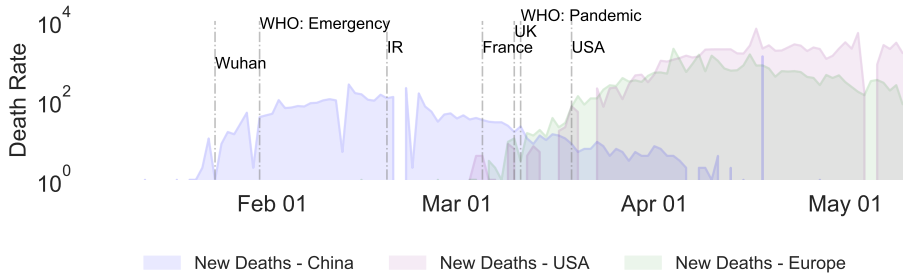
3.5.2 Limitations

Note that as it is infeasible to collect all reactions. Hence, I define a limitation of 500 comments and 500 likes per post. I monitor reactions for up to two weeks to reach this limitation. In line with Instagram's Terms, Conditions, and Policies [69] as well as user privacy, I only gather publicly available data that is obtainable from Instagram. I also only rely on Instagram Posts and Reactions. I do not collect other data types such as Stories or Highlights.

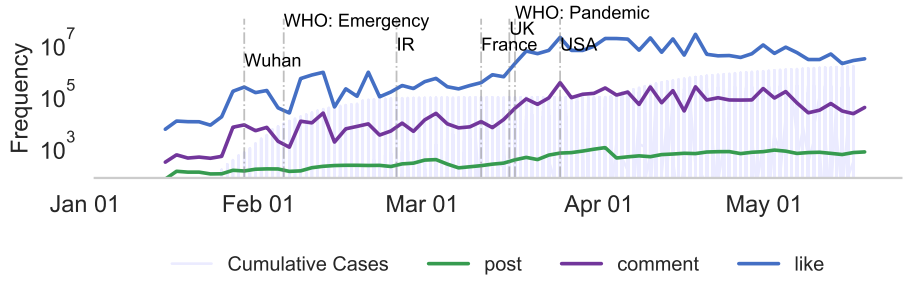
3.5.3 Data Summary

Figure 3.8.a presents the death rate in different locations as reported by the World Health Organization (WHO). I compare that with the trends of total '*posts*', '*comments*', and '*likes*' in Figure 3.8.b. As reported by the WHO, the COVID-19 outbreak began several months before January 2020 in Wuhan (China), but at the time, it was not considered a global crisis yet. That is the reason why I see figures start with large numbers. Post content is published by various groups such as ordinary people, politicians, companies, news media, governments, fake identities, *etc.* The publishing rate increases continuously (post and reactions) during this period. However, there are some surges and fluctuations in numbers in several critical points: (*i*) The outbreak in the city of Wuhan and the first peak of death rate in China (Jan 2020); (*ii*) The announcement of a state of emergency by the WHO (Feb

2020); and (*iii*) The beginning of the first wave and the surging death rates in Europe and the USA (March 2020).



(a) World Health Organization Rep



(b) Content

Figure 3.8: (a) The death rate reported by the WHO. (b) The overall trends of the published content.

In total, I have collected 829K comments and 3.2M likes from 25.7K public posts. Posts are distributed by 13.3K publishers. Table 3.6 summarizes the general stats regarding the posts, profiles, and reactions. Each Instagram part may contain various data types. For example, in a post, there exist 'caption', 'location', 'date', 'hashtags', 'mentions', *etc.* In Table 3.7, I summarize and describe all data features. This covers four main data types: '*text*', '*numeric*', '*boolean*', '*date*', and '*binary*'. I store images in a binary format.

Table 3.6: General Dataset Stats

Post count	25.7K	avg. follower per account (median)	2.3M (695)
Unique publishers	13.3K	avg. followee per account (median)	725 (276)
Total Comment	829K	avg. mediacount per account (median)	1.6K (127)
Total Like	3.2M	avg. received like per post (median)	10K (43)
Total reactions	4M	avg. received comment per post (median)	141 (2)

Table 3.7: List of Data Features

Type	Attribute	Type	Description
Profile Metadata	follower count	*numeric*	*audience size*
	followee count	*numeric*	*friend size*
	media count	*numeric*	*published posts*
	is verified	*boolean*	*verified by instagram*
	is private	*boolean*	*public or private*
	full name	*text*	*full name text*
	biography	*text*	*biography text*
	username	*text*	*account username*
	id	*numeric*	*unique id*
	profile pic url	*text*	*picture url*
	external url	*text*	*if exists*
	is business account	*boolean*	*if exists*
Post Metadata	caption	*text*	*post caption*
	date	*date*	*publish date*
	like count	*numeric*	*number of likes*
	comment count	*numeric*	*number of comments*
	shortcode	*numeric*	*unique id*
	hashtags	*text*	*list of hashtags*
	mentions	*text*	*list of caption mentions*
	is video	*boolean*	*video or image*
	video url	*text*	*if video == true*
	location	*numeric*	*location tag*
	tagged users	*text*	*tagged users in photo*
	thumbnail	*binary*	*content thumbnail*
	id	*numeric*	*unique post id*
Like Reaction	username	*text*	*username*
	id	*numeric*	*unique user id*
Comment Reaction	username	*text*	*username*
	id	*numeric*	*unique user id*
	date	*date*	*publish date*
	text	*text*	*comment text*

3.5.4 Characterising Publishers

First, I strive to understand COVID-19 related publishers. I argue that this can offer insight into how this information is generated and distributed [70]. I particularly focus on understanding how much COVID-19 generated information can be considered reliable [71].

Overall, I identify approximately 13.3K unique publishers. I observe a range of account characteristics. For example, some accounts have a high number of followers, and some represent well-known figures such as celebrities or brands. I categorize publishers into the following groups, as summarized in Table **??**:

(1) News agencies. To identify News agencies, I make a list of English speaking agencies on Instagram using two sources [72, 73]. Then, I filter and verify more than twenty News media accounts in my dataset. While all these accounts are already verified and categorized

as 'media/news companies' by Instagram, they usually have millions of followers. I list all existing News agencies in Table 3.8. I find that 12.2% of posts, 0.7% of unique publishers, and 26% of total reactions belong to News agencies.

Table 3.8: List of News Agencies in my dataset

@washingtonpost	@nbcnews	@abcnews	@dwnews
@skynews	@foxnews	@wsj	@france24_en
@bbcnews	@cnn	@time	@nytimes
@dailymail	@euronews.tv	@the.independent	@telegraph
@politico			

(2) Celebrities. I also witness the existence of posts from popular singers, actors, artists, sports players, and other figures. I compile a list of popular celebrities using [74] [75] and then search for them in the dataset. I find that these celebrity accounts tend to be verified public profiles, usually with millions of followers (avg. 80M) yet few followees (avg. 230). Some of the top figures that I see are *'@ladygaga', '@arianagrande', '@jlo', '@oprah', '@leonardodicaprio', '@christiano', '@leomessi', '@serenawilliams', '@davidbeckham', '@eltonjohn', '@jenniferaniston', '@theellenshow', '@kimkardashian', '@beyonce',* etc. The number of celebrity accounts is not as large as other groups, and they usually publish more Instagram stories or live broadcasts rather than posts. However, they obtain a large number of reactions, especially comments, which make them a valuable source (see Figure **??**). This group holds 4.3% of all posts, 0.5% of unique publishers, and 45.2% of total reactions.

(3) Business Pages. These cover the official pages of companies on Instagram. To identify such accounts, I rely on [76, 77], and use the Instagram Category feature (as a company) [78]. Using these two resources, I extract all known business pages. I identify two types of business accounts: (i) profiles that are already verified by Instagram as business profiles [79] such as '@Nike', '@google', '@chanelofficial', *etc.* with hundreds of followers. (ii) Profiles that represent small businesses that are not verified and have few followers. Business pages produce the longest caption length (average 628 characters) and tag the most people (1.5 on average). Business Pages hold 4.7% of the total posts, 26% of total reactions, and 2% of unique publishers.

(4) Influencers. Some accounts are known as "influencers". These are refer to accounts that specifically attempt to influence public opinion, often in return for financial payments [67]. I filter and extract influencers based on feature set from [67]. Influencers utilized the highest number of hashtags within their posts (avg. 18). This group holds 4.8% of total posts, 1.3% of unique publishers, and 0.8% of total reactions.

(5) Bots. I further identify a set of bot accounts. These refer to accounts that are computationally operated [70]. I use a Contextual LSTM Neural Network classifier with 88%

accuracy in order to train to identify bot accounts. The process of training the classifier, feature set, and results are explained in Section **??** in detail. Bot generate 6.9% of total posts, 2% of unique publishers, and 0.2% of total reactions.

(6) Public Accounts. I refer to the rest of the publishers as "Public Accounts". In this category, profiles are non-verified public accounts that have a few to millions of followers. This group holds 67.1% of total posts (the most populated group), 93.5% of unique publishers, and 1.1% of total reactions.

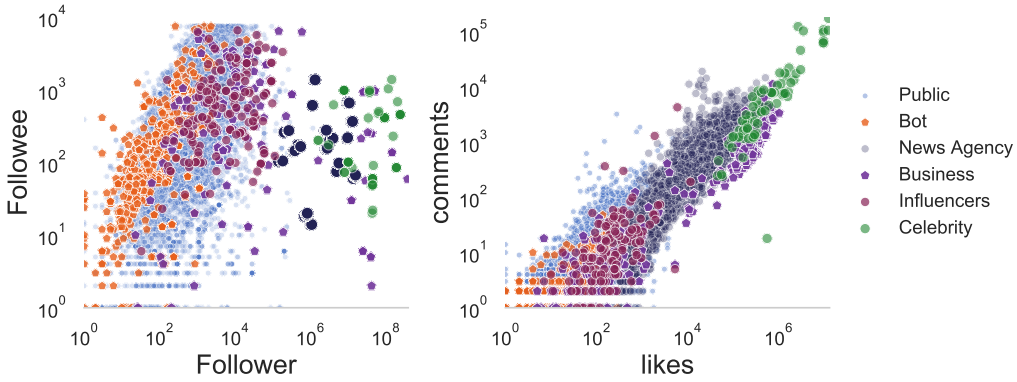


Figure 3.9: The Dot Plot of a) follower/followee count of categories, and b) received reactions across categories.

Validation. In order to validate categories, I manually check each one individually. For News agencies, Business pages, and Celebrities, I examine all samples and 100% of accounts are identified correctly. 86% of these accounts are already verified and approved by Instagram. For the influencer category, I randomly select 25% of samples and examine each by hand. 94.3% of influencers are identified correctly. To validate influencers I use the feature set from [67]. In the bot category, I randomly select 25% of samples and examine each manually. 94.3% of bots are identified correctly. To validate bots, I use [64–66] metrics. The process of bot detection is presented in Section **??**. Note, for the prior analysis, I remove incorrect samples from groups.

Here, most of the posts are published by the 'Public' category (79% of posts) followed by 'News agencies' (12.2% of posts). Overall, I see a growing number of weekly posts. Public publishers have the highest rate, thanks to the volume of accounts in this category (79%). Similarly, the Celebrity group publishes the fewest points (as they constitute just 0.3% of accounts). I find that these trends are also impacted by key events. The main surges occur, first, after Europe announces the pandemic on 14 March 2020 with 18.2%, followed by the USA on 26 March 2020 with 4.5%. News posts tend to be driven by dedicated coverage given to COVID-19. For instance, 13 of the well-known agencies have launched a dedicated

sections to cover the latest headlines. For example, BBC Coronavirus Stories [80], Euronews Special Coverage [81], Time COVID-19 Track [82], Skynews COVID-19 Section [83], and Foxnews Latest Coronavirus Headlines [84].

Perhaps most interesting is the Bot category. First, they publish a large volume of posts (4.9% of posts). This is the third most active category Second, they publish an almost fixed amount of posts during this pandemic, without the fluctuations seen in other groups. For example, compared to the News Agency group, I do not witness any noticeable peaks. This may be because of the computational manner in which such accounts generate content. The same trend is also reported in Twitter in [85] which shows an uptick in the frequency of bots' tweets referencing COVID-19 in the same period, and active bots sent 185K tweets and 1.4K retweets.

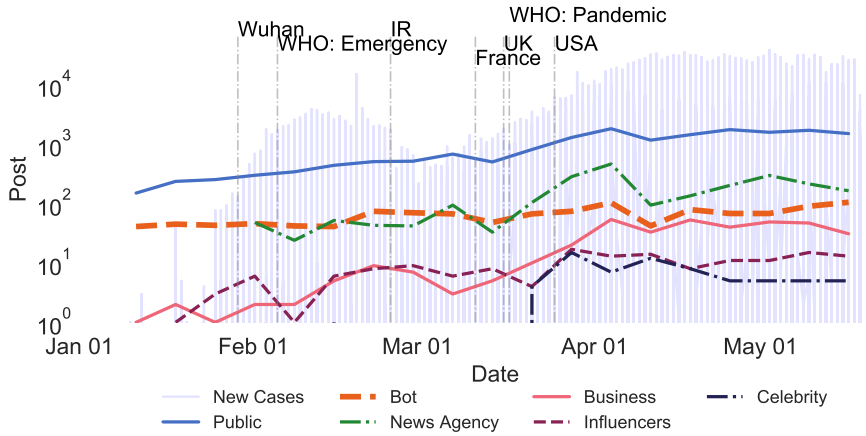


Figure 3.10: The trends of published posts by categories.

The Business category also exhibits unique trends. Due to the national lockdowns (largely introduced in April 2020), many businesses released information via Instagram and shifted activities online. Thus, I see noticeable growth in posts during this period. I also find that Influencers, ranging from 'nano' to 'mega' [67], increase their number of posts during this period. Similar trends can be seen amongst Celebrity accounts. Note that most in my dataset are American English-speaking figures. Therefore, there is almost no content until the start of the pandemic in Europe (March 2020). The authors of this study [86] also reported the same trends in Twitter and Instagram posts associated with COVID-19 content.

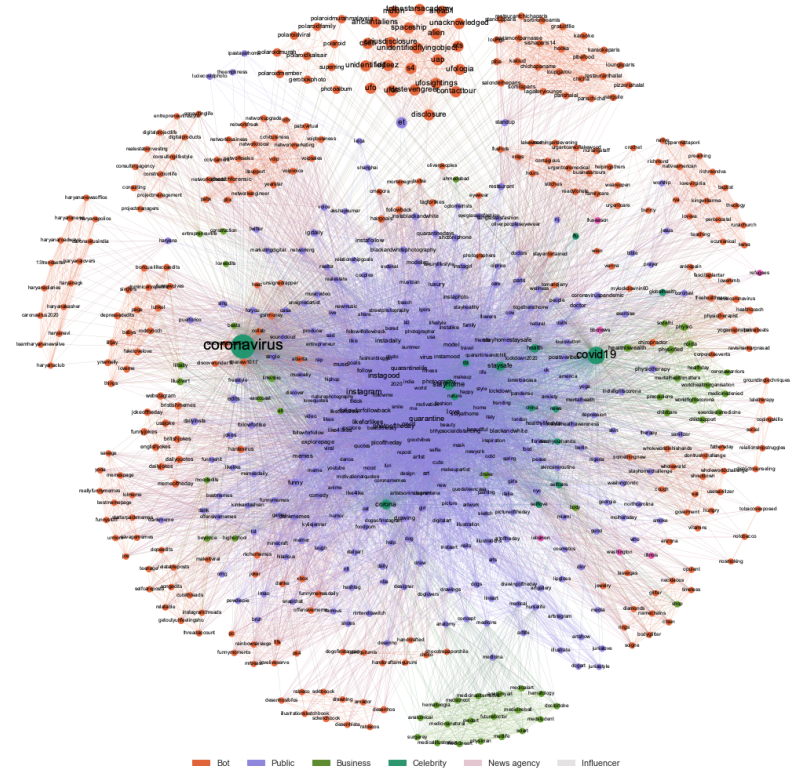
3.5.5 Characterising Hashtags

Figure 3.11 presents the graph of all hashtags. I colour code the hashtags based on the category of accounts (each colour represents one category). If a hashtag belongs to two groups, that node gets the colour of the category that is greater in size. In this network, main hashtags such as '*#Coronavirus*' and '*#Covid19*' (Table 3.5) are located at the centre, surrounded by more connected nodes.

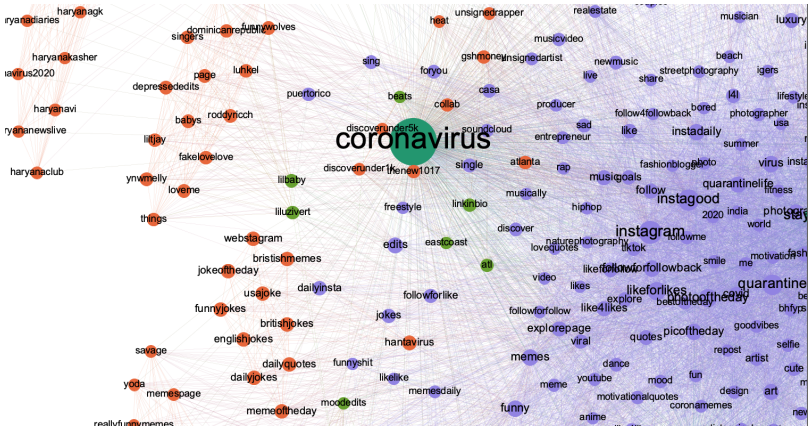
Nodes with larger sizes (frequency) are located closer to each other. I see this characteristic in hashtags used by 'Public', 'Celebrity', 'News agency', and 'Influencer' categories. That is the reason why they are located closer to the main nodes. Furthermore, they are more topic related, and hold higher connection rates with others. In contrast, interestingly, hashtags primarily used by bots (red nodes) are located far from the centre with smaller sizes and fewer node connections. I also see the same behaviour in 'Business' nodes (green nodes).

I witness 36 isolated islands. Islands are where there are a set of inter-connected hashtags that are disconnected from the main network of hashtags. Each island contains between 5 and 39 nodes. These islands have several characteristics: (*i*) In each island, all nodes are well-connected to each other (avg. 21 internal connections), but there is a weak connection to external nodes (avg. 6). (*ii*) Island nodes are used together in the same posts. So, node sizes are equal. (*iii*) Islands connect directly to the main network node or through a few nodes (max 4 connections). In this case, some islands are connected directly to the '*#Coronavirus*' node, but others with some extra nodes. (*iv*) Individual islands are disconnected from each other. In 21 islands (out of 36), I see no direct connection between them. This behaviour can be seen in the bot category (with 36 islands), which is presented in red nodes, and two green islands from business accounts. The same mechanism is also reported from authors of this study [70] which investigate the anatomy of online misinformation networks.

I look into two islands which are shown in Figure 3.11.b. The first island, from bot category (in red), is using 10 hashtags of '*#britishmemes*', '*#britishjokes*', '*#dailyjokes*', '*#webstagram*', '*#jokeoftheday*', '*#dailyquotes*', '*#usajoke*', '*#funnyjokes*', '*#memoftheday*', and '*#englishjokes*' to talk about jokes. These hashtags are used in 432 posts together in the same individual posts, and as results, the node sizes are the same. Each node is connected to all other island's internal nodes (9 nodes). This island connects to the main network node directly (through the '*#coronavirus*' hashtag), and indirectly (*e.g.* through '*#dailyinsta*', '*#jokes*', and '*#funnyshit*'). In the other island from bots, 14 hashtags of '*#depressededits*', '*#babys*', '*#fakelovelove*', '*#things*', '*#singers*', '*#loveme*', and others are used together. I also see that all internal nodes are connected to each other (each to 13 nodes). The node sizes are the same as they appear together in posts, and the island is connected to



(a) Hashtag Graph



(b) Zoom preview

Figure 3.11: The use of hashtags across categories.

the network graph directly to the main node (‘*#coronavirus*’) and indirectly through some other nodes (‘*#puertorico*’, ‘*#lilbaby*’, ‘*#sing*’, ‘*#freestyle*’, and others). There is no connection between this island and other identified islands. These hashtags are used in 561

posts together.

3.5.6 Dataset Usage

This dataset has been used in:

- K. Zarei, R. Farahbakhsh, N. Crespi and G. Tyson, "Dataset of Coronavirus Content From Instagram With an Exploratory Analysis," in IEEE Access, vol. 9, pp. 157192-157202, 2021, doi: 10.1109/ACCESS.2021.3126552. [URL]
- Koosha Zarei, P. Rajapaksha, R. Farahbakhsh, N. Crespi, G. Tyson, "Multi-Domain and Social Media Aware Language Model Adaptation for Fake Content", IEEE Access 2022.

3.5.7 Access To Dataset

This dataset is accessible through: `https://github.com/kooshazarei/COVID-19-InstaPostIDs`. I publish my dataset in agreement with Instagram's Terms & Conditions [69]. Thus, as it is not permissible to release the post content and reactions, I share the post IDs (known as *shortcodes*). Researchers can then use tools such as Instaloader [87] to dehydrate the dataset. For any further question, please contact Koosha Zarei (*koosha.zarei@telecom-sudparis.eu*).

3.5.8 Ethics

In line with Instagram policies and ethical consideration on user privacy defined by the community, I only collect publicly available data through public API excluding any potentially sensitive data.

3.6 Conclusion

In this chapter, I explained three datasets that are obtained from social media in different domains: COVID_19, Impersonations, and Influencers. All datasets are crawled from Instagram through a dedicated crawler that I implemented in line with Instagram policies and ethical considerations on user privacy. The details of data collection, limitations, a summary of the data, and data characteristics are discussed separately for each dataset.

In the next chapter, I will mainly focus on the challenge of Impersonators as an important type of fake identities on social media and I will analyse their activities in terms of published content and fake engagements.

Chapter 4

Impersonator: Fake identities & Ingenuine Content in Social Media

Contents

4.1 Overview **59**
4.2 Related Work **59**
4.2.1 User Behaviour 60
4.2.2 Fake account 60
4.2.3 Bot generated content 61
4.3 Definition **61**
4.3.1 Bots 61
4.3.2 Political Bots 61
4.3.3 Impersonator or Imposter 61
4.3.4 Impersonation and Social Media Profile Theft (SMPT) 61
4.3.5 Profile Similarity 62
4.3.6 Types of Impersonators 63
4.4 Case Study Accounts **64**
4.5 Data Collection & Data Pre-Processing **64**
4.6 Identification of Impersonating Accounts **64**
4.6.1 Identifying Impersonators 64
4.6.2 Primary Account Analysis 66
4.6.3 Clustering 66
4.6.4 Manual inspection for validation. 68
4.7 A Deep Neural Approach **68**

4.7.1 Dataset Overview 69
4.7.2 Over-Sampling 69
4.7.3 Feature Engineering 69
4.7.4 Proposed DNN Architecture 70
4.7.5 Feature Analysis 72
4.8 Assessing Published Content 72
4.8.1 Politicians 74
4.8.2 Sports Players. 74
4.8.3 Musicians 75
4.9 Conclusion 76

4.1 Overview

Impersonation is where (sometimes malicious) users create social media accounts mimicking a legitimate account [52]. For example, impersonators or imposters maybe accounts that pretend to be someone popular or a representative of a known brand, company, *etc.* Such impersonators are found on all major social media platforms such as Facebook, Twitter, Instagram, YouTube and LinkedIn. Among these platforms, Instagram is widely used by celebrities, influencers, businesses, and public figures with different levels of popularity. Although many impersonators may be innocuous, there also exists malicious fake accounts. These often have clear plans, where they make accounts appear more popular than they are, produce pre-planned untrustworthy content, perform brand abuse or generate fake engagement [7]. Therefore several lawsuits have taken place in the United State (along with other countries), where criminal impersonation is a crime. It involves assuming a false identity with the intent to defraud another or pretending to be a representative of another person or organisation [8]. However, identifying such activities is often slow and laborious — hence, developing techniques for automated detection would have real value to social media companies.

In this thesis, I aim to identify impersonator-generated content and understand the role of impersonators on content propagation in Instagram. Towards that end, I pick several different and important communities with verified genuine accounts inside each. Through the pool of collected public content, I identify a set of impersonator accounts. Bots are fake accounts or social bots that tend to mimic the real user and accomplish a specific purpose [88] and interacts with humans on social media [18]. In contrast, fans are (semi-) human-operated accounts that are created and maintained by a fan or devotee about a celebrity, thing or particular phenomenon. I use clustering techniques to create necessary labels for building and training a Deep Neural Network to predict post types: (*i*) bot-generated, (*ii*) fan-generated, or (*iii*) genuine content. I finally focus on the published content of impersonators to shed light on their behavioral patterns. I leverage natural language processing (NLP) techniques to understand post captions, get written topics and sentiments, and compare results to genuine content.

4.2 Related Work

The related work to this study includes the behavioral analysis of users and the fake entities in social media, as well as fake content and fake engagement detection and analysis.

4.2.1 User Behaviour

The authors in [26] [27] look at the profile and behavioural patterns of users and discuss existing challenges on different OSNs. By integrating semantic similarity and existing relationships between users, it is possible to match profiles across various OSNs [28] [29]. Also, [30] conduct a detailed investigation of user profiles and proposed a matching scheme. On Instagram, for the sake of mitigating impersonation attack, [25] explored fake behaviours and built an automated mechanism to detect fake activities.

On another line of research, the authors in [26] [27] look at the profile and behavioural patterns of a user and discussed existing challenges on different OSNs. By integrating semantic similarity and existing relationships between users, it is possible to match profiles across various OSNs [28] [29]. Also, [30] conducted a detailed investigation of user profiles and proposed a matching scheme. On Instagram, for the sake of mitigating impersonation attack, [25] explored fake behaviours and built an automated mechanism to detect fake activities.

4.2.2 Fake account

Recent research has worked on related research problems and dedicated a fair amount of work to study different aspects of OSNs. In this era, looking to behavioural aspect of users and understanding the different patterns is still a hot topic of research. Several studies shed light on this direction by profiling users based on their activities and reactions. This work [16] presents a novel technique to discriminate real accounts on social networks from fake ones. One study [17] provides a review of state-of-the-art Sybil detection methods.

Caruccio et al. [89] proposed a novel technique to discriminate real accounts on social networks from fake ones. They collected data from 9K Twitter accounts. An algorithm was used to discover relaxed functional dependencies (RFDs) from data. The extracted RFDs were used to differentiate fake, real, and verified accounts. Phad and Chavan [90] suggested a model for identifying a compromised profile on social networks. They gathered data from Twitter accounts for their study using the Twitter archive. The dataset contained 26 363 tweets from 48 high-profile accounts. Out of these, 25 363 tweets are legitimate, and 1000 were malicious.

Chakraborty et al. [91] have proposed a framework called social profile abuse monitoring. They gathered information from the Twitter profile of 5000 users along with their 200 latest tweets. The SVM classifier was used to analyze the dataset. They introduced a four-class classification model for calculating profile similarity indexing based on fine-grained interface similarity characteristics. Adewole et al. [92] proposed a model that detected both the spam message and the spam account in the OSN websites. For the identification of spam

messages, a dataset was used, which was compiled from three sources: SMS collection V.1, SMS Corpus V.0.1 Big, and Twitter Spam Corpus with a total of 5574, 1324, and 18 000 samples, respectively. 20 998 twitter accounts and 3 755 367 tweets were used to detect spam accounts.

4.2.3 Bot generated content

On the other hand, the existence of Bots can alter the perception of social media influence, artificially enlarging the audience of some people. The problem of rising social bots are discussed in [18]. There are various strategies to tackle the problem of bot detection. [19] suggested a profile-based approach and [20] proposed a novel framework on detecting spam content. Also, [21] present a machine learning pipeline for detecting fake accounts and authors in [22, 23] present a method to classify bots and understand their behaviour in scale. The use of political bots during the UK referendum on EU membership is explained in [93] and also, [94] [95] described computational propaganda and define political bots designed to manipulate public opinion In the US context.

4.3 Definition

4.3.1 Bots

Bots are (semi-) automatic agents that are designed to accomplish a specific purpose [88] and automatically produce content and interacts with humans on social media [18]. Bots are normally defined with the condition of mimicking human behaviour [96].

4.3.2 Political Bots

Political bots are automated accounts that are particularly active on public policy issues, elections, and political crises.

4.3.3 Impersonator or Imposter

Impersonator/Imposter is someone on social media who builds a profile using the information of another legitimate account and pretends to be that entity or copies the behaviour/actions of that profile [7] (Figure 4.1).

4.3.4 Impersonation and Social Media Profile Theft (SMPT)

SMPT takes place when an impostor sets up a fake profile on social media which mimics another user as a prank or to mock them [97]. By using this account, they gain the trust of the original user's followers for different purposes such as fake promotions, to generate

Figure 4.1: Two samples of impersonators of Donald J. Trump on Instagram. The first snapshot belongs to the genuine account and the others are imposters.

followers, to gather information, spreads political views, supports or oppose actions etc. Some criminals are using this strategy to deceive the public and commit crimes. They attempt to establish relationships using false facts and then defraud unsuspecting targets. A fake social media account could result in legal action against the impersonator. On Instagram, it can be possible to report a fraud.

4.3.5 Profile Similarity

I use this term to indicate whether there is any similarity or correlation between two Instagram profiles. Similarity can be in (*i*) text features [98] such as username, full name, or biography *e.g.* '*@barackobama*' and '*@barack_ _obama*', or (*ii*) profile photos (if the same person exists in both photos). The '*Similarity Level*' could be high (similar in all metrics), low (just in one metric), or between. An example of the genuine Theresa May account and her impersonator with a high degree of similarity is shown in Figure 4.3.

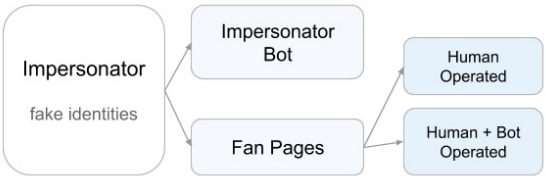


Figure 4.2: The taxonomy of impersonators.

In [52] [99] I introduced the problem of impersonation and discussed the identification methods. Then, I uncovered unknown groups of impersonators and examined their behaviours. For example, fan pages have a higher number of followers and are completely public pages. But bots, have very fewer followers and publish a lot of posts in a shorter period of time. Then, in [7] I studied the comments they generated under the post of genuine figures in details. For example, bots produce much higher duplicated comments than

others and give likes (passive reaction) faster.

4.3.6 Types of Impersonators

Eventually, in this thesis, I divide impersonators into two broad types of public accounts (Figure 4.2):

1. **Bot Impersonator (Bot)**: these public fake accounts or social bots tend to mimic the real user and generally generate specific content. First, from profile characteristics, bots are usually simple accounts that use default Instagram settings: no full name, no biography, and sometimes no profile photos. The follower count is low and they follow a lot of other accounts. From similarity viewpoint (compared to a genuine user), bots have weak profile similarity degrees: they have no similar profile photo and have low similarity in username, full name, or biography. From activity viewpoint, bots receive very limited engagement (like or comment) per post, are lazy in publishing stories, are so active in giving comments and likes to others, and the rate of issuing duplicated comments is high. Existing bots vary in sophistication. Some bots are very simple and merely re-publish posts, whereas others are sophisticated and can even interact with human users or post comment. In this study, '*Bot Impersonator*' and '*bot*' terms are interchangeable.

2. **Fan Impersonator (Fan)**: is a (semi-) human-operated account that is created and maintained by a fan or devotee about a celebrity, thing or particular phenomenon. From profile perspective, fans have a greater follower number than bots, are completely public accounts, have a biography, and usually use a URL. From impersonation viewpoint, fans have higher profile similarity in photo, username, full name, and biography metrics. From behaviour viewpoint, fans are interested in publishing posts and stories, are more productive than bots, receive higher engagement within their posts (both like and comment), and the owner barely shares self-generated content. From managing viewpoint (who controls the page), I can divide fans into two different types (Figure 4.2): (*i*) A fan page which is regulated by 'human'. In this situation, there is no automation movement and all content and activities are published by a human. (*ii*) A fan page which is regulated by 'human and bot'. In this type, page owner which is a human usually use some automation and bot services to gain attention. For example, using a bot to comment or like on related pages.

4.4 Case Study Accounts

To seed my analysis, I select a set of 15 ground-truth verified accounts from three communities: *politicians*, *sports stars*, and *musicians (celebrities)*. I pick these communities to compare the impersonation problem in divided societies. For each community, I select the top 5 most popular verified accounts manually, then I confirm the popularity by [100]. Details of the data crawling is explained in Section 3.3.

1. **Politicians:** Donald J. Trump (*@realdonaldtrump*) the president of the United States, Barack Obama (*@barackobama*) the previous president of the United States, Emmanuel Macron (*@emmanuelmacron*) the president of France, Boris Johnson (*@borisjohnsonuk*) the Prime Minister of the United Kingdom, and Theresa May (*@theresamay*) the former prime minister of the UK are considered in this group.

2. **Sports Stars:** Leo Messi (*@leomessi*), Cristiano Ronaldo (*@cristiano*), Rafael Nadal (*@rafaelnadal*), Roger Federer (*@rogerfederer*), and Novak Djokovic (*@djokernole*) are selected.

3. **Musicians:** Lady Gaga (*@ladygaga*), Beyonce (*@beyonce*), Taylor Swift (*@taylorswift*), Adele (*@adele*), and Madonna (*@madonna*). All use cases are well-known singers with verified accounts on Instagram.

4.5 Data Collection & Data Pre-Processing

These steps are completely described in Section 3.3.

4.6 Identification of Impersonating Accounts

4.6.1 Identifying Impersonators

Based on the methodology that I presented in [52], I measure the profile similarity of the publishers to identify impersonators across case studies. The methodology is based on the Instagram profile similarity and I consider major profile metrics:

- **Username (text)** is a string that individuals use on Instagram to define their profile address composed of 30 symbols. Username, must contain only letters, numbers, periods and underscores. For example, the usernames of genuine accounts in this paper are *@realdonaldtrump*, *@barackobama*, and *@emmanuelmacron*.

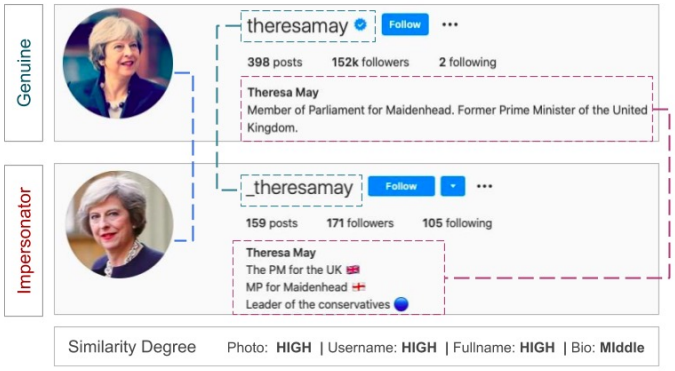


Figure 4.3: Identifying Impersonators through profile similarity.

- **Full name (text)** is what shows up on their profile page, as well as next to a user's comments. "*President Donald J. Trump*", "*Barack Obama*", and "*Emmanuel Macron*" are the Display names of the genuine accounts.
- **Biography (text)** is a section where users can include information about themselves and it is limited to 150 characters.
- **Profile Picture (image)** on Instagram represents the account personage. This photo, whether a profile is public or private, is visible to everyone. Impersonator accounts copy the same (or a very similar) photo as their profile picture.
- **Follower/Followee/Shared Media** are the activities of a user in terms of social connections and publications. Later I show fan pages are duplicating the post of the genuine account in their accounts.
- **Account age** which is obtained from the date of the first published post.

(*i*) For text metrics, I use the Cosine Similarity technique [98] and I define the minimum threshold to 30%. (*ii*) To measure the photo similarity, I use a convolutional neural network face detection in [101]. I compare the face of all accounts (if exist) to the face of the genuine users (*e.g.* R. Federer) and if the same person is detected, I mark it as similar photos. Eventually, if an account has at least 30% similarity in one of the text metrics or has a similar profile photo, I consider it as an impersonator. Otherwise, it is a non-similar account (*not* impersonator) and I exclude it from the dataset. (*iii*) In terms of activities and social connections, impersonators have different characteristics: fewer followers, higher followees, and fewer published posts. In addition, in most cased, the account is created after the creation of the genuine account.

In this thesis, I discover 1.6K impersonators with different levels of similarity.

4.6.2 Primary Account Analysis

I continue by making some primary analysis. Table 4.1 presents some of the fundamental differences between real accounts and impersonators. Impersonators tend to have few followers, but they follow many others. Normally, they do this to develop a network of relevant accounts (other impersonators) and increase their followers. Furthermore, in terms of the number of comments and likes per post, impersonators suffer from lower engagement. For example, with Trump, impersonators on average have 528 followers (vs. 16M), 1.1K followees (vs. 8), receive 27 comments per post (vs. 19.5K) and earn 690 likes per post (vs. 340K). I notice the same pattern for all others.

Table 4.1: Real Accounts vs. Impersonators

	follower		**followee**		**avg. #comment per post**		**avg. #like per post**	
use case	*Imp (avg)*	*real account*	*Imp (avg)*	*real account*	*Imp*	*real account*	*Imp*	*real account*
D. Trump	528	16M	1.1K	8	27.14	19.5K	690.14	340K
B. Obama	256	2.5M	446	14	40.00	13.5K	1.4K	1M
E. Macron	435	1.5M	738	91	12.45	3.8K	302.03	65K
B. Johnson	431	367K	318	254	11.78	600	274.14	15K
T. May	312	157K	253	1	2.21	350	54.25	5.6K
Ch. Ronaldo	432	197M	832	445	12.16	35K	1.6K	5.5M
L. Messi	447	140M	650	227	13.08	28K	2.8K	4.1M
R. Nadal	121	8.4M	513	65	12.17	2.5K	768.23	290K
R. Federer	189	7.1M	479	71	9.45	2.9K	670.12	400K
N. Djokovic	148	6.6M	236	777	6.67	1.5K	320.05	220K
Lady Gaga	7.2K	39M	653	46	5.46	19.5K	219.46	1.1M
Beyonce	130	138M	701	0	3.92	25.8K	353.18	2.9M
Taylor Swift	2.4K	125M	1.3K	0	4.84	0.0*	177.83	1.8M
Adele	5.3K	33M	459	0	3.76	12.7K	291.15	1.3M
Madonna	6.6K	14.7M	842	243	4.74	1.8K	134.45	98K

*T. Swift disabled comments.

4.6.3 Clustering

To find the potential hidden impersonators among the dataset, I use clustering technique. The whole process is explained in Figure 4.4. In general, at first, I use the impersonator dataset from section 3.3 as input. Next, based on "*profile characteristics*" and "*behaviour activities*", I run the clustering to find the potential clusters. I experiment this with a number of clustering methods: K-means, Gaussian Mixture Modeling, and Spectral Clustering. I find similar results with all techniques. So, for the rest of this section, results are based on the K-means algorithm.

The feature list used in clustering consist of several important metrics that are listed in Table 4.2. This process, identifies two clusters (the optimal number is obtained from the

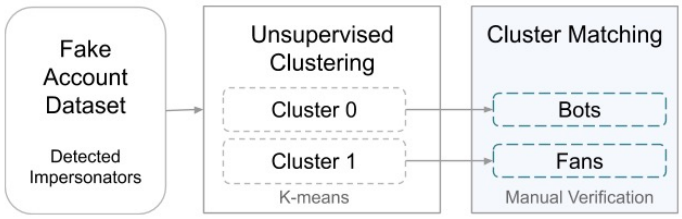


Figure 4.4: High-Level view of the process of discovering impersonators.

Elbow Method). The two clusters are highly diverse in profile characteristics and publishing behaviour (Table 4.3). Based on manual confirmation I match discovered clusters with types of impersonators defined in section 4.3. Inspection of these clusters reveals two clear populations:

Table 4.2: Clustering Feature Set.

similarity username	avg received like	follower
similarity full name	avg hashtag length	followee
similarity biography	avg caption length	media count
similarity photo	avg received comment	private
external url	account age	verified
MSF*	LSF*	

*The most and least number of features that the similarity exist.

Table 4.3 presents a summary of the distinctive characteristics of the two clusters that have been observed.

Table 4.3: Characteristics of the clusters.

Metrics	**Fans**	**Bots**
avg. username similarity per imp*	**0.49**	0.13
avg. full_name similarity per imp	**0.40**	0.18
avg. bio similarity per imp	0.25	0.18
avg. photo similarity per imp	**0.71**	0.17
the Least number of features that have similarity	1	1
the Most number of features that have similarity	3.32	1.53
avg. follower per imp	**101.6K**	16.5K
avg. followee per imp	757	927
avg. media count per imp	**808**	679
avg. received comment per post	**24.15**	10.01
avg. received like per post	**1.6K**	774

*Impersonator

Next, I analyse each cluster separately:

4.6.3.1 Impersonator Bots - Cluster 0

I believe this cluster captures bot entities (Section "4.3") that exist to achieve specific tasks. In this thesis, bots are fake entities that are programmed to publish pre-defined content as posts, use a particular network of hashtags, and target specific issues. Bots have a quite low similarity in all profile metrics (less than 20%) and the number of followers is almost 6 times fewer than fans (Table 4.3). However, the rate of post-distribution is higher in bots. One of the important metrics is the received attention per post (passive or active) and bots earned nearly half of fans (almost 10 comments and 770 likes).

4.6.3.2 Impersonator Fan Pages - Cluster 1

Based on assessing characteristics, I acknowledge that this cluster represents Fans. Fans spread content regarding a genuine figure (in favour of or against). There is nearly 50% similarity in the username, 40% in the full name, 20% in biography, and 70% similarity in profile photos. Moreover, they hold similarity at most in 3 metrics. The number of followers is higher than the bots (avg. 101.6K vs. 16.5K) and on average, each post got 24 comments and nearly 1.6K likes (Table 4.3).

4.6.4 Manual inspection for validation.

To validate the correctness of the proposed clustering, from each cluster I pick 80% of profiles (nearly 850 of the population) and check each one manually. Based on the definitions in Section 4.3, generally 112 accounts were identified incorrectly. As I were not sure if those accounts represent a bot character or a fan entity, I recognized them as outliers and excluded from the clusters. The rest of this thesis is based on these validated impersonators.

4.7 A Deep Neural Approach

I next exploit the above dataset to explore the possibility of automatically identifying impersonator posts. I believe that a bot, as a fake identity, also produces untrustworthy content and fake engagements. Likewise, fan pages, in some cases may distribute fake content *e.g.* a political fan page may publish rumours. So, I use the labelled data from the previous section and present a DNN classifier to distinguish content types. This classifier can predict whether a post is impersonator-generated (fan or bot) or genuine-generated. Note that I do not consider the question of classifying the veracity of information shared by the accounts.

I start by presenting the design of a Deep Neural Network classifier that can detect whether a post is Impersonator-generated or not. The entire process is shown in Figure 4.5.

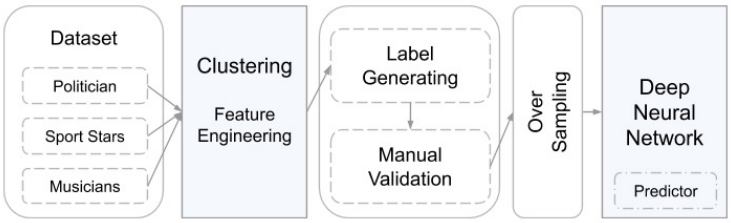


Figure 4.5: The high-level overview of impersonator-generated content predictor.

4.7.1 Dataset Overview

For classification, I use the post dataset obtained after clustering which is described in Section 4.6.3. This dataset consists of 10K post from 2.2K impersonators across 3 communities. Since I conduct manual annotation of impersonators, I am confident that posts are labelled correctly (Section 4.6.4).

4.7.2 Over-Sampling

The dataset is highly unbalanced: 31% genuine, 45% fan-generated, and 34% bot-generated post content. To solve this problem, I use the combination of Synthetic Minority Over-sampling Technique (SMOTE) [102] and Random Under-sampling algorithm [103]. So, I produce similar examples from the minor class to increase the total number and, meanwhile, I under-sample the major class and randomly remove some samples. The final dataset contains an equal amount of samples from class types. This helps me to increase the final accuracy by 8.5%.

4.7.3 Feature Engineering

I build a set of features from post metadata and profile metrics that help us to train the proper model. Table 4.4 summarizes features (with types) that are employed. I break the feature list into two principal categories: "*post features*" and "*publisher features*". Post features comprises all features that are obtained from the content of the post such as number of likes, number of comments, the caption, *etc.* . Publisher features are extracted from the profile of the publisher profile. To prepare the feature set, I directly use some features such as numbers. However, some others are derived from the content. For example, the account age is taken from the date of the first post and the profile similarities are calculated previously in section 4.6.1. Then, to do text vectorization, the caption text, user biography text, and other text metrics are vectorized using Keras Tokenizer [68] class with 30000 num_words. This class allows vectorizing a text corpus, by turning each text into either a sequence of integers.

Table 4.4: Feature Set used in Deep Neural Network.

Post Features		**Publisher Features**	
Feature	**Type**	**Feature**	**Type**
caption text	*text*	similarity username	*numeric*
caption topics (LDA)	*text*	similarity fullname	*numeric*
post hashtag	*text*	similarity bio	*numeric*
tagged users in post	*text*	profile biography	*text*
like count	*numeric*	similarity photo	*numeric*
comment count	*numeric*	follower/followee/post	*numeric*
tagged users count	*numeric*	full name	*text*
mention users count	numeric	biography	*text*
hashtag count	numeric	username	*text*
overall sentiment of caption	numeric	following followers ratio [104]	*numeric*
overall sentiment of hashtag	numeric	followers posts ratio	*numeric*
media type (image or video)	*numeric*	bio emoji count	*numeric*
emoji count	*numeric*	bio hashtag count	*numeric*
url/website exist	*numeric*		*numeric*
date	*numeric*		

4.7.4 Proposed DNN Architecture

Using the above feature set, I propose a Deep Neural Network architecture that exploits CNN, LSTM, BERT and Dence Layers to process post content and profile metadata (Figure 4.6). The complete structure is as follows:

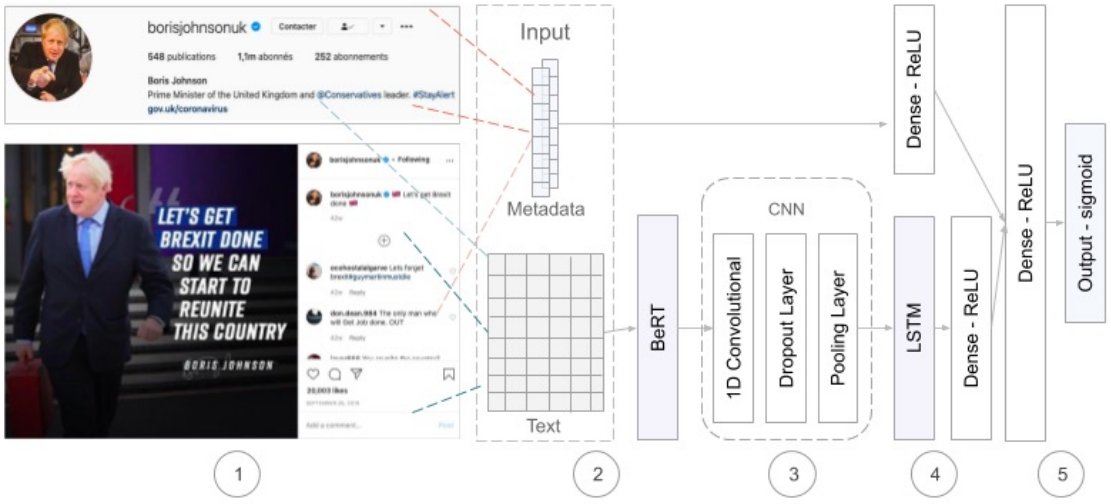


Figure 4.6: The proposed Deep Neural Network architecture to detect impersonator content.

1. First, in the input layer, I extract and pre-process all profile metrics and post features that are listed in Table 4.4. This architecture accepts two inputs types: (i) text content (*e.g.* post caption, hashtags, profile bio) which I combine them into a single

corpus. (*ii*) the metadata features (*e.g.* like, comment, follower, followee) that come from both profile and post content and then are transformed into a single vector.

2. Next, to transform the texts into a form amenable for processing, I adopt a pre-trained neural language model, Bidirectional Encoder Representations from Transformers (BERT). BERT is a text representation technique which is designed to pre-train deep bidirectional representations from the unlabeled text by jointly conditioning on both left and right context in all layers [105]. This results in an output vector by BERT (vectorized text) and then given as input to a CNN layer.

3. Then, the tokenized output of the BERT layer passes through a Convolution Neural Networks. This network contains 1D CNN with ReLU activation function (and 128 filters and a kernel size of 6) followed by a Dropout Layer (value of 0.2) for regularization, then a 1D Pooling Layer.

4. Then, (*i*) the result of CNN layer connects to a LSTM layer which processes vectorized text data and outputs a single 32-dimensions vector that is then fed forward through a ReLU activated Dense layer of size 16. (*ii*) Meanwhile, numerical metadata passes through a Dense Layer with ReLU activation of size 16.

5. Finally, I concatenate the output of the text and metadata layers into a single vector (size 32) that is then fed forward through a Dense layer with ReLU activation function and then an Output Layer. The final output layer forms the type of the post (bot, fan, genuine). I develop this model using Tensorflow and Keras Functional API [68]. I pick a random split of 75% (training set) and 25% (test set) and run with 10-Fold Cross-Validation. The Accuracy, Precision, Recall, and F1-Score results are listed in Table 4.5. I compare the proposed classifier with a tradition Random Forest Classifier.

Table 4.5: Performance of the proposed architecture

Model	**Accuracy**	**Precision**	**Recall**	**F1**
Random Forest Classifier	0.76	0.78	0.77	0.76
Proposed DNN (post)	0.78	0.79	0.76	0.78
Proposed DNN (post + profile)	0.83	0.82	0.83	0.82
Proposed DNN (post + profile) + BERT	**0.86**	**0.85**	**0.86**	**0.85**

The traditional RF Classifiers give approximately 77% in all metrics (text tokenized using TF-IDF). First, I do classification using the proposed DNN architecture with only 'post content' (CNN + LSTM), and I observe an increase in overall result by nearly 2% (Accuracy 78%). Then I re-run the classifier with both '*post content*' and '*profile metadata*' (CNN + LSTM). This helps to improve by almost 4.5% (Accuracy 83%). Finally, I add the

BERT layer to my architecture (BERT + CNN + LSTM). This step additionally assists us to improve the overall efficiency by almost 4%, and I eventually achieve the accuracy of 86% in detecting post type.

4.7.5 Feature Analysis

To gain an understanding of what characteristics are most prominent in prediction, I explored features and I found: (*i*) Impersonators usually have longer profile biography that contains more hashtag and mention (*ii*) genuine users tag and mention fewer people in posts and use a few hashtags compared to impersonators, (*iii*) Genuine contents receive much higher reactions (like and comment), (*iv*) Impersonators use a large number of emojis in profile biography and post caption, (*v*) In genuine users, the following_followers_ratio is remarkably lower than impersonators, and in contrast, the follower_post_ratio is greater. (*vi*) There are more numbers of neutral sentiment text (post and hashtag) in genuine contents. (*vi*) usually in impersonator-generated contents, the main genuine page is tagged or mentioned.

4.8 Assessing Published Content

First, I desire to know when impersonators are commenting on Instagram? And compared to others, what is the rate of publishing? So, Figure 4.7 presents the age comments that are published. Plot a) is the cumulative distribution of the age of the comments (hour) which compared imposters to the whole dataset. For better presentation, I limit the figure in the x-axis. Nearly 30% of the comments (for both) are posted in the first hour. As it continues, imposters comment more, and in the first 10 hours, they posted 80% of their total comments, while this number is around 60% for others. This means imposters, in term of commenting, are really engaged in the first 10 hours.

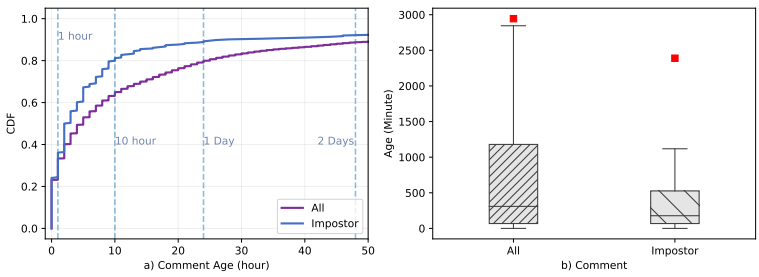


Figure 4.7: Age of the comments earned by all users vs imposters.

Figure 4.7.b is the Boxplot representation of the age of the comments (minutes). As it

can be seen, the range of all group is wider than imposters. Also, imposters commented by the minute 100 (median), while others posted by the minute 250. The average point for both groups is large. For better distinction of comments that are issued, I calculate the average published time of the comments per unique user and plot it on Figure 4.8. By considering the first hour, while on average 60% of the imposter's comments are issued, others are publishing less than 30% of their total. Furthermore, nearly 90% of the imposter's comments are published on the first day, but this number is around 80% for other. This means imposters are eager to comment really quick (abnormal activity). So, from the perspective of traffic management, they are producing huge network traffic and in a large-scale format that could contribute to traffic jams.

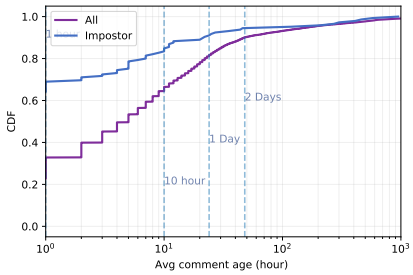
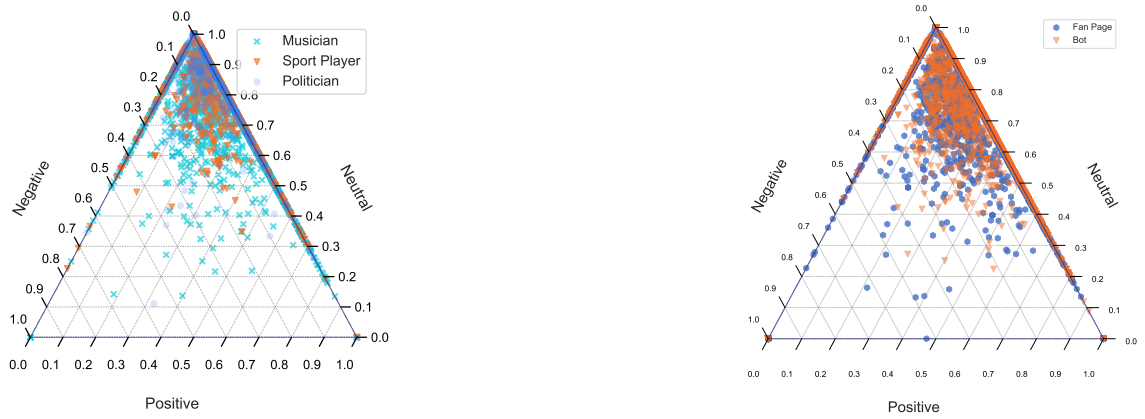


Figure 4.8: CDF of the average of the comments posted by unique users.

Next, by using the presented deep neural network classifier (Section 4.7), I classify impersonator dataset into bot-generated and fan-generated content. In general, bots and fans produce 4.8K and 5.2K posts. Next, I perform the following analysis: (*i*) I get the most discussed topics in captions (how relevant impersonators publish), (*ii*) I measure the text sentiment (the viewpoint), (*iii*) I investigate hashtag frequency (hashtags are an effective way to get more eyes and engagement [106]), and (*iv*) if possible hashtag topics. Meanwhile, I match the findings to the genuine content.

To obtain sentiments, I use the Afinn [107] and vaderSentiment [108] algorithms. I checked post frequency with [109] and validate it by manual inspection through the dataset. Figure 4.9 represents the polarity of posts as a Ternary plot. A post has three values in the positive, neutral and negative axis which sum to 1. I notice that fan-generated content have more numbers of negative post over positive ones, whereas bot-generated posts have a greater ratio of positive sentiments in captions. Among communities, musician and then politician have the most negative post sentiments.



(a) Community (b) Cluster

Figure 4.9: Ternary plot of the ratio of the positive, neutral and negative sentiment post caption across A) communities, and B) Clusters.

4.8.1 Politicians

Regarding post topics (Table 4.6) I observe, in some cases, impersonators target some specific events. For example, Trump generally talks about internal issues such as "*jobs*", "*election*", "*maga*" in 86% of posts. Meanwhile, fans publish relavant issues in 37% of posts, and mention his policies (positive or negative viewpoints) such as "*conservative*", " *make america great again*". On the other hand, Bots talk principally about "*support trump*", "*best president*", "*2020 election*" in 68% of posts (possitive sentiment). In terms of hashtag, bots focus on related upcoming US election hashtags such as "*trump 2020*", "*maga*" in 74% of posts (3.2 times more than fans). Additionally, bot posts in B. Johnson (59%), and T. May (34% of the posts) are about "*Brexit*" and "*vote election*" with the positive sentiment. Bots utilize "*election 2019*", "*brexit, Brexit memes*", and "*conservatives*" hashtags in 64% of posts.

4.8.2 Sports Players.

In football, fans and bots regularly post very relevant news, videos, and images to the real users in 80% of posts. Yet, bots invite the audience to be more engaged ("*comment*", "*follow*"). In tennis, the situation is different: While fans are covering sports events, still bots regularly promote sport wearings. I witness that in Nadal 37%, in Federer 39%, and in Djokovic nearly 31% of the posts are promotions. 63% of the bots, use call-to-action words of "comment" and "likeforlike". I also observe this habit in hashtags (Fig 4.11): bots promote sports brands in 49% of the posts in Federer ("*Nike*", "*Babolat*"), in 40% of the posts in Nadal ("*Wilson*", "*Uniqlo*", "*Rolex*") and in nearly 30% of the posts in Djokovic

Table 4.6: (A) Politician: Most Discussed Topics in posts. [color code: the real account: green, Fan: blue, Bot: red].

Use Case	Most Discussed Caption Topic
D. Trump	nation maga mexico jobs democrats iran repost donald trump obama conservative trumptrain suppor trump ivanka election best president
B. Obama	family healthcare memories michelle obama barack memes happy family donald trump religious michelle obama obama girls
E. Macron	carbone green ukraine planete actforaustralia politic paris germany armistic gorgeous macron handsome paris
B. Johnson	brexit vote conservative delay getbrexitdone referendum british brextention english memes elections campaigning brexit memes
T. May	nato, politic, great futur, conservatives conservatives photos brexit british mem theresa may style

("*Lacoste*").

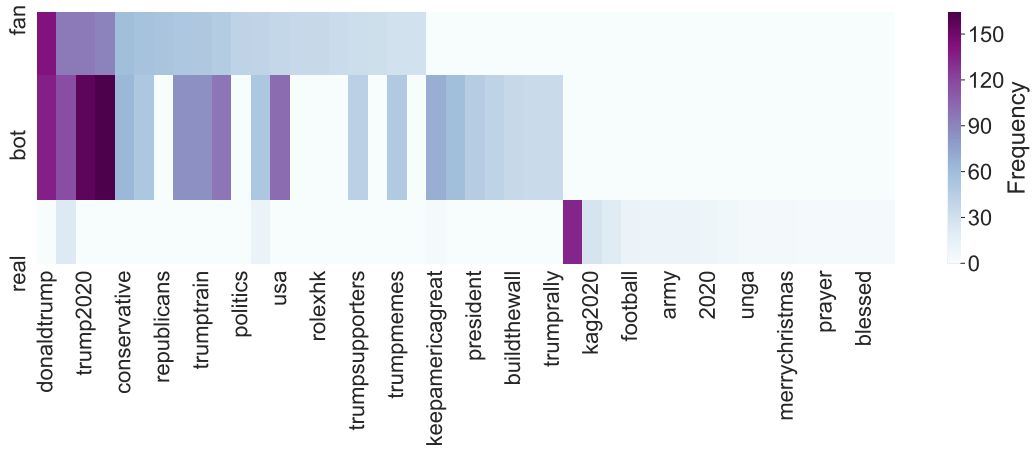


Figure 4.10: Heatmap of topmost hashtags for D. Trump.

4.8.3 Musicians

In this community, bots and fans published with the same positive sentiment rate. For example, in "Madonna", bots on average publish with 2.2 sentiment polarity among 535 posts, while fans, publish with an average of 1.95 in 578 posts. Fans cover more news surrounding each use case in almost 79% of their posts and topics as well as hashtags are more relevant to the genuine posts. For example, T. Swift and her fans address "*nation concert*" and "*her videos*" in most of the posts. Also, fans support T. Swift amid her

crisis with"*i stand with taylor*" hashtag. However, bots regularly address fashion styles to promote brands ("*look, beautiful*" in T. Swift, "*Louis Vuitton,*" in Madonna, "*tattoo*" in L. Gaga, "*ivy*" in Beyonce). Additionally, I witness bots ask the audience to be more active ("*like*", "*likeforlikeback*") in 72% of the posts.

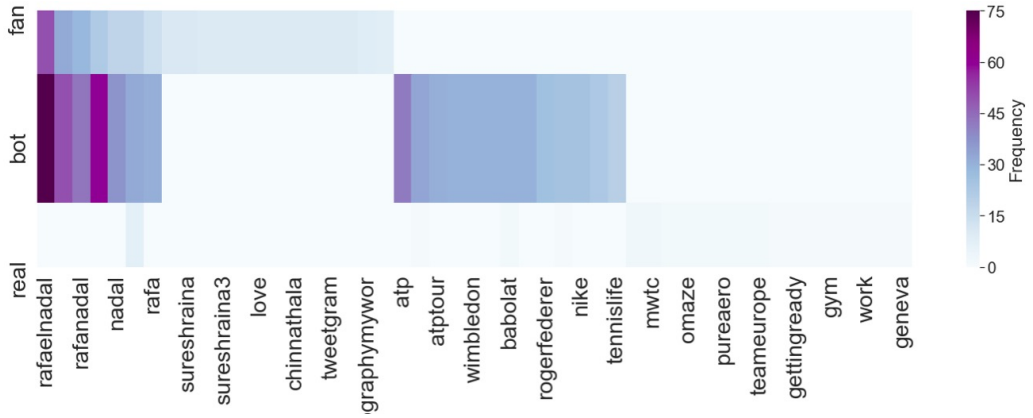


Figure 4.11: Hashtag Heatmap for R. Nadal.

4.9 Conclusion

This chapter focuses on fake identity detection on social media to propose an automated deep learning model in order to detect ingenuine content. It aims at impersonation problem and the challenge of identifying the impersonator generated content on social media.

In summary, (*i*) I used the fake identity dataset that I crawled from Instagram (impersonators). In this dataset, I collected public information in leading communities of politicians, sports stars, News agencies, and celebrities. Inside each community, several well-known figures have been selected (Section 3.3). (*ii*) Then, I proposed a technique to identify impersonators by leveraging profile similarities. Using this technique, I was able to detect more than 4K impersonator accounts across various communities with different levels of similarity in their profiles. (*iii*) Then, based on "profile characteristics" and "user behaviours", I clustered them as "Fan impersonator" and "Bot impersonator". This process helps me to identify unknown hidden groups inside impersonators. (*iv*) In order to track and analyse what they broadcast in the shape of posts, I proposed a Deep Neural Network model which can correctly classify post content as 'bot-generated', 'fan-generated', or 'genuine' content. This model accepts textual and visual input from profile, post, comments, hashtags, and metadata to analyse the genuineness of the published post. (*vi*) Then, I investigated the content produced by each group. Based on advanced NLP techniques, I observed discrete characteristics in publishing from bots and fans. The results of this study

help community better understanding the phenomena of bot-generated content in social media.

In the next chapter, I will mainly focus on Transfer Learning and Language Modeling on social media textual data to be able to detect fake content in online social networks with higher accuracy.

Chapter 5

Multi-Domain & Social Media Aware Language Modeling for Fake Content

Contents

5.1 Overview **81**
5.2 Background & Related Work **82**
5.2.1 Multi-domain learning 82
5.2.2 Transfer Learning for fake Content detection 83
5.2.3 Contextual Language Modeling 84
5.2.4 Multi-Domain Adaptation and Language Modeling 85
5.2.5 Social Media Context Aware Model 86
5.3 Fake RoBERTa for Social Media **87**
5.3.1 Architecture 87
5.3.2 Character-Level Tokenization 89
5.3.3 Data Corpus & preparation 90
5.4 Fake Content Classification using FakeRoBERTaSM **92**
5.4.1 Methodology 92
5.4.2 PreTraining Procedure 93
5.5 Evaluation Setup **95**
5.5.1 List of Models 95
5.5.2 Model Configuration 96
5.5.3 Evaluation Metrics 96
5.6 Experiments & Results **96**
5.6.1 Evaluation Datasets 96

5.6.2 Performance Analysis 97
5.7 Discussion & Conclusion 98

5.1 Overview

Natural Language Processing (NLP) helps empower machines to understand, process, and analyze human language [110]. Its has involves various tasks of the field, such as text classification [90], named entity recognition [111] or summarization [37,112]. In this area, Language Modeling (LM) is considered a central task of language understanding and language processing [9]. In contrast to traditional context-free text embedding techniques, transformer-based Pretrained Language Models (PLMs) use much deeper network architectures [10], and are pre-trained on much larger text corpora to learn contextual text representations. The self-attention mechanism is a key defining characteristic of Transformer models. So, textual data becomes more meaningful through a deeper understanding of its context, which in turn facilitates text analysis and mining. Bidirectional Encoder Representations from Transformers (BERT) is one of the first tranformer-based PLMs that has achieved state-of-the-art results in a broad range of NLP tasks [11]. BERT and other BERT-based transformers (*e.g.* RoBERTa, DistilBERT) are designed to pretrain deep bidirectional representations from unlabeled text and, then, be fine-tuned for downstream tasks [12]. In this process, pre-training is unsupervised (or self-supervised), and fine-tuning is supervised learning. Most PLMs are trained on general-domain text corpora (*e.g.* Wikipedia, Books) [12]. If the target domain is completely different from the general domain, the final task result could be poor. In this situation, I might consider adapting the PLM using domain-specific data. However, PLMs can be pretrained with multi-domain topics to increase the target task accuracy. For example, to address COVID-19-related textual challenges, a PLM can be adapted on medical literature.

Nowadays, we see a considerable linguistic differences between the language spoken on social media (*e.g.* daily conversation, Tweets, comments) and formal corpora (*e.g.* books, Wikipedia). Misspelling, new vocabularies, abbreviations, slang, *etc.* are some examples that could impose an impact on downstream NLP tasks. A major problem with statistical language models was the inability to deal well with synonyms or Out-of-Vocabulary (OOV) words that were not present in the training corpus [13]. However, proper word representation in transformers is an important step in order to process textual data. While various distributed word representations exist, few are capable of handling OOVs especially in daily conversations in social media. Character-level representation has been found useful for exploiting explicit sub-word-level information (such as prefixes and suffixes). Furthermore, it naturally handles OOV tokens easily [113].

In this chapter, I replace RoBERTa's tokenization layer with the Character-CNN to handle the OOV problem and then, compare the performance of the model with baseline models. I also argue that Language Models pre-trained on multi-domain corpora could

handle many contextual representations compared to other transformer models that are pre-trained on general-domain text corpora. In order to evaluate the proposed models on a downstream task, I consider multi-domain PLM adaptation on several major topics to address fake content classification on social media.

Fake content can be defined as a verifiably false piece of information shared intentionally to mislead readers [1] and has been used to create a political, social, and economic bias in the minds of people for personal gains. One main reason of disseminating many fake content on social media is that they often encourage impersonators, malicious accounts, trolls, and social bots to produce information [2] [3] without considering the credibility of the content as an attempt to entice users to read them [4]. Compared with traditional news, fake news attract readers and get rapid dissemination causing large-scale negative effects. The best example for this is that within the first three months of the USA presidential election 2016, fake news generated to favor both nominees was believed and shared by almost 37 million social media users [5]. Since social media content is relayed among users without filtering, editorial judgment, or fact-checking, it is required to introduce highly efficient models to detect fake content with high accuracy to control the spread of fake content on internet platforms. Due to the above reasons, Fake content detection on social media has recently become an active area of research. However, Fake content detection on social media is really challenging as they are inherently multilingual and in multiple forms such as textual, visual, and auditory forms. The lack of labeled data is another major challenge in exploring fake content on social media especially when using traditional machine learning-based models and algorithms. In addition, social media platforms have their own characteristics in terms of data types, user relations, user behaviors, and linguistic differences and which require special attention when handled at once. Furthermore, social media permit users to share information on a variety of topics such as memes, events, politics, health, and celebrities. Due to the scale of this problem, I argue that a semi-automatic approach is necessary to explore fake content shared on multiple social media platforms on different topics. I believe this helps users to gain a better understanding of the credibility of the information they observe on social media.

5.2 Background & Related Work

5.2.1 Multi-domain learning

Most existing fake content detection techniques are trained and evaluated using domain specific datasets (*e.g.* politics, news, entertainment), and therefore failed to identify fake news in real-world scenarios on multiple domains. Hence, previous works introduced multi-domain and cross-domain fake content detection strategies [114] [115] [116] [117]. Their

experiments on multi-domain fake news detection approaches demonstrated significant improvements on the fake news detection tasks compared with the baseline models. In the literature, there have been some studies that proposed unsupervised domain adaptation techniques for pre-training general models which do not require labeled target domain in NLP tasks [118]. When there is a domain shift, performance drops on the target, which undermines the ability of the models to generalize to real scenarios. As a result, multi-domain adaptation techniques performs best in real applications [116].

Wang et al. [119] proposed a multi-model approach to detect fake news which can derive event-invariant features and thus benefit the detection of fake news on newly arrived events. Their multi-model feature extractor is responsible for extracting both textual and visual features from social media posts. Jin el al. [120] proposed another multi-model approach for rumor detection on Twitter. They proposed a RNN with the attention mechanism to fuse features from text, image and social context for detecting rumors. The performance of the model was considerable higher compared to their baseline models. Apart from that, multi-model approaches were introduced as the integration of different models together to obtain better results.

In this chapter, I combine multi-domain learning strategies to build a deep model by integrating multiple models to archive better performance in fake content classification.

5.2.2 Transfer Learning for fake Content detection

There have been several attempts to apply transfer learning to fake news detection. Santiago González-Carvajal et al. studied the general comparison between BERT against traditional machine learning classification in [37]. They differentiate types of approaching NLP problems into two categories: a linguistic approach that generally uses different features of the text, and a machine/deep learning approach. From tokenizing perspective, [38] highlighted various challenges in the BERT model which if solved could significantly boost the model's accuracy, especially in domain-specific applications. With the advancement of Transformers in NLP tasks as reviewed in the above-mentioned research works, several recent works have used Transfer Learning for fake news detection.

Liu et al. [39] proposed a BERT-based method for fake news detection. They treated fake news detection as a fine-grained multiple-classification task and their model exhibited superior performance to the baselines and other competitive approaches. A data-driven BERT-based automatic fake news detection method was proposed by Heejung et al. [40]. This model analyzed the relationship between the headline and the body text of news. CT-BERT model was introduced in [41] which proposed an approach using the BERT-based ensemble model focused on COVID-19 fake news detection. Xiangyang et al. [42] have also proposed ensemble method of different pre-trained language models such as BERT, Roberta

and Ernie, targeting COVID-19 fake news detection. FakeBERT [43] is a BERT-based deep learning approach that integrates a deep convolution neural network having different kernel sizes and filters with the BERT. Their proposed model outperformed many other existing models with 98.9% accuracy. Khan et al. [44] conducted a benchmark study to assess performance 19 different machine learning models for fake news detection. Their experimental results show that BERT-based models have achieved better performance than all other models across datasets. For interested readers, Rogers et al. [45] reviewed how BERT works, what kind of information it learns and how it is represented, common modifications to its training objectives and architecture.

As BERT-based deep learning models perform better than many baseline models on fake news classification, in this chapter, I enhance Transfer Learning-based models on both pre-training and fine-tuning to improve fake content classification metrics.

5.2.3 Contextual Language Modeling

Since 2018 [12], we have seen the rise of a set of large-scale Transformer-based Pre-trained Language Models (PLMs) in the domain of NLP. Transformer-based models use deeper network architectures (e.g., 48-layer Transformers [144]), and are pre-trained on much larger text corpora to learn contextual text [11]. Contextual representations can further be unidirectional or bidirectional. For example, in the sentence "I accessed the bank account," a unidirectional contextual model would represent "bank" based on "I accessed the" but not "account." However, BERT represents "bank" using both its previous and next context — "I accessed the ... account" — starting from the very bottom of a deep neural network, making it deeply bidirectional.

Transformer-based PLMs such as BERT, RoBERTa, or DistilBERT are pre-trained on large corpora and can be fine-tuned to solve many NLP tasks [12]. During pre-training, the model is trained on unlabeled textual data which is an unsupervised (or self-supervised) task, and in the fine-tuning part, the pre-trained parameters are fine-tuned using labeled data (supervised task). With this technique, I get the word relationships in different context and use their weight to solve the downstream task. Below I describe base PLMs:

5.2.3.1 Google BERT

Google developed Bidirectional Encoder Representation from Transformers (BERT) based on bidirectional transformer [12]. It is one of the most widely used auto-encoding PLMs, and is the baseline for embedding models. The trend of using larger models and more training data continues [11]. The multi-layer architecture heavily relies on the original implementation that is described in [10]

BERT is trained on the BooksCorpus which is a large collection of free novel books dataset (800M words) and text passages of English Wikipedia (2,500M words). There are two different pre-trained model sizes for BERT: (1) BERT base, which is a BERT model consists of 12 layers of Transformer encoder, 12 attention heads, 768 hidden size, and 110M parameters. (2) BERT large, which is a BERT model consists of 24 layers of Transformer encoder,16 attention heads, 1024 hidden size, and 340 parameters. The pre-trained available model and code are available online[1,2].

Pre-training BERT is performed on two unsupervised tasks: (1) a 'Masked Language Model (MLM), where 15% of the tokens are randomly masked and replaced with the "[MASK]" token. Then, the model is trained to predict the masked tokens [45]. (2) A 'Next Sentence Prediction' (NSP) task, where the model is given a pair of sentences and is trained to identify when the second one follows the first. This task is meant to capture more long-term information [**?**,48].

5.2.3.2 ROBERTA

Robustly optimized BERT approach (RoBERTa) [121] is more robust version of the BERT. It makes a few changes to the famous BERT model and achieves some improvements. The changes include: (1) Training the model longer with larger batches and more data; (2) Removing the Next Sentence Prediction (NSP) objective; (3) Training on longer sequences; (4) Dynamically changing the masked positions during pretraining [122]. The base RoBERTa has 125M parameters, 12 layers, 12 attention heads, and 768 hidden units.

The optimized RoBERTa produces state-of-the-art results on the widely used NLP benchmark, General Language Understanding Evaluation (GLUE). RoBERTa is pretrained on five English-language corpora of varying sizes and domains, totaling over 160GB of uncompressed text including BookCorpusm CC-News, OpenWebText, and Stories [121]. In terms of word embedding, the authors of RoBERTa considered training BERT with a larger byte-level BPE vocabulary containing 50K subword units, without any additional preprocessing or tokenization of the input. The pre-trained RoBERTa code is available online[3].

5.2.4 Multi-Domain Adaptation and Language Modeling

Most pretraining efforts focus on general domain unlabelled corpora, such as news wire, Wikipedia, open source articles, and the broader Web. Such corpora lack domain-specific knowledge such as daily News, scientific topics, and daily user conversations on social media. A prevailing assumption is that even domain-specific pretraining can benefit by starting from

[1] https://github.com/google-research/bert
[2] https://github.com/huggingface/transformers
[3] https://github.com/pytorch/fairseq

general-domain language models [123]. However, I argue that considering several domains together, can increase the knowledge of LM for the downstream task. Formally, a domain is defined as:

$$D = \{X, P(X)\} \tag{5.1}$$

where X is the feature space (*e.g.* the text representations), and $P(X)$ is the marginal probability distribution over that feature space. A task (here is text classification) is defined as:

$$T = \{Y, P(Y|X)\} \tag{5.2}$$

where Y is the label space. Estimates for the prior distribution $P(Y)$ and the likelihood $P(Y|X)$ are learned from the training data $\{(x_i, y_i)\}_{i=1}^{n}$. Domain adaptation aims to learn a function f from a source domain D_S that generalizes well to a target domain D_T, where:

$$P_S(X) \neq P_T(X) \tag{5.3}$$

Typically in NLP, domain is meant to refer to some coherent type of corpus. This may relate to topic, style, genre, or linguistic register [118]. For example, there are some models that have been developed including BioBERT (biomedical sciences) [124], SciBERT (scientific publications) [125], FinBERT (financial communciations) [126], and ClinicalBERT (clinical notes) [127].

5.2.5 Social Media Context Aware Model

Applying a general-based BERT model to domain-specific issues especially on social media such as fake News, hate speech, impersonification and others, is problematic since there is a disconnect between the language as found in general open-source corpora and that daily spoken language on social media. In addition, informal language speaking requires proper modeling in order to obtain the best of transfomer-based pretrained models.

Domain specific corpora also often contain a large amount of jargon that can be misspelled frequently [38]. From a linguistic perspective [128], social media has changed the way we speak and write. I summarize important existing challenges that are related to this chapter as follows:

- **Limited Text length.** Sentences and paragraphs are shorter. The text limit is 280 characters on Twitter, and 2200 characters on Instagram (captions longer than 125 characters will be truncated). In addition, Writers may also use incomplete sentences or ellipses (. . .) to make points. I also witness the same limitation for comments [129].

- **Acronyms** are an abbreviation form from the initial letters of other words. This is a commonplace substitutes to whole sentences. For example, LOL (laugh out loud), OMG (Oh my God), and TTYL (talk to you later) are well-known acronyms in English.

- **Emojis** are a pictoral representatin of something, such as a smile or frown . This technique is used to show what the user is feeling or to express the intended tone. This could be a lazy form of writing, but social media is a fast and convenient way of interacting with an audience.

- **Domain-Specific Out-Of-Vocabulary (OOV) words** BERT relies on the WordPiece algorithm [130] to create the vocabulary, that chooses those sub-word units for the vocabulary. However, the tokenization using this vocabulary is not done semantically. This leads to poor tokenization that induces semantic information loss in terms of dealing with OOV words for domain centric downstream tasks [38].

- **Slang & Colloquialisms** are words and phrases that are regarded as informal, which are more common in speech than writing. Typically these are restricted to a particular context or group of people. For example, "*It was raining cats and dogs*", "*dope!*", or "*No biggie, Sally*".

- **New Vocabulary** is often introduced to social media, and most is not included in formal corpora (*e.g.* books or Wikipedia) at the time of first appearance. For example, in 2019, the UK Prime Minister, Boris Johnson, utilized "Brexit (Britain Exit)", "GetBrexitDone", "backboris", and "TakeBackControl" [131]. The same happened in the United States in 2016 when Donald Trump used "make America great again (MAGA)", "keep America great", and "border control" hashtags [132].

- **Grammar** is often misused on social media. Thus, posts often contain sentences without adherence to grammatical rules and syntax [133].

5.3 Fake RoBERTa for Social Media

In this Section, I introduce the pretrained transformer-based language model called FakeRoBERTaSM.

5.3.1 Architecture

The proposed pretrained RoBERTa-based transformer or "*Fake RoBERTa for Social Media (FakeRoBERTaSM)*", is constructed based on the architecture of the RoBERTa_base

transformer [121]. I select RoBERTa as it is a more robust version of the BERT. Some important optimization steps are performed on RoBERTa: (1) dynamically changing the masked positions, and (2) removing the NSP objective (Section 5.2.3.2) which makes it a suitable model for my task. The authors of [44], have shown that RoBERTa obtains higher accuracy compared with other ML and TL models on fake news detection task. However, I replace the default tokenization module with the Character CNN module to tackle various social media challenges on textual data (Section 5.2.5). The overall architecture of FakeRoBERTaSM is shown in Figure 5.1.

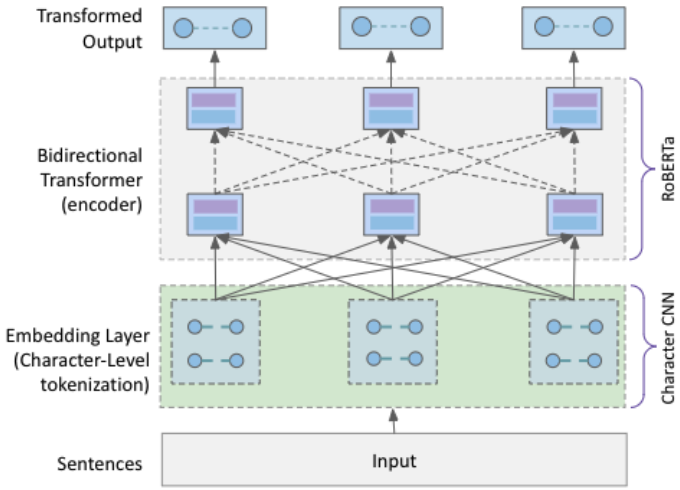


Figure 5.1: FakeRoBERTaSM: The underlying architecture is based on the RoBERTa [121] and the tokanization part is replaced with Character CNN.

FakeRoBERTaSM is trained on a corpus of XXM data containing several general and specific topics (Section 5.3.3). In order to adapt with the informal writing style on Online Social Networks, the majority of the data corpus is obtained from social media platforms including Twitter and Instagram. In general, two data types are used: (1) Long text-length sizes data from Wikipedia and News media websites with formal English writing style to capture word relations in the formal English language. (2) Social media short text-length sizes data with informal English writing style from Twitter and Instagram platforms (280 chars on Twitter and 2200 chars on Instagram). This part is necessary to capture the word relations in daily conversations on social media (*e.g.* tweets, comments, posts).

In addition, to improve the awareness of the proposed model concerning fake content textual data, I include several datasets from social media that contain fake content data (Table 5.1). FakeRoBERTaSM is a tranformer-based and context-aware language model that is pretrained and maintained for social media (social media aware), and could be used for fake content NLP task-specific problems on OSNs. The training procedure, underlying

infrastructure, and parameters are described in Section 5.4.2.

5.3.2 Character-Level Tokenization

In this chapter, I use a Character-CNN [9, 134] module used in ELMo's architecture [135] which helps the characters of a token to produce a single representation [136]. ELMo employs a character convolutional neural network (CNN) to construct the word representations based on character embeddings, which not only successively mitigates the out-of-vocabulary (OOV) problem but also reduces the number of parameters [137]. While there are some downsides of using character-level tokenization [138, 139], I strongly believe this approach can significantly alleviate the problem of "unknown tokens" and OOVs in text.

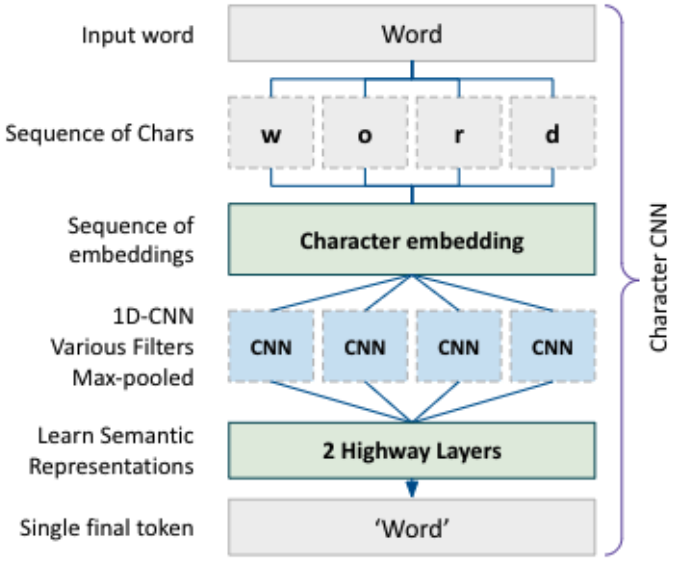


Figure 5.2: High-level diagram of the word representation with Character CNN [9, 134, 140] used in FakeRoBERTaSM.

This module constructs context-independent token representations. In pretraining procedure and in MLM task, instead of predicting single wordpieces, entire words are predicted. Using this technique, it is possible to learn a subword vocabulary of modest size that can encode any input without getting "unknown" (UNK) tokens. It will start building its vocabulary from an alphabet of single chars, so all words will be decomposable into tokens [137, 140]. As suggested by [141], instead of spliting tokens into wordpieces (default BERT tokenizer), each input token is assigned a single final contextual representation by the model. Figure 5.2 illustrates the overall process of word tokenization utilizing Character CNN technique.

Input. First, tokens are converted into a sequence of characters (UTF-8). Boukouri et al. [141] proposed a maximum sequence length of 50. Then, each character is represented by a vector.

Feature Extraction Layer. Next, sequence of vectors are fed forward to multiple 1D-CNN [142] networks. 1 to 10 kernels with 8, 16, 32, 64, and 128 filters are used. Then, outputs are max-pooled and concatenated.

Highway. Next, the output is fed to 2 highway layers [143]. The result, then, is projected down to a final embedding size which is aligned with 512-dimensional of the RoBERTa model.

Table 5.1: List of Datasets Used in Pretraining of the FakeRoBERTaSM.

Domain	Dataset	Platform	Description	Size	Refrence
General	General Corpus	Wikipedia	A resampled portion of official Wikipedia English database.	4.1M pages	144
	News Dataset 1	News Web	A collection of daily English News from different sources between 2017 and 2020.	790K text	145
	News Dataset 2	News Web	A collection of daily English News from different sources.	100K	Manual Crawling*
	Public Dataset 1	Twitter	Publicly available geotagged tweets.	10K tweets	146
	Public Dataset 2	Instagram	English public posts from various publishers.	14K posts	147
	Public Dataset 3	Twitter	Public tweets.	19.5M Tweets	Manual Crawling*
	Public Dataset 4	Instagram	Public posts/comments.	923K posts 3.2M comments	Manual Crawling*
Celebrity	Twitter Dataset 2	Twitter	Top 20 most followed users.	53K tweets	148
	Insta Influencers	Instagram	Various English speaking influencers (nano/macro/micro/mega)	21K posts 830K comments	149
	Celebrities	Instagram	Various English speaking celebrities.	9.3M comments	Manual Crawling*
Fake Content	Impersonator	Instagram	Ingenuine post content from impersonators on Instagram.	10K posts 60K comments	3
Health	English COVID_19	Instagram	English Covid_19 public post from different sources.	25K posts 820K comments	150
Politics	US/UK Politicians	Instagram	Comments on the US presidents and the UK prime ministers.	3.3M comments	Manual Crawling*

*I crawled the data manually and it is not published publicly.

5.3.3 Data Corpus & preparation

To improve the quality of the proposed approach and to capture the awareness of the social media content, I made attempts to add extra data to the base model. In contrast to BERT [12] and RoBERTa [121] (base models that are pretrained on a large corpus of formal text data), I follow a different strategy. My pretraining step relies on social media textual data. This is because of the linguistic differences between formal corpora and informal daily English-language writings on social media platforms (Section 5.2.5).

Multi-Domain Topics. Figure 5.3 illustrates the high-level scheme of the various data sources that are used for pretraining as follows: (1) A part of the data corpus is captured from web-based News media and Wikipedia pages (Light English version). In this part, I obtain the word relation in the formal English-language style. These resources consist of "*general topics and articles*" with formal English-language writing style. (2) A part of the textual data is obtained directly from several most used social network platforms including Twitter and Instagram. From Instagram, I include posts and comments, and from

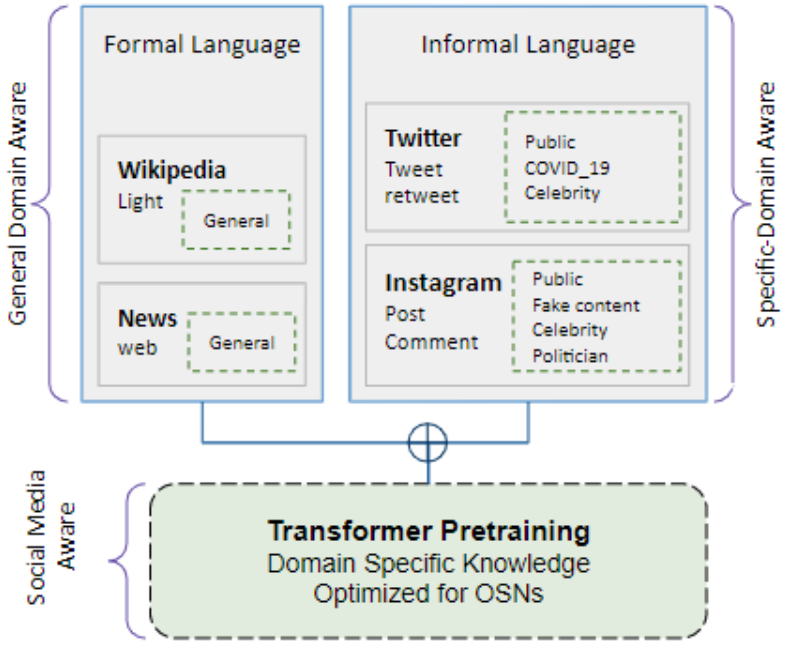


Figure 5.3: Multi-Domain and Social Media Aware Language Model Pretraining.

Twitter, I consider tweets and retweets. In social media, I focus on multi-domain topics: *"COVID_19", "celebrity content", "fake content", "political content", "daily content", and "public data"*. From linguistic perspective, the style of English-language writing is informal, daily, and contains many misspelling as it is distributed by ordinary users. All the challenges are described in Section 5.2.5. (3) Eventually, I combine these datasets to be used for the model pretraining.

Table 5.1 details the list of datasets and their characteristics. 39% of data is from formal textual data (*e.g.* Wikipedia, News websites), and 61% is from informal text posted on social media. It contains XXM words with different sentence lengths. In social media, 82.7% of the records are in the shape of comments or retweets.

Text Pre-processing. I apply several light text pre-processing steps as follows: (1) I lowercase all records, (2) I replace URL links with the "url" word, (3) I replace emails with the "email" word, (4) I replace usernames (word starts with @) with the "entity" word, (5) I remove newline signs, (6) I map emojis into a word [151], and (7) I remove duplicate records. To keep the informal nature of writing, I do not remove punctuation and other unknown characters such as abbreviations and OOVs (Section 5.2.5).

5.4 Fake Content Classification using FakeRoBERTaSM

Next, with the aim of classifying fake content data on social media (task-specific), I fine tune the FakeRoBERTaSM language model, and propose a deep neural network architecture which is presented in Figure 5.4. The proposed deep learning model consists of three different layers: (1) FakeRoBERTaSM (Section 5.3), which is used as the encoder and the text embedding layer to perform the first feature extraction process. (2) Several multi-headed Convolutional Neural Networks (CNN) with different kernels, which are used as a second feature extractor layer to obtain a better understanding of the textual vectors/relations. (3) A final 1D-CNN, which is used as a third feature extraction layer to concatenate the prior outputs.

5.4.1 Methodology

Figure 5.4 shows the underlying model architecture. The layers are as follows:

Input Layer. First, the textual parts of a post (Section 5.3) such as the main text (caption or tweet) and hashtags are combined together. Words in a sentence can be represented as in Equation 5.4. Given a text sequence W of length X, I apply pre-processing steps that are explained in Section 5.3.3 to get the normalized input sentences:

$$W_j = \{w_1, ..., w_{X_j}\} \tag{5.4}$$

Tokenization Layer. Character-level token representations (Character CNN) are applied on input sentences (Section 5.3.2). The first token x_1 is always the [CLS] token. If X contains a sentence pair $(X1, X2)$, I separate the two sentences with a special token [SEP].

Transformer Layer. Next, the pretrained transformer FakeRoBERTaSM is used with t number of Layers. This number is equal to number of layers in the base RoBERTa. Different hidden layers can capture different kinds of information of the text, and the last four hidden layers of BERT-Based transformers are good for extracting information in a feature-based approach [12]. So, I map the input representation vectors into a sequence of contextual embedding vectors in Equation 5.5. R^T is the contextualized representations of the input tokens.

$$R^T = \{r_1, ..., r_x\} \in \mathbb{R}^{X \times dim} \tag{5.5}$$

2nd Feature Extraction Layer. Next, I perform the second feature extraction on the textual data. I pass the word embedding through convolutions with kernel sizes 2, 3, 4, and 5 and $d_T = 512$ to extract more information from different sets of the word vectors for

prediction. I experiment with various combinations for the number of CNNs and hyper-parameters, and this combination obtained the best performance (Section 5.5.2). Multiple kernels are applied to form a random number of feature maps. Feature maps provide an insight into the internal representations and reflect characteristics of the specific sentence. So, each kernel can be considered as a different feature extractor. The size and number of kernels are the two main tuning parameters of the convolution operation. R is then fed into a convolutional layer. For each l-words embedding:

$$u_i = [r_i, ..., r_{i+l-1}] \in \mathbb{R}^{l \times dim}; 0 \leq i \leq X - 1 \tag{5.6}$$

For each filter $f_j \in \mathbb{R}^{l \times dim}$ $\langle u_i, f_i \rangle$ is calculated and the convolution results in matrix $F \in \mathbb{R}^{X \times m}$:

$$F_{ij} = \langle u_i, f_j \rangle \tag{5.7}$$

3rd Feature Extraction Layer. Next, I concatenate the CNN outputs (Equation 5.8). The superscript 2, 3, 4 and 5 are kernel sizes. Then, I pass the result into an additional 1D-CNN layer ($d_T = 512$) (Equation 5.7) with residual connections and obtain the output T_m. This step helps to perform the third step of textual feature extraction. By performing several experiments, I obtain the best result with one CNN (Section 5.5.2).

$$P = P^2 \oplus P^3 \oplus P^4 \oplus P^5 \tag{5.8}$$

Connection Layers. Next, to flatten T_m, I pass it through two fully connected layers, $d_T = 512$ and $d_T = 32$, and get the final vector size of the textual representation T_f. At the end, I use a sigmoid activation function for classification.

$$Y_{reliable/unreliable} = sigmoid(T_f) \tag{5.9}$$

5.4.2 PreTraining Procedure

I conducted my experiments on Google Colab Pro (CPU: Intel(R) Xeon(R) CPU @ 2.20GHz; RAM: 25.51 GB; GPU: Tesla P100-PCIE-16GB with CUDA 10.1). Each model was trained on the training set for 5 epochs and evaluated on the validation set. Datasets are presented in Section 5.6.1. The models are optimised using AdamW [152] with a learning rate of 0.001, max sequence length of 512 tokens, and a batch size of 60. Loss and accuracy were calculated through the pretraining procedure. For every 10000 training steps, I save a checkpoint.

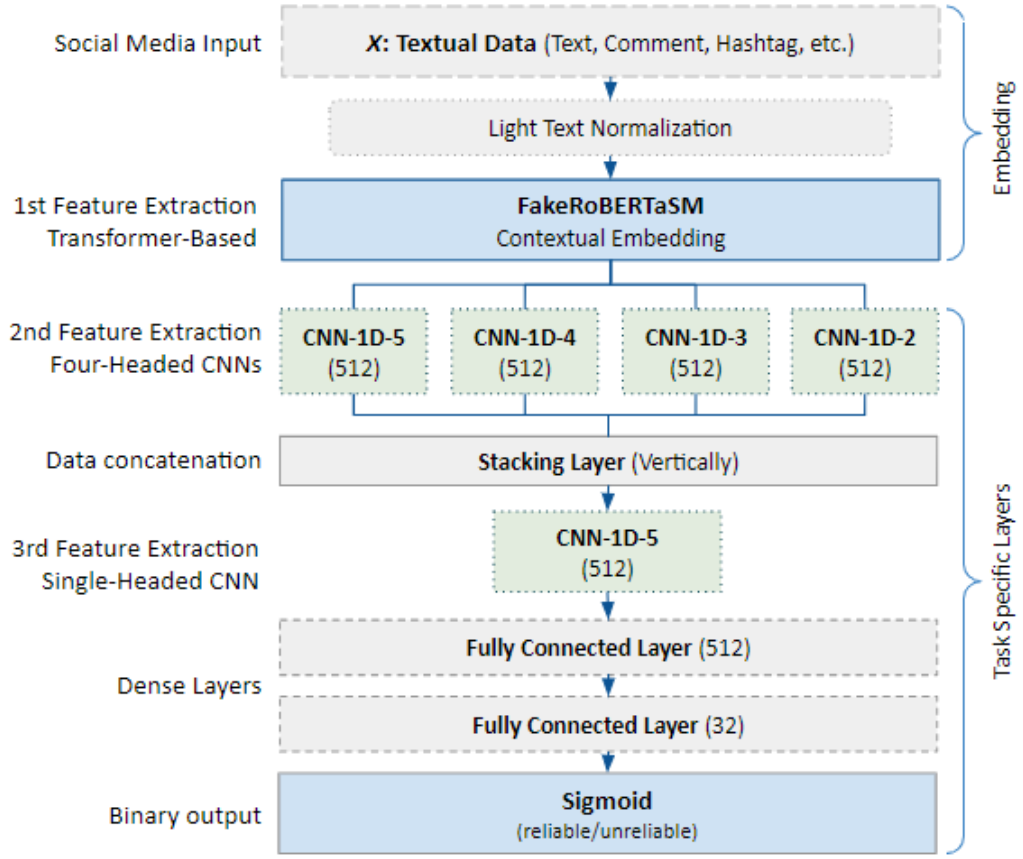


Figure 5.4: The proposed deep learning architecture for fake content classification. FakeRoBERTaSM is used as the first feature extractor. To obtain better textual relationships, a Four-Headed 1D-CNNs for the 2nd feature extraction layer and a Single-Headed 1D-CNN for the 3rd feature extraction layer have been utilized.

I implemented my models using Keras [68], Tensorflow [153], Pytorch [154] and Huggingface's Transformers [155] libraries. For text preprocessing, I used NLTK [156]. In this chapter, to find the optimum hyper-parameter values, several experiments are performed which is described in Section 5.5.2.

For the task of fake content classification, I use the Binary Cross-Entropy loss function as the objective (BCE). BCE loss allows this model to assign independent probabilities to the labels.

5.5 Evaluation Setup

In this section I present list of models that are being used for model comparison alongside their configurations.

5.5.1 List of Models

I compare my results against several alternative baseline models. I divide the models into types of "*Baseline*" and "*Multi-Model*" architectures and below, I describe each one separately:

Baseline Models:

- ***Model 1: RoBERTa.*** In this model I use RoBERTa_base as the embedding layer and the output is directly fed forward to a binary dense layer with the Sigmoid activation. This is the baseline model as my proposed pretrained model, FakeRoBERTaSM, is based on this. This architecture is called "Model 1".

- ***Model 2: BERT.*** In this model, I use BERT_base as the embedding layer and the output is directly fed forward to a binary dense layer with the Sigmoid activation. This architecture is called "Model 2".

- ***Model 3: FakeRoBERTaSM.*** In this model I use FakeRoBERTaSM (Section 5.3) as the embedding layer and the output is directly fed forward to a binary dense layer with the Sigmoid activation. I call this architecture the "Model 3".

Multi-Models (fine-tuned):

- ***Model 4: RoBERTa_ base + Proposed Model.*** In this model, I use the proposed deep learning model architecture illustrated in Figure 5.4, but I replace the transformer layer (FakeRoBERTaSM) with RoBERTa_base baseline [10, 12]. This architecture is called the "Model 4".

- ***Model 5: BERT_ base + Proposed Model.*** In this model, I use the proposed deep learning model architecture illustrated in Figure 5.4, but I replace the transformer layer (FakeRoBERTaSM) with BERT_base baseline [10, 12]. I call this the "Model 5".

- ***Model 6: FakeRoBERTaSM + Porposed Model.*** In this model, I use the full deep learning model architecture presented in Figure 5.4. For the embeddings, my

pretrained language model, FakeRoBERTaSM (Section 5.3) is used. In this model, there is a Four-Headed 1D-CNN layer as the second feature extractor, and one 1D-CNN layer as the third feature extractor layer (Section 5.4). For simplicity, I call this "Model 6".

5.5.2 Model Configuration

Regarding hyperparameters, I perform tuning using random search on Dataset 1. (1) I randomly select hyperparameters. (2) For each selection, dataset is splitted into 80% train and 20% test sub-datasets. (3) To avoid over fitting, I apply 10-fold cross-validation. (4) Next, I train each model on the train dataset with epoch size of 5 and use it on the test dataset to analyse the performance. (5) I repeat this process for all proposed models (Section 5.5.1). I consider following parameters: (1) A Four-Headed CNNs for the 2nd feature extraction layer and one 1D-CNN for the 3rd feature extraction layer are selected; (2) For the 2nd feature extractor layer, kernel sizes of 2, 3, 4, and 5 are selected; (3) For the 3rd feature extractor layer, a kernel size of 5 is selected; (4) The filter size of 32 for all CNNs is selected; (5) The batch size of 60 is chosen. (6) The number of epoch is set to 5.

5.5.3 Evaluation Metrics

To measure the performance of the classification output, I use accuracy, precision, recall, F1-score, and Mathews Correlation Coefficient (MCC).

5.6 Experiments & Results

To evaluate my fake content classifier, I utilise various independent datasets. The details of datasets are described in Section 5.6.1. Next, I discuss the baseline methods in Section 5.5.1. Finally, the performance evaluation is presented in Section 5.6.2.

5.6.1 Evaluation Datasets

I use several publicly available fake content/news datasets that are prepared from different sources of social media and web.

Dataset 1: Political Fake Content. This dataset contains news articles and is part of the Kaggle competition [157] for identifying unreliable News. It contains 10K reliable and 10K unreliable articles regarding the US election.

Dataset 2: COVID_19 Fake Content. This dataset contains fake News data from Twitter about the COVID-19 health crisis, published by [158]. There are 3K fake and 3.3K real news items.

Dataset 3: Fake Content Data. This dataset contains fake content, including headlines and articles [159]. This dataset contains 3.1K fake and 3.2K real articles.

Table 5.2: Datasets Used to Benchmark Models.

Dataset	**Dataset 1**	**Dataset 2**	**Dataset 3**
Task	Classification	Classification	Classification
No. Total Posts	20759	6420	6335
No. Fake Posts	10373	3060	3164
No. Real Posts	10386	3360	3171
No. Total Tokens	68M	173K	4.8M
Max sentence len	24K	1.4K	20K
Mean sentence len	774	27	765

Table 5.2 summarizes the datasets used for evaluation. The data is preprocessed according to the methodology explained in the Section 5.3.3. I perform a 10-fold cross-validation technique to split the dataset into the training and testing subsets and evaluate model prediction performance. Details of the hyperparameters and metrics is described in Section 5.5.3.

Table 5.3: Performance comparison of the proposed multi-model (Model 6) with other models (baselines and multi-models) across three datasets. All results are compared with the output of Model 1. (+) and (-) indicates whether the model increased the evaluation metric or not. Results in bold indicate the best values among all.

	Model	**Architecture**	**Precision**	**Recall**	**F1-score**	**Accuracy**	**MCC**	**Specificity**
			Dataset 1					
Baseline	Model 1	*RoBERTa_base* [121]	0.940	0.940	0.940	0.940	0.883	**0.981**
	Model 2	*BERT_base* [12]	0.971 (+)	0.971 (+)	0.971 (+)	0.971 (+)	0.944 (+)	0.970 (-)
Proposed*	Model 3	*FakeRoBERTaSM*	0.966 (+)	0.977 (+)	0.967 (+)	0.966 (+)	0.933 (+)	0.968 (-)
Multi-Model	Model 4	*RoBERTa + Multi-Headed 1D-CNNs + 1D-CNN*	0.951 (+)	0.951 (+)	0.951 (+)	0.951 (+)	0.902 (+)	0.931 (-)
	Model 5	*BERT + Multi-Headed 1D-CNNs + 1D-CNN*	0.970 (+)	0.962 (+)	0.970 (+)	0.969 (+)	0.942 (+)	0.962 (-)
	Model 6	*FakeRoBERTaSM + Multi-Headed 1D-CNNs + 1D-CNN*	**0.981** (+)	**0.982** (+)	**0.980** (+)	**0.981** (+)	**0.952** (+)	0.970 (-)
			Dataset 2					
Baseline	Model 1	*RoBERTa_base* [121]	0.890	0.890	0890	0.890	0.785	0.929
	Model 2	*BERT_base* [12]	0.921 (+)	0.920 (+)	0.921 (+)	0.920 (+)	**0.854** (+)	0.875 (-)
Proposed*	Model 3	*FakeRoBERTaSM*	0.921 (+)	0.921 (+)	0.921 (+)	0.921 (+)	0.847 (+)	0.929 (=)
Multi-Model	Model 4	*RoBERTa + Multi-Headed 1D-CNNs + 1D-CNN*	0.852 (-)	0.871 (-)	0.852 (-)	0.861 (-)	0.719 (-)	0.807 (-)
	Model 5	*BERT + Multi-Headed 1D-CNNs + 1D-CNN*	0.920 (+)	0.920 (+)	0.920 (+)	0.920 (+)	0.853 (+)	**0.948** (+)
	Model 6	*FakeRoBERTaSM + Multi-Headed 1D-CNNs + 1D-CNN*	**0.923** (+)	**0.922** (+)	**0.923** (+)	**0.923** (+)	0.847 (+)	0.946 (+)
			Dataset 3					
Baseline	Model 1	*RoBERTa_base* [121]	0.850	0.871	0.850	0.850	0.721	0.936
	Model 2	*BERT_base* [12]	0.952 (+)	0.952 (+)	0.952 (+)	0.952 (+)	0.891 (+)	0.949 (+)
Proposed*	Model 3	*FakeRoBERTaSM*	0.951 (+)	0.951 (+)	0.951 (+)	0.951 (+)	0.901 (+)	**0.959** (+)
Multi-Model	Model 4	*RoBERTa + Multi-Headed 1D-CNNs + 1D-CNN*	0.882 (+)	0.881 (+)	0.882 (+)	0.882 (+)	0.753 (+)	0.876 (-)
	Model 5	*BERT + Multi-Headed 1D-CNNs + 1D-CNN*	0.911 (+)	0.911 (+)	0.911 (+)	0.910 (+)	0.826 (+)	0.872 (-)
	Model 6	*FakeRoBERTaSM + Multi-Headed 1D-CNNs + 1D-CNN*	**0.961** (+)	**0.960** (+)	**0.961** (+)	**0.961** (+)	**0.928** (+)	0.952 (+)

Proposed*: The proposed pre-trained language model.

5.6.2 Performance Analysis

Table 5.3 lists the full comparative results of the presented models and baselines across three datasets. Several important observations can be seen.

By considering baseline models (Model 1 and 2) and the proposed FakeRoBERTaSM (Model 3): (*i*) Interestingly, while the underlying architecture of the FakeRoBERTaSM is based on the Model 1, but FakeRoBERTaSM performs better in all metrics than the Model 1 across all datasets. (*ii*) In dataset 1, Model 2 obtained the best results. However, In other datasets, the results are close to Model 3. For example, the recall values in dataset 2 are 0.920 and 0.921 for Model 2 and Model 3 respectively.

By considering multi-models (Model 4, 5, and 6): (*i*) The proposed multi-model FakeRoBERTaSM (Model 6), obtains the best scores all metrics across all datasets. However, the best specificity in dataset 2 is for Model 5. (*ii*) Compared to Model 1 (the baseline), all multi-models increased the final score results in precision, recall, F1-score, accuracy, and MCC. However, Model 4 in dataset 2 shows an opposite behaviour (result in red).

In general, the proposed FakeRoBERTaSM multi-model (model 6) obtains the best result in precision, recall, F1-score, accuracy, and MCC across all datasets.

5.7 Discussion & Conclusion

In this section, I discuss related challenges that might affect this chapter. In model comparison, I consider the "*base*" version of BERT and RoBERTa transformers to perform experiments. Base models are pretrained using smaller corpora compared to "*large*" versions. Furthermore, FakeRoBERTaSM, which is mostly pretrained on textual corpora from social media, has the best result for fake content classification task on the same data type and optimized for daily informal conversations. So, it could have a weak performance in other domains and long-length datasets.

In conclusion, this chapter presents a pre-trained language model called FakeRoBERTaSM which is pretrained from scratch. This PLM is optimized with several English-language data corpora from the web and social media to catch different types of information and writing styles. In this model, I use the base configurations of the RoBERTa model but I replaced the tokenization part with the Character CNN to overcome unknown tokens on social media. Next, FakeRoBERTaSM is used in a deep neural network model for the downstream task of fake content classification on social media. This model holds three layers of textual feature extractors to catch more word and sentences relations. The effectiveness of the presented model is tested on three different benchmark datasets. By performing experiments, my model is effective in classifying reliable/unreliable posts on social media.

Chapter 6

Conclusion and Future Work

Contents

6.1 Conclusion **100**

6.1.1 Summary and Insights of Contributions 100

6.2 Future Work and Challenges **102**

6.1 Conclusion

Fake content detection on social media is really challenging as they are inherently multilingual and in multiple forms such as textual, visual, and auditory forms. Social media platforms have their own characteristics in terms of data types, user relations, user behaviors, and linguistic differences and which require special attention when handled at once. Furthermore, social media permit users to share information on a variety of topics such as memes, events, politics, health, and celebrities. Due to the scale of this problem, I argue that a semi-automatic approach is necessary to explore fake content shared on multiple social media platforms on different topics.

This thesis makes timely and constructive contributions to social media content analysis in terms of (*i*) providing valuable datasets in different topics, (*ii*) detecting impersonator accounts, (*iii*) detecting ingenuine published content on social media, and (*iv*) providing a social media aware language model based on transfer learning for fake content classification tasks.

To that end, I first have collected proper datasets from social media by implementing a dedicated crawler to study the behaviour of fake identities on online social networks. Then I studied the problem of impersonation on social media and proposed a method to detect these fake identities on Instagram. I have studied impersonator's behaviour in terms of user behaviours, profile characteristics, and user engagements. Then, I proposed a deep neural network model in order to detect ingenuine content which is published by impersonator accounts. Next, I have expanded my work on textual data on social media and proposed a transfer learning-based pretrained language model which is optimized for social media textual data and could be used for the fake content detection task. I have considered low-level linguistic differences between formal and informal English language that is spoken on social media to reduce the problem of Unknown tokens in pretraining language models.

6.1.1 Summary and Insights of Contributions

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

6.1.1.1 Social Media Content Analysis & Datasets

This contribution aims at providing an insight into the user-generated content in social media focusing on posts and engagements of public profiles in various communities including politicians, sports stars, celebrities, and News agencies. I designed and implemented a dedicated crawler which is able to collect data from social media platforms. I used this crawler to prepare three different datasets in line with my main research direction that is

fake content and fake identity detection on social media. Public data and public pages are considered in all data collection processes. Three datasets are about: (*i*) impersonation in different communities on social media, (*ii*) influencers and sponsored content, and (*iii*) COVID_19 related content during the first lockdown on Instagram. For each dataset, I provided a comprehensive description of user behaviours, user engagements, and distributed content. Some of datasets are released for the research community for further researches with respect to GDPR rules, and are available in GitHub.

First, I studied the activity of influencers in terms of publishing sponsored content on Instagram. I considered posts, reactions, social connections, and visual data in order to analyse how many of influencers have income from their content. Then, I proposed a deep neural network to identify declared sponsored posts of influencers to understand what percentage are escaping from paying taxes [**?**]. In another dataset, I collected the public content of COVID_19 related posts during the first lockdown in 2020 [**?**]. I crawled posts, comments, users, communities, and well-known pages. Then I analysed the behaviour of different publishers: normal people, celebrities, News agencies, and fake identities. For example, the majority of bots publish off-topic content (with regards to COVID_19). They exploit the COVID_19 hashtags to spread their content (4.9% of posts). Or, the number of reactions to trusted publishers (Celebrities, News Agencies, and Business Pages) is 110x times higher than unreliable publishers. This highlights the importance of trustworthy accounts in critical moments. The last dataset, impersonators [**?**], is described completely in a separated chapter.

6.1.1.2 Impersonation on Social Media

This contribution focuses on the impersonation problem and the challenge of identifying the impersonator-generated content on social media platforms. To summarize: (*i*) I used the impersonator dataset that I crawled from Instagram. In this dataset, I collected public information in leading communities of politicians, sports stars, News agencies, and celebrities. Inside each community, several well-known figures have been selected (Section 3.3). (*ii*) Then, I proposed a technique to identify impersonators by leveraging profile similarities. Using this technique, I was able to detect more than 4K impersonator accounts across various communities with different levels of similarity in their profiles. (*iii*) Then, based on "profile characteristics" and "user behaviours", I clustered them as "Fan impersonator" and "Bot impersonator". This process helps me to identify unknown hidden groups inside impersonators. (*iv*) In order to track and analyse what they broadcast in the shape of posts, I proposed a Deep Neural Network model which can correctly classify post content as 'bot-generated', 'fan-generated', or 'genuine' content. This model accepts textual and visual input from profile, post, comments, hashtags, and metadata to analyse the genuineness of

the published post. (*vi*) Then, I investigated the content produced by each group. Based on advanced NLP techniques, I observed discrete characteristics in publishing from bots and fans. The results of this study help community better understanding the phenomena of bot-generated content in social media.

6.1.1.3 Social Media Aware Language Modeling

This contribution aims at providing a proper language modeling for textual data in social media platforms. In addition, the fake content classification task is one of the greatest challenges of researchers on Online Social Networks that could be addressed using Pre-trained Language Models. In this contribution: (*i*) I leveraged RoBERTa to propose a pretrained transformer-based language model, called FakeRoBERTaSM, which is pre-trained from scratch and optimized for social media textual data to overcome "*informal English-language textual*" challenges. (*ii*) Next, to overcome "*unknown tokens*" on daily conversations on social media, I used the Character CNN model which is a character-level tokenization technique. I replace this model with the default characterization level in the base RoBERTa. (*iii*) Finally, I proposed a fine-tuned and multi-domain deep learning architecture that is optimized for fake content classification on social media. The experimental results shows that the deep model architecture trained with FakeRoBERTaSM embedding, performed better than the remaining baseline models considered in my analyses.

6.2 Future Work and Challenges

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

In the direction of impersonation on social media, I believe the accuracy of the Deep Neural Network model that is presented to classify genuine and impersonator-generated content can be improved by looking into comments, stories, Reels as well as leverage the connection between impersonators to see the impact of bots on the propagation of fabricated content. From another perspective, it is priceless to understand what percentage of the impersonator-generated content can be considered as a type of fake information. In addition, this study could be extended by focusing on the user behaviour of impersonators in terms of publishing content and engagements.

In the direction of influencers and their activities on social media, I have two specific lines of future work. First, I wish to expand my analysis across multiple platforms (*e.g.* Tik Tok, YouTube) and to gain a deeper understanding of the strategies employed by influencers. Second, I wish to revisit my classifier to improve performance. Although the current implementation obtains good results, I manually found 11% mis-classifications. For exam-

ple, I posit that my classifier may be susceptible to mis-classification of genuine personal endorsements. Consequently, I wish to expand my training dataset and complement it with further manual annotations to identify key behaviour traits important in the classification. Statista [160] reports that the global Instagram influencer marketing industry was worth $2.38 billion in 2019 and like any other trades, there exist *Instagram Tax*. As far as social media influencers are independent contractors, they must pay self-employment tax (SE tax) [161]. However, it is not possible to precisely know how much money does an influencer make, but I revealed how different groups of influencers distribute *hidden advertisements*. This is a serious growing concern for governments.

In my ongoing study on fake content language modeling on social media, there are some important research directions: (*i*) the work can be extended towards addressing the linguistic challenges. Nowadays, we see a considerable linguistic differences between the language spoken on social media (*e.g.* daily conversation, Tweets, comments) and formal corpora (*e.g.* Book, Wikipedia). Misspelling, new vocabularies, abbreviations, slang, *etc.* are some examples that could impose an impact on downstream NLP tasks. Considering more low-level features could increase the output of the language models. (*ii*) The other direction could be considering larger corpora with various topics for the task of language model pretraining. However, power and computing resource limitations are the major bottleneck that require more investigations. (*iii*) It is possible to consider post metadata and images to increase the accuracy rather than focusing only on the textual content. However, it requires more complex architecture and different embedding layers. (*iv*) In addition, user information, user relations, and user behavior could absolutely be helpful to decrease fake content detection errors. Therefore, this information will be considered in future analyses to improve the fake content classification accuracy. (*v*) The proposed FakeRoBERTaSM model is applied to detect only the fake content. However, in the future, the FakeRoBERTaSM model will be applied to several other downstream tasks, mainly on online social media textual content classification. Apart from that, pruning methods will be used to compress my proposed model architecture to identify attention heads that are important for classification and prune unimportant heads from the model.

References

[1] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. Fake news detection on social media: A data mining perspective. *SIGKDD Explor. Newsl.*, 19(1):22–36, sep 2017.

[2] Emilio Ferrara, Onur Varol, Clayton Davis, Filippo Menczer, and Alessandro Flammini. The rise of social bots. *Communications of the ACM*, 59(7):96–104, 2016.

[3] Koosha Zarei, Reza Farahbakhsh, Noël Crespi, and Gareth Tyson. Impersonation on social media: A deep neural approach to identify ingenuine content. In *2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 11–15, 2020.

[4] Praboda Rajapaksha, Reza Farahbakhsh, and Noel Crespi. Bert, xlnet or roberta: The best transfer learning model to detect clickbaits. *IEEE Access*, 9:154704–154716, 2021.

[5] Hunt Allcott and Matthew Gentzkow. Social media and fake news in the 2016 election. *Journal of economic perspectives*, 31(2):211–36, 2017.

[6] Pradeep Kumar Roy and Shivam Chahar. Fake profile detection on social networking websites: A comprehensive review. *IEEE Transactions on Artificial Intelligence*, 1(3):271–285, 2020.

[7] Koosha Zarei, Reza Farahbakhsh, and Noel Crespi. How impersonators exploit instagram to generate fake engagement?, 2020.

[8] uslegal. https://definitions.uslegal.com/c/criminal-impersonation/, 2019.

[9] Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. Exploring the limits of language modeling, 2016.

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

[11] Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu, and Jianfeng Gao. Deep learning–based text classification: A comprehensive review. *ACM Comput. Surv.*, 54(3), apr 2021.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv*, abs/1810.04805, 2019.

[13] Daniel W. Otter, Julian R. Medina, and Jugal K. Kalita. A survey of the usages of deep learning for natural language processing. *IEEE Transactions on Neural Networks and Learning Systems*, 32(2):604–624, 2021.

[14] Monther Aldwairi and Ali Alwahedi. Detecting fake news in social media networks. *Procedia Computer Science*, 141:215–222, 2018.

[15] Apurva Wani, Isha Joshi, Snehal Khandve, V Wagh, and R Joshi. Evaluating deep learning approaches for covid19 fake news detection. In *Combating Online Hostile Posts in Regional Languages during Emergency Situation: First International Workshop, CONSTRAINT 2021, Collocated with AAAI 2021, Virtual Event, February 8, 2021, Revised Selected Papers*, page 153. Springer Nature, 2021.

[16] L. Caruccio, D. Desiato, and G. Polese. Fake account identification in social networks. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 5078–5085, Dec 2018.

[17] Devakunchari Ramalingam and Valliyammai Chinnaiah. Fake profile detection techniques in large-scale online social networks: A comprehensive review. *Computers & Electrical Engineering*, 65:165 – 177, 2018.

[18] Emilio Ferrara, Onur Varol, Clayton Davis, Filippo Menczer, and Alessandro Flammini. The rise of social bots. *Commun. ACM*, 59(7), 2016.

[19] Supraja Gurajala, Joshua S White, Brian Hudson, Brian R Voter, and Jeanna N Matthews. Profile characteristics of fake twitter accounts. *Big Data & Society*, 3(2):2053951716674236, 2016.

[20] Saeedreza Shehnepoor, Mostafa Salehi, Reza Farahbakhsh, and Noel Crespi. Netspam: A network-based spam detection framework for reviews in online social media. *IEEE Transactions on Information Forensics and Security*, 12(7):1585–1595, Jul 2017.

[21] Cao Xiao, David Mandell Freeman, and Theodore Hwa. Detecting clusters of fake accounts in online social networks. In *Proceedings of the 8th ACM Workshop on Artificial Intelligence and Security*, AISec '15, pages 91–101. ACM, 2015.

[22] Zafar Gilani, Reza Farahbakhsh, Gareth Tyson, Liang Wang, and Jon Crowcroft. Of bots and humans (on twitter). In *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*, ASONAM '17, pages 349–354, New York, NY, USA, 2017. ACM.

[23] Zafar Gilani, Reza Farahbakhsh, Gareth Tyson, and Jon. Crowcroft. A large-scale behavioural analysis of bots and humans on twitter. *ACM Trans. Web*, 13(1):7:1–7:23, February 2019.

[24] Yixuan Li, Oscar Martinez, Xing Chen, Yi Li, and John E. Hopcroft. In a world that counts: Clustering and detecting fake social engagement at scale. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, pages 111–120, Republic and Canton of Geneva, Switzerland, 2016. International World Wide Web Conferences Steering Committee.

[25] Indira Sen, Anupama Aggarwal, Shiven Mian, Siddharth Singh, Ponnurangam Kumaraguru, and Anwitaman Datta. Worth its weight in likes: Towards detecting fake likes on instagram. In *Proceedings of the 10th ACM Conference on Web Science*, WebSci '18. ACM, 2018.

[26] Francesco Buccafurri, Gianluca Lax, Serena Nicolazzo, and Antonino Nocera. Comparing twitter and facebook user behavior. *Comput. Hum. Behav.*, 52(C):87–95, November 2015.

[27] Bang Hui Lim, Dongyuan Lu, Tao Chen, and Min-Yen Kan. #mytweet via instagram: Exploring user behaviour across multiple social networks. IEEE/ACM ASONAM '15, pages 113–120. ACM, 2015.

[28] Ali Choumane, Zein Al Abidin Ibrahim, and Bilal. Chebaro. Profiles matching in social networks based on semantic similarities and common relationships. In *Proceedings of the International Conference on Compute and Data Analysis*, ICCDA '17, pages 14–18. ACM, 2017.

[29] Katharina Krombholz, Dieter Merkl, and Edgar". Weippl. Fake identities in social media: A case study on the sustainability of the facebook business model. *Journal of Service Science Research*, 4(2), Dec 2012.

[30] Oana Goga, Patrick Loiseau, Robin Sommer, Renata Teixeira, and Krishna P. Gummadi. On the reliability of profile matching across large online social networks. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, pages 1799–1808. ACM, 2015.

[31] Xinyi Zhou, Atishay Jain, Vir V Phoha, and Reza Zafarani. Fake news early detection: A theory-driven model. *Digital Threats: Research and Practice*, 1(2):1–25, 2020.

[32] Verónica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, and Rada Mihalcea. Automatic detection of fake news. *arXiv preprint arXiv:1708.07104*, 2017.

[33] Xinyi Zhou and Reza Zafarani. A survey of fake news: Fundamental theories, detection methods, and opportunities. *ACM Computing Surveys (CSUR)*, 53(5):1–40, 2020.

[34] William Yang Wang. " liar, liar pants on fire": A new benchmark dataset for fake news detection. *arXiv preprint arXiv:1705.00648*, 2017.

[35] Tian Bian, Xi Xiao, Tingyang Xu, Peilin Zhao, Wenbing Huang, Yu Rong, and Junzhou Huang. Rumor detection on social media with bi-directional graph convolutional networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 549–556, 2020.

[36] Kai Shu, Suhang Wang, and Huan Liu. Beyond news contents: The role of social context for fake news detection. In *Proceedings of the twelfth ACM international conference on web search and data mining*, pages 312–320, 2019.

[37] Santiago González-Carvajal and Eduardo C. Garrido-Merchán. Comparing bert against traditional machine learning text classification, 2021.

[38] Anmol Nayak, Hariprasad Timmapathini, Karthikeyan Ponnalagu, and Vijendran Gopalan Venkoparao. Domain adaptation challenges of BERT in tokenization and sub-word representations of out-of-vocabulary words. In *Proceedings of the First Workshop on Insights from Negative Results in NLP*, pages 1–5, Online, November 2020. Association for Computational Linguistics.

[39] Chao Liu, Xinghua Wu, Min Yu, Gang Li, Jianguo Jiang, Weiqing Huang, and Xiang Lu. A two-stage model based on bert for short fake news detection. In *International Conference on Knowledge Science, Engineering and Management*, pages 172–183. Springer, 2019.

[40] Heejung Jwa, Dongsuk Oh, Kinam Park, Jang Mook Kang, and Heuiseok Lim. exbake: Automatic fake news detection model based on bidirectional encoder representations from transformers (bert). *Applied Sciences*, 9(19):4062, 2019.

[41] Anna Glazkova, Maksim Glazkov, and Timofey Trifonov. g2tmn at constraint@ aaai2021: exploiting ct-bert and ensembling learning for covid-19 fake news detection. In *International Workshop on Combating On line Ho st ile Posts in Regional Languages dur ing Emerge ncy Si tuation*, pages 116–127. Springer, 2021.

[42] Xiangyang Li, Yu Xia, Xiang Long, Zheng Li, and Sujian Li. Exploring text-transformers in aaai 2021 shared task: Covid-19 fake news detection in english. In *International Workshop on Combating On line Ho st ile Posts in Regional Languages dur ing Emerge ncy Si tuation*, pages 106–115. Springer, 2021.

[43] Rohit Kumar Kaliyar, Anurag Goswami, and Pratik Narang. Fakebert: Fake news detection in social media with a bert-based deep learning approach. *Multimedia tools and applications*, 80(8):11765–11788, 2021.

[44] Junaed Younus Khan, Md Tawkat Islam Khondaker, Sadia Afroz, Gias Uddin, and Anindya Iqbal. A benchmark study of machine learning models for online fake news detection. *Machine Learning with Applications*, 4:100032, 2021.

[45] Anna Rogers, Olga Kovaleva, and Anna Rumshisky. A primer in bertology: What we know about how bert works, 2020.

[46] Stephen E. Robertson. Understanding inverse document frequency: on theoretical arguments for idf. *J. Documentation*, 60:503–520, 2004.

[47] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics.

[48] Faiza Khan Khattak, Serena Jeblee, Chloé Pou-Prom, Mohamed Abdalla, Christopher Meaney, and Frank Rudzicz. A survey of word embeddings for clinical text. *Journal of Biomedical Informatics*, 100:100057, 2019. Articles initially published in Journal of Biomedical Informatics: X 1-4, 2019.

[49] Instagram. Official api graph instagram. https://developers.facebook.com/docs/graph-api/, September 2021.

[50] Instagram. Instagram hashtag search. https://developers.facebook.com/docs/instagram-api/guides/hashtag-search, February 2020.

[51] Instagram. api access tokens. https://developers.facebook.com/docs/facebook-login/access-tokens/#usertokens, September 2021.

[52] Koosha Zarei, Reza Farahbakhsh, and Noel Crespi. Deep dive on politician impersonating accounts in social media. In *2019 IEEE Symposium on Computers and Communications (ISCC) (IEEE ISCC 2019)*, Barcelona, Spain, June 2019.

[53] Word Ninja Github. https://github.com/keredson/wordninja., 2019.

[54] R. Smith. An overview of the tesseract ocr engine. In *Proceedings of the Ninth International Conference on Document Analysis and Recognition - Volume 02*, ICDAR '07, page 629–633, USA, 2007. IEEE Computer Society.

[55] Matthew Honnibal and Ines Montani. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear, 2017.

[56] ASA. An influencer's guide to making clear that ads are ads, 2018.

[57] US Federal Trade Commission. Us federal trade commission. https://www.ftc.gov/news-events/press-releases/2017/09/csgo-lotto-owners-settle-ftcs-first-ever-complaint-against.

[58] Meeyoung Cha, Hamed Haddadi, Fabricio Benevenuto, and Krishna P Gummadi. Measuring user influence in twitter: The million follower fallacy. In *AAAI conference*, 2010.

[59] Instagram. What are the requirements to apply for a verified badge? https://help.instagram.com/312685272613322, September 2019.

[60] Json Hjh. Instagram stories vs feed ads. which is more effective at driving traffic? https://www.agorapulse.com/social-media-lab/instagram-stories-ads, Feb 2018.

[61] Hashtagify. https://hashtagify.me/hashtag/coronavirus, January 2020.

[62] Best Hashtags. http://best-hashtags.com/hashtag/coronavirus/, January 2020.

[63] Mathieu Bastian, Sebastien Heymann, and Mathieu Jacomy. Gephi: An open source software for exploring and manipulating networks, 2009.

[64] Zafar Gilani, Reza Farahbakhsh, Gareth Tyson, and Jon Crowcroft. A large-scale behavioural analysis of bots and humans on twitter. *ACM Transactions on the Web (TWEB)*, 13(1):1–23, 2019.

[65] Sneha Kudugunta and Emilio Ferrara. Deep neural networks for bot detection. *Information Sciences*, 467:312–322, 2018.

[66] Mariam Orabi, Djedjiga Mouheb, Zaher Al Aghbari, and Ibrahim Kamel. detection of bots in social media: A systematic review. *Information Processing & Management*, 57(4):102250, 2020.

[67] Koosha Zarei, Damilola Ibosiola, Reza Farahbakhsh, Zafar Gilani, Kiran Garimella, Noel Crespi, and Gareth Tyson. Characterising and detecting sponsored influencer posts on instagram, 2020.

[68] François Chollet et al. Keras. `https://keras.io`, 2015.

[69] Data Policy. Instagram data policy. https://help.instagram.com/519522125107875, March 2020.

[70] Chengcheng Shao, Pik-Mai Hui, Lei Wang, Xinwen Jiang, Alessandro Flammini, Filippo Menczer, and Giovanni Luca Ciampaglia. Anatomy of an online misinformation network. *PLOS ONE*, 13(4):1–23, 04 2018.

[71] Kai-Cheng Yang, Christopher Torres-Lugo, and Filippo Menczer. Prevalence of low-credibility information on twitter during the covid-19 outbreak, 2020.

[72] The top ten most-followed news accounts on Twitter. https://www.pressgazette.co.uk/the-top-ten-most-followed-news-accounts-on-twitter/, Jan 2020.

[73] List of news agencies. shorturl.at/E0126, Jan 2020.

[74] Statista. Instagram accounts with the most followers worldwide 2020. https://www.statista.com/statistics/421169/most-followers-instagram/, Nov 2020.

[75] Social Book. Top 100 Most-Followed Instagram Accounts. https://socialbook.io/instagram-channel-rank/top-100-instagrammers, Nov 2020.

[76] Trackalytics. The most followed instagram profiles. https://www.trackalytics.com/the-most-followed-instagram-profiles/page/1/, January 2020.

[77] 20 Most followed brands on Instagram in 2019. https://blog.unmetric.com/most-followed-brands-instagram, Jan 2020.

[78] Instagram. Stand our with instagram. https://business.instagram.com/getting-started, Jan 2020.

[79] Instagram. Instagram business account. https://www.facebook.com/business/profiles, September 2020.

[80] BBC News. Coronavirus pandemic. https://www.bbc.com/news/coronavirus, Feb 2020.

[81] Euronews. Special coronavirus. https://www.euronews.com/special/coronavirus, Feb 2020.

[82] Time. Time covid-19 track. https://time.com/tag/covid-19/, Feb 2020.

[83] Skynews. Covid-19. https://news.sky.com/topic/covid-19-8518, Feb 2020.

[84] FoxNews. Latest coronavirus headlines. https://www.foxnews.com/category/health/infectious-disease/coronavirus, Feb 2020.

[85] Ahmed Al-Rawi and Vishal Shukla. Bots as active news promoters: A digital analysis of covid-19 tweets. *Information*, 11(10), 2020.

[86] Tim Ken Mackey, Jiawei Li, Vidya Purushothaman, Matthew Nali, Neal Shah, Cortni Bardier, Mingxiang Cai, and Bryan Liang. Big Data, Natural Language Processing, and Deep Learning to Detect and Characterize Illicit COVID-19 Product Sales: Infoveillance Study on Twitter and Instagram. *JMIR PUBLIC HEALTH AND SURVEILLANCE*, 6(3):360–376, JUL-SEP 2020.

[87] Instaloader. Instaloader. https://github.com/instaloader/instaloader, January 2020.

[88] Christian Grimme, Mike Preuss, Lena Adam, and Heike Trautmann. Social bots: Human-like by means of human control?, 2017.

[89] Loredana Caruccio, Domenico Desiato, and Giuseppe Polese. Fake account identification in social networks. *2018 IEEE International Conference on Big Data (Big Data)*, pages 5078–5085, 2018.

[90] Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura Barnes, and Donald Brown. Text classification algorithms: A survey. *Information*, 10(4), 2019.

[91] Ayon Chakraborty and Jyotirmoy Sundi. Spam : A framework for social profile abuse monitoring.

[92] Kayode Sakariyah Adewole, Nor Badrul Anuar, Amirrudin Kamsin, and Arun Kumar Sangaiah. Smsad: A framework for spam message and spam account detection. *Multimedia Tools Appl.*, 78(4):3925–3960, feb 2019.

[93] Philip N. Howard and Bence Kollanyi. Bots, #strongerin, and #brexit: Computational propaganda during the UK-EU referendum. *CoRR*, abs/1606.06356, 2016.

[94] Philip N. Howard, Samuel Woolley, and Ryan Calo. Algorithms, bots, and political communication in the us 2016 election: The challenge of automated political communication for election law and administration. *Journal of Information Technology & Politics*, 15(2):81–93, 2018.

[95] Jessica Baldwin-Philippi. The myths of data-driven campaigning. *Political Communication*, 34(4):627–633, 2017.

[96] Stefan Stieglitz, Florian Brachten, Björn Ross, and Anna-Katharina Jung. Do social bots dream of electric sheep? a categorisation of social media bot accounts, 2017.

[97] Bob Burg Bryan Kramer Jay Baer Kim Garst David Meerman Scott Mark Schaefer Sue Zimmerman Tyler J. Anderson Jon Mitchell Jackson, Chris Brogan. *The Ultimate Guide to Social Media For Business Owners, Professionals and Entrepreneurs*. Independently published, 2018.

[98] Baoli Li and Liping Han. Distance weighted cosine similarity measure for text classification. In *Intelligent Data Engineering and Automated Learning – IDEAL 2013*, pages 611–618, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[99] Koosha Zarei, Reza Farahbakhsh, and Noel Crespi. Typification of impersonated accounts on instagram. In *2019 IEEE 38th International Performance Computing and Communications Conference (IPCCC) (IPCCC 2019)*, London, United Kingdom (Great Britain), October 2019.

[100] Top 1000 Instagram Influencers Ranking. https://hypeauditor.com/top-instagram-sports/?source=imh&source2=imh-ig, 2019.

[101] Face Recognition. Face recognition. `github.com/ageitgey/face_recognition`, January 2020.

[102] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, Jun 2002.

[103] Guillaume Lemaître, Fernando Nogueira, and Christos K. Aridas. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, 18(17):1–5, 2017.

[104] Kai-Cheng Yang, Onur Varol, Pik-Mai Hui, and Filippo Menczer. Scalable and generalizable social bot detection through data selection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(01):1096–1103, Apr 2020.

[105] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018.

[106] The Ultimate Guide to Instagram Hashtags. The ultimate guide to instagram hashtags. `later.com/blog/ultimate-guide-to-using-instagram-hashtags`, January 2020.

[107] Finn Årup Nielsen. A new anew: Evaluation of a word list for sentiment analysis in microblogs, 2011.

[108] E.E. Hutto, C.J. & Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. Eighth International Conference on Weblogs and Social Media (ICWSM-14)., June 2014.

[109] Carson Sievert and Kenneth Shirley. LDAvis: A method for visualizing and interpreting topics. In *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces*, pages 63–70, Baltimore, Maryland, USA, June 2014. Association for Computational Linguistics.

[110] Christopher Manning and Hinrich Schutze. *Foundations of statistical natural language processing*. MIT press, 1999.

[111] Arya Roy. Recent trends in named entity recognition (ner), 2021.

[112] Wafaa S. El-Kassas, Cherif R. Salama, Ahmed A. Rafea, and Hoda K. Mohamed. Automatic text summarization: A comprehensive survey. *Expert Systems with Applications*, 165:113679, 2021.

[113] Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, 34(1):50–70, 2020.

[114] Amila Silva, Ling Luo, Shanika Karunasekera, and Christopher Leckie. Embracing domain differences in fake news: Cross-domain fake news detection using multi-modal data. 2021.

[115] Ahmadreza Mosallanezhad, Mansooreh Karami, Kai Shu, Michelle V Mancenido, and Huan Liu. Domain adaptive fake news detection via reinforcement learning. *arXiv preprint arXiv:2202.08159*, 2022.

[116] Qiong Nan, Juan Cao, Yongchun Zhu, Yanyan Wang, and Jintao Li. Mdfend: Multi-domain fake news detection. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 3343–3347, 2021.

[117] Yichuan Li, Kyumin Lee, Nima Kordzadeh, Brenton Faber, Cameron Fiddes, Elaine Chen, and Kai Shu. Multi-source domain adaptation with weak supervision for early fake news detection. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 668–676. IEEE, 2021.

[118] Alan Ramponi and Barbara Plank. Neural unsupervised domain adaptation in nlp—a survey. *arXiv preprint arXiv:2006.00632*, 2020.

[119] Yaqing Wang, Fenglong Ma, Zhiwei Jin, Ye Yuan, Guangxu Xun, Kishlay Jha, Lu Su, and Jing Gao. Eann: Event adversarial neural networks for multi-modal fake news detection. In *Proceedings of the 24th acm sigkdd international conference on knowledge discovery & data mining*, pages 849–857, 2018.

[120] Zhiwei Jin, Juan Cao, Han Guo, Yongdong Zhang, and Jiebo Luo. Multimodal fusion with recurrent neural networks for rumor detection on microblogs. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 795–816, 2017.

[121] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[122] Qi Liu, Matt J. Kusner, and Phil Blunsom. A survey on contextual embeddings, 2020.

[123] Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare*, 3(1):1–23, Jan 2022.

[124] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, Sep 2019.

[125] Iz Beltagy, Kyle Lo, and Arman Cohan. Scibert: A pretrained language model for scientific text, 2019.

[126] Dogu Araci. Finbert: Financial sentiment analysis with pre-trained language models, 2019.

[127] Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. Clinicalbert: Modeling clinical notes and predicting hospital readmission, 2020.

[128] Edoardo Maria Ponti, Helen O'Horan, Yevgeni Berzak, Ivan Vulić, Roi Reichart, Thierry Poibeau, Ekaterina Shutova, and Anna Korhonen. Modeling Language Variation and Universals: A Survey on Typological Linguistics for Natural Language Processing. *Computational Linguistics*, 45(3):559–601, 09 2019.

[129] Katharina Ehret and Maite Taboada. Characterising online news comments: A multi-dimensional cruise through online registers. *Frontiers in Artificial Intelligence*, 4, 2021.

[130] Mike Schuster and Kaisuke Nakajima. Japanese and korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5149–5152, 2012.

[131] First Draft. Hundreds of users bearing us hashtags or created since june helped promote the british prime minister in activity experts called suspicious.

[132] CNN. Donald trump reveals when he thinks america was great, 2016.

[133] language services direct. How is social media changing the english language?

[134] Yoon Kim, Yacine Jernite, David Sontag, and Alexander M. Rush. Character-aware neural language models, 2015.

[135] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[136] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations, 2018.

[137] Yuxuan Wang, Yutai Hou, Wanxiang Che, and Ting Liu. From static to dynamic word representations: a survey. *International Journal of Machine Learning and Cybernetics*, 11:1611–1630, 2020.

[138] Amirsina Torfi, Rouzbeh A Shirvani, Yaser Keneshloo, Nader Tavaf, and Edward A Fox. Natural language processing advancements by deep learning: A survey. *arXiv preprint arXiv:2003.01200*, 2020.

[139] Daniela Gerz, Ivan Vulić, Edoardo Ponti, Roi Reichart, and Anna-Leena Korhonen. On the relation between linguistic typology and (limitations of) multilingual language modeling. 2020.

[140] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification, 2016.

[141] Hicham El Boukkouri, Olivier Ferret, Thomas Lavergne, Hiroshi Noji, Pierre Zweigenbaum, and Junichi Tsujii. Characterbert: Reconciling elmo and bert for word-level open-vocabulary representations from characters, 2020.

[142] Yoon Kim. Convolutional neural networks for sentence classification, 2014.

[143] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Training very deep networks, 2015.

[144] Wikimedia Downloads. A complete copy of all wikimedia wikis, in the form of wikitext source and metadata embedded in xml.

[145] Huggingface. Cc-news dataset contains news articles from news sites all over the world.

[146] Su Lin Blodgett, Johnny Wei, and Brendan O'Connor. A dataset and classifier for recognizing social media English. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 56–61, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[147] Koosha Zarei, Reza Farahbakhsh, and Noel Crespi. Deep dive on politician impersonating accounts in social media. In *2019 IEEE Symposium on Computers and Communications (ISCC)*, pages 1–6, 2019.

[148] Raad Bin Tareaf. Tweets Dataset - Top 20 most followed users in Twitter social platform, 2017.

[149] Koosha Zarei, Damilola Ibosiola, Reza Farahbakhsh, Zafar Gilani, Kiran Garimella, Noël Crespi, and Gareth Tyson. Characterising and detecting sponsored influencer posts on instagram. In *2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 327–331, 2020.

[150] Koosha Zarei, Reza Farahbakhsh, Noel Crespi, and Gareth Tyson. Dataset of coronavirus content from instagram with an exploratory analysis. *IEEE Access*, 9:157192–157202, 2021.

[151] Emoji for Python. Github: Emoji for python.

[152] Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam, 2018.

[153] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[154] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019.

[155] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics.

[156] Steven Bird and Edward Loper. NLTK: The natural language toolkit. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 214–217, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[157] Dataset on Kaggle. Build a system to identify unreliable news articles.

[158] Parth Patwa, Mohit Bhardwaj, Vineeth Guptha, Gitanjali Kumari, Shivam Sharma, Srinivas PYKL, Amitava Das, Asif Ekbal, Shad Akhtar, and Tanmoy Chakraborty. Overview of constraint 2021 shared tasks: Detecting english covid-19 fake news and hindi hostile posts. In *Proceedings of the First Workshop on Combating Online Hostile Posts in Regional Languages during Emergency Situation (CONSTRAINT)*. Springer, 2021.

[159] opendatascience. How to build a "fake news" classification model.

[160] Statista. Global instagram influencer market size from 2017 to 2020. https://www.statista.com/statistics/748630/global-instagram-influencer-market-value/.

[161] Drucker and Scaccetti. How social media influencers are taxed in the u.s.. https://www.taxwarriors.com/blog/how-social-media-influencers-are-taxed-in-the-u.s.

List of figures

3.1 The architecture of the Instagram crawler. 34
3.2 Follower counts: (a) CDF of followers and followees of all the influencers in the data. (b) CDF of number of followers per account separated in groups. . 39
3.3 CDF of number of comments received per-post: (a) absolute number; (b) normalized. 42
3.4 (a) attracted attention CDF of number of likes received per-influencer; (b) number of comments performed by each user per-influencer 42
3.5 CDF of the number of posts and stories published per influencer. 43
3.6 CDF of number of products promoted by influencers across categories. This only covers stories because equivalent metadata is not available for posts. . 44
3.7 Number of products promoted in stories based on their type (identified via the Instagram API). 45
3.8 (a) The death rate reported by the WHO. (b) The overall trends of the published content. 48
3.9 The Dot Plot of a) follower/followee count of categories, and b) received reactions across categories. 51
3.10 The trends of published posts by categories. 52
3.11 The use of hashtags across categories. 54

4.1 Two samples of impersonators of Donald J. Trump on Instagram. The first snapshot belongs to the genuine account and the others are imposters. . . . 62
4.2 The taxonomy of impersonators. 62
4.3 Identifying Impersonators through profile similarity. 65
4.4 High-Level view of the process of discovering impersonators. 67
4.5 The high-level overview of impersonator-generated content predictor. . . . 69
4.6 The proposed Deep Neural Network architecture to detect impersonator content. 70
4.7 Age of the comments earned by all users vs imposters. 72
4.8 CDF of the average of the comments posted by unique users. 73
4.9 Ternary plot of the ratio of the positive, neutral and negative sentiment post caption across A) communities, and B) Clusters. 74

4.10 Heatmap of topmost hashtags for D. Trump. 75
4.11 Hashtag Heatmap for R. Nadal. 76

5.1 FakeRoBERTaSM: The underlying architecture is based on the RoBERTa [121] and the tokanization part is replaced with Character CNN. 88
5.2 High-level diagram of the word representation with Character CNN [9, 134, 140] used in FakeRoBERTaSM. 89
5.3 Multi-Domain and Social Media Aware Language Model Pretraining. 91
5.4 The proposed deep learning architecture for fake content classification. FakeRoBERTaSM is used as the first feature extractor. To obtain better textual relationships, a Four-Headed 1D-CNNs for the 2nd feature extraction layer and a Single-Headed 1D-CNN for the 3rd feature extraction layer have been utilized. 94

List of tables

3.1 Use Cases and Corresponding Hashtags on Instagram . . . 35
3.2 Summary of Dataset . . . 35
3.3 Influencer Profile Characteristics . . . 39
3.4 Examples of influencers . . . 40
3.5 Tracking Hashtags on Instagram . . . 46
3.6 General Dataset Stats . . . 48
3.7 List of Data Features . . . 49
3.8 List of News Agencies in my dataset . . . 50

4.1 Real Accounts vs. Impersonators . . . 66
4.2 Clustering Feature Set. . . . 67
4.3 Characteristics of the clusters. . . . 67
4.4 Feature Set used in Deep Neural Network. . . . 70
4.5 Performance of the proposed architecture . . . 71
4.6 (A) Politician: Most Discussed Topics in posts. [color code: the real account: green, Fan: blue, Bot: red]. . . . 75

5.1 List of Datasets Used in Pretraining of the FakeRoBERTaSM. . . . 90
5.2 Datasets Used to Benchmark Models. . . . 97
5.3 Performance comparison of the proposed multi-model (Model 6) with other models (baselines and multi-models) across three datasets. All results are compared with the output of Model 1. (+) and (-) indicates whether the model increased the evaluation metric or not. Results in bold indicate the best values among all. . . . 97