الجمهورية الجزائرية الديمقراطية الشعبية
RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي و البحث العلمي
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

Université FERHAT-ABBAS Sétif 1
Faculté des Sciences
Département d'Informatique

جامعة فرحات عباس سطيف 1
كلية العلوم
قسم الإعلام الآلي

THÈSE

Présentée au Département d'Informatique

Pour l'obtention du diplôme de

DOCTORAT

Domaine	**:**	**Mathématiques et Informatique**
Filière	**:**	**Informatique**
Spécialité	**:**	**Data, Text and Web Mining**

Par

FERHAT HAMIDA Zineb

THÈME

FAKE NEWS DETECTION ON SOCIAL MEDIA DOCUMENTS

Soutenue le 23 / 11 / 2023 devant le jury :

M.	**BOUAMAMA Salim**	**Professseur**	**Université Ferhat–Abbas Sétif 1**	**Président**
M^{me}	**DRIF Ahlem**	**MCA**	**Université Ferhat–Abbas Sétif 1**	**Directrice de Thèse**
M.	**GUESSOUM Ahmed**	**Professseur**	**ENSIA, Alger**	**Examinateur**
M^{me}	**KARA-MOHAMED Chafia**	**MCA**	**Université Ferhat–Abbas Sétif 1**	**Examinatrice**
M.	**TOUMI Lyazid**	**MCA**	**Université Ferhat–Abbas Sétif 1**	**Examinateur**
M^{me}	**GIORDANO Silvia**	**Professseur**	**SUPSI, Suisse**	**Examinatrice**
M.	**REFOUFI Allaoua**	**Professseur**	**Université Ferhat–Abbas Sétif 1**	**Invité**

الجمهورية الجزائرية الديمقراطية الشعبية
PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA

وزارة التعليم العالي و البحث العلمي
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

Sétif 1 University – Ferhat ABBAS
Faculty of Sciences
Department of Informatics

جامعة فرحات عباس سطيف 1
كلية العلوم
قسم الإعلام الآلي

FAKE NEWS DETECTION ON SOCIAL MEDIA DOCUMENTS

THESIS

submitted at the Department of Informatics by

FERHAT HAMIDA Zineb

23 November 2023

in partial fulfillment of the requirements for the degree of

DOCTOR

Domain	:	Mathematics and Informatics
Field of Study	:	Informatics
Speciality	:	Data, Text and Web Mining

Examining Board

Mr.	**BOUAMAMA Salim**	**Professsor**	**Sétif 1 University – Ferhat ABBAS**	**President**
Mrs.	**DRIF Ahlem**	**MCA**	**Sétif 1 University – Ferhat ABBAS**	**Supervisor**
Mr.	**GUESSOUM Ahmed**	**Professsor**	**ENSIA, Alger**	**Reviewer**
Mrs.	**KARA-MOHAMED Chafia**	**MCA**	**Sétif 1 University – Ferhat ABBAS**	**Reviewer**
Mr.	**TOUMI Lyazid**	**MCA**	**Sétif 1 University – Ferhat ABBAS**	**Reviewer**
Mrs	**GIORDANO Silvia**	**Professsor**	**SUPSI, Suisse**	**Reviewer**
Mr.	**REFOUFI Allaoua**	**Professsor**	**Sétif 1 University – Ferhat ABBAS**	**Invited**

Abstract:

Due to the rise of social media platforms, a new political, economical and cultural climate arose in which the prevalence of fake news grew significantly. Thus, there are wide implications of false information for both individuals and society. For humans, it is difficult to identify and classify fake news through heuristics, common sense, and analysis. The objective of this Ph.D. research is to suggest automated intelligent approaches for detecting fake news sources, especially social bots. Social bots are autonomous entities that generate significant social media content. In our thesis, we present two main contributions: the first one presents “Sentiment Analysis-based Model for Bot Detection on Social Media” (Deep Bi-LSTM) that incorporates different sentiment and semantic features to perform the bots detection. Experiment on the cresci-2017 dataset shows that our approach can achieve competitive performance with 97.36% of accuracy. The second contribution captures the linguistic-based features by developing a novel framework that we have called “Hybrid Mixing Engineered Linguistic framework Features Based on Autoencoder”. This framework is split into two segments: the features learner and a deep neural networks classifier. The feature learner aims at performing the feature extraction task due to a deep autoencoder based on dense layers and a BiLSTM autoencoder. We enhance the feature extractor: (i) by feeding the lexical and syntactic features to the first autoencoder to represent the high-order features in latent space; (ii) by building the semantic and the context features using the BiLSTM autoencoder; (iii) the merging of the two previous trained encoder blocks would generate a compacted data based on elite features. This architecture help us to discover human writing style patterns accurately. Experiments conducted on real datasets show that a significant improvement can be achieved for fine-grained bots detection with 92.22% of accuracy.

Keywords: fake news ; social networks ; social bot detection ; natural language processing ; feature engineering ; classification.

الملخص:

بسبب ظهور منصات التواصل الاجتماعي، ظهر مناخ سياسي واقتصادي وثقافي جديد زاد فيه انتشار الأخبار المزيفة بشكل كبير وبالتالي، فإن الأخبار المزيفة لها آثار واسعة على كل من الأفراد والمجتمع. بالنسبة للبشر، من الصعب تحديد الأخبار المزيفة وتصنيفها من خلال الاستدلال والفطرة السليمة والتحليل. الهدف من أطروحة الدكتوراه هذه هو اقتراح مناهج آلية ذكية لاكتشاف مصادر المعلومات الخاطئة، ولا سيما الروبوتات الاجتماعية. الروبوتات الاجتماعية هي كيانات مستقلة تنشئ محتوى مهمًا على الشبكات الاجتماعية. في أطروحتنا، نقدم مساهمتين رئيسيتين: النهج الأول هو نموذج "Sentiment Analysis-based Model for Bot Detection on Social Media" (Deep Bi-LSTM). تُظهر التجارب على مجموعة بيانات cresci-2017 أن نهجنا يمكنه تحقيق أداء تنافسي بدقة تصل إلى 97.36٪. المساهمة الثانية تلتقط الميزات القائمة على اللغة من خلال تطوير إطار عمل جديد يسمى "Hybrid Mixing Engineered Linguistic Features Framework Based on Autoencoder" يتكون هذا الإطار من عنصرين: متعلم الميزات ومصنف الشبكة العصبية العميقة. يتكون متعلم الميزات من برنامج تشفير تلقائي عميق يعتمد على طبقات كثيفة وجهاز تشفير تلقائي ثانٍ يرتكز على نموذج BiLSTM. هذا جعل من الممكن استخراج الميزات ذات الصلة بواسطة التشفير أثناء تطبيق نقل التعلم. تميز هذه الهندسة بشكل صحيح الاختلافات في أسلوب الكتابة بين البشر والروبوتات. بعد ذلك، أدى تهيئة المصنفات بالوظائف المنقولة إلى تحسين أداء اكتشاف الروبوتات بدقة بلغت 92.22٪.

الكلمات المفتاحية: الأخبار المزيفة ؛ وسائل التواصل الاجتماعي ؛ كشف الروبوتات الاجتماعية ؛ معالجة اللغة الطبيعية ؛ هندسة الميزات ؛ التصنيف.

Résumé :

En raison de la montée en puissance des plateformes de médias sociaux, un nouveau climat politique, économique et culturel est apparu dans lequel la prévalence des fausses nouvelles a considérablement augmenté. Ainsi, les fausses informations ont de vastes implications tant pour les individus que pour la société. Pour les humains, il est difficile d'identifier et de classer les fausses nouvelles par l'heuristique, le bon sens et l'analyse. L'objectif de cette thèse de doctorat est de proposer des approches intelligentes automatisées pour détecter les fausses sources d'informations, en particulier les robots sociaux. Les robots sociaux sont des entités autonomes qui génèrent un contenu important sur les réseaux sociaux. Dans notre thèse, nous présentons deux contributions principales : la première approche est un modèle " Sentiment Analysis-based Model for Bot Detection on Social Media " (Deep Bi-LSTM) qui intègre différentes fonctionnalités de sentiment et de la sémantique pour effectuer la détection des bots. Les expérimentations sur le jeu de données cresci-2017 montrent que notre approche peut atteindre des performances compétitives avec une précision de 97.36%. La deuxième contribution capture les fonctionnalités basées sur la linguistique en développant une nouvelle framework appelée "Hybrid Mixing Engineered Linguistic Features Framework Based on Autoencoder". Cette framework comporte deux composants : l'apprenant de caractéristiques et un classificateur de réseaux de neurones profonds. L'apprenant de fonctionnalités est constitué d'un autoencodeur profond basé sur des couches denses et un deuxième autoencodeur BiLSTM. Ce qui a permis d'extraire des caractéristiques pertinentes par les encodeurs tout en appliquant l'apprentissage par transfert. Cette architecture discerne correctement les différences dans le style d'écriture des humains et des bots. Ensuite, les initialisation des classificateurs avec les fonctionnalités transférées a nettement amélioré les performances de la détection de bots avec un précision de 92.22%.

Mots-clés : fausses nouvelles ; réseaux sociaux ; détection de bots sociaux ; traitement du langage naturel ; ingénierie des fonctionnalités ; classification.

Dedications

*****

I dedicate this work to my lovely family especially
to my father and mother,
to my brothers Oussama and Mohamed El Hacen,
to my sister Kaouther,
and to my grand mother.

Acknowledgements

*****

I am one of those people who firmly believe that there is no strength or power except from ALLAH. My thanks go first of all to Almighty ALLAH for the will, the health, and the patience that has given me during all these years of study.

I would like to express my sincere thanks to my supervisors, Professor REFOUFI ALLAOUA and Doctor DRIF AHLEM, who have honoured me by accepting to supervise my work, for the interest and the precious advices they provide me throughout this work, and for their assistance in writing this thesis. During my thesis, they guided me by offering many ideas and perspectives. Thanks to their logical thinking and their experience in the NLP field which is one of the most complicated field in artificial intelligence domain, my initial ideas and proposals have been improved and developed. Also, I would like to express my gratitude for the support and the contribution provided by Professor Silvia Giordano and Doctor Luca Luceri, which have greatly enriched this work.

I wish to express my thanks to the distinguished President of the jury, Mr. Salim BOUAMAMA Professor at Setif 1 University – Ferhat Abbas.

My sincere thanks also to the member of the jury: Mr. Ahmed GUESSOUM Professor at the National School of Artificial Intelligence in Algeria, Mrs. Chafia KARA-MOHAMED Doctor at Setif 1 University – Ferhat Abbas, Mr. Lyazid TOUMI Doctor at Setif 1 University – Ferhat Abbas and Mrs. Silvia GIORDANO Professor at the University of Applied Sciences and Arts of Southern Switzerland for having accepted to be responsible for evaluating this project.

Finally, I would like to express my deepest gratitude to my family, my colleagues and my friends for providing me with ongoing support and encouragement throughout my years of study and throughout the process of researching and writing this thesis. My warmest thanks also go to all those who have helped, by any means, to carry out this work. This accomplishment would not have been possible without them.

Contents

*****

List of Figures

*****

List of Tables

*****

Chapter 1

Introduction

*****

The Internet has expanded and profoundly altered the world over the recent decades. It steadily displaced conventional media, and its applications have expanded to include expressing news and opinions, facilitating social contact, providing many online services (shopping, banking, freelancing) and many others. As social media has grown through the internet, our personal zones have transformed, as well as how we engage and trade viewpoints online. These changes also affect our different decisions. Furthermore, it offers us fresh ways to create large-scale political trends. This encourages some influencers and evil groups to exploit social media for covert ends like the proliferation of fake news. Therefore, False information can be shared for a variety of reasons and may be widely disseminated with potentially devastating consequences for individuals and society, such as:

Politically, Russian forces invaded Ukraine on February 24, 2022, and immediately began actively disseminating false information about the conflict in an effort to erode Ukraine public support [1]. In times of crisis, when having access to reliable information is crucial, these internet threats are particularly relevant. There are numerous recorded examples that demonstrate Russia's interference in other countries' political systems. An example of this is the current U.S. Congress inquiry into Russian meddling in the 2016 presidential election, which accuses Russia of using trolls (fraudulent accounts made with the intention of manipulating) and bots (automated accounts) to disseminate false information and politically motivated information [2, 3].

Healthily, Several false and deceptive statements about Covid-19 vaccines have been making the rounds online amid the Covid-19 issue, despite the lack of any supporting data. New vaccines, such ”Moderna” and ”BioNTech/Pfizer,” are being blamed by so-called experts on social media for having a harmful impact on the fertility of women who have received them, despite the denial of many medical professionals. Such claims are deceptive and intended to arouse unwarranted alarm. On June 5, the Centers for Disease Control (CDC) published a study on 502 adults in the United States that revealed 39% of responders had engaged in risky behaviors, such as washing food products with bleach, putting household cleaners directly on skin, and purposefully inhaling or ingesting disinfectants to avoid contracting COVID-19 [4]. In fact, false information is a very harmful force that exacerbates a delicate emotional situation.

Socially, Growing mistrust in society is a result of stories that are false and purposefully deceive readers. This mistrust occasionally manifests as rudeness, irrational outrage, or even physical violence. We can think of the devastating flooding that occurred in western Germany in the middle of July 2021 and was accompanied by a deluge of false information. It was reported that 600 newborns and children's bodies washed ashore in the flood zone. Evidence included a segment from a German news station's broadcast report. A local in the disaster area reported that they discovered children's bodies carried away by water inside their homes, and the station's correspondent briefly discussed their traumatic experiences. The video made no mention of the 600 children and newborn bodies that were found. Despite the fact that some youngsters may have been victims of such floods, no dead corpses have been reported, according to the police. Even though the erroneous information had long ago been debunked by fact-checking reports, many stunned people still spread it.

False information (fake news) spreading on social media is a plague of contemporary civilizations because it undermines the fundamental components of a just society, namely the truth. As a result, it's critical to understand how social media works, how fake news is created, how it spreads through social networking sites, and where it comes from. This doctoral dissertation specifically looks at false automated ”social bots” accounts that may post information and communicate with other accounts as if they were run by actual people. And we propose an improvement of social bots detection methods in order to spot fake news on social media and counter its spread. This improvement is achieved through the proposal of two Bots detection approaches based on deep learning models and Automatic Language Processing (TAL) techniques. The first approach study the sentiment characteristics that contribute to an accurate detection of bots. In fact, the aggregation of these characteristics, with textual features produces large, incomplete, unstructured and noisy data, which need an architecture that has multiple inputs concatenated before being passed to a Bidirectional Long-short-term memory (Bi-LSTM network). In the second contribution, we propose a new deep approach based on the linguistic characteristics and an extensive knowledge of the behavior of the writing style of a “social bot”. The proposed architecture generates elite elements from the pre-trained coder part from the latent spaces with transfer learning. Experiments on real dataset reveals that the writing style is a key element for an accurate detection of sources (bots) of false information.

Main issues and research objectives

- Giving a comprehensive survey spanning diverse aspects of both fake news and social bots and their detection method.

- Developing a Bidirectional Long Short Term Memory network model based on set of features which include sentiment polarity and subjectivity, the number of the happy emoticons and the sad emoticons, the number of the interjections (i.e. ah!, ooh!..) in the tweets. Besides, we extract the semantic features using embedding vectors, which are well suited for modeling tasks related to sequences and sentences.

- In contrast to most previous research on bot detection, which has examined without accounting for the unique type-token ratio and vocabulary knowledge, a more extensive collection of features, this study examines the writing style of the bot to determine whether it is possible to achieve by employing only a few pertinent linguistic features, a competitive detection performance.

- We intend to delve deeper to demonstrate that the linguistic aspects can contribute considerable value to distinguish human accounts from bot accounts because a successful bot can employ a linguistic approach based on the linguistic structure. The following analysis of writing style features will be covered in our exploratory study: lexical features based on text richness and diversity measures, syntactic features based on Post-tagging, and word embedding methodologies are used to extract semantic features.

- Several deep learning and machine learning techniques have been used to implement bot identification, but there is still much to be done. In fact, it can be difficult to employ only the bots' automatic writing style characteristics because their combination results in noisy, fragmentary, and unstructured data. In order to enhance the detection performance, we will create a hybrid deep learning strategy based solely on linguistic features.

Thesis outline

The research work consists of several chapters, namely:

Chapter 2: In this chapter, we present an introduction to Automatic Language Processing (NLP) techniques, deep learning models for NLP, their characteristics, and their architectures.

Chapter 3: This chapter presents the literature review that we have carried out. First, we present the basic concepts of the field of social documents and false information, and their different sources. Secondly, we detailed the most relevant works in the literature dealing with the verification and detection of false information on social media, bringing out a comparison of these approaches in terms of the features levels. In a third step, we present the different approaches for detecting "social bots". This chapter constitutes in fact a background of our contributions of chapters 3 and 4. Our academic paper related to this literature review is:

1. Ferhat Hamida Zineb, Refoufi Allaoua, and Drif Ahlem. (2022). "Fake News Detection Methods: A Survey and New Perspectives." In: Kacprzyk, J., Balas, V. E.,

Ezziyyani, M. (eds) Advanced Intelligent Systems for Sustainable Development (AI2SD'2020). AI2SD 2020. Advances in Intelligent Systems and Computing, vol 1418. Springer, Cham. https://doi.org/10.1007/978-3-030-90639-9_11 [5].

Chapter 4: This chapter presents our first contribution which consists of the proposal of a detection approach of "social bots" named "Sentiment Analysis-based Model for Bot Detection on Social Media". This deep model is designed to answer the following research questions: - do human and bots texts allude to the same feelings? - is there a difference between the spreading positive and negative emotions of humans and bots? - could bot detection be improved by extracting each type of sentiment? In order to reply to these research questions, first, we apply a feature extraction technique to identify text polarity, subjectivity, interjections, and emoticon types, with a focus on understanding the features importance for this task. The words are then accurately transformed into a vector space using the lexical embedding approach "word embedding," where comparable words are represented by similar vectors. The final phase consists of exploiting both the semantic functionalities and the sentiment functionalities through the implementation of an architecture with several concatenated entries before being transmitted to a network of Bidirectional short and long-term memory recurrent neurons (bi-LSTM network). This combination of features has been shown to slightly improve bot detection performance. The experimental results show a high accuracy of the bidirectional recurrent neural network ability that models efficiently the merging of content and sentimental traits. Finally, this first contribution has attempted to answer the research questions, leading to the emergence of a research problem that will be the subject of our main contribution. The academic publication of this work is:

2. Ferhat Hamida, Z., Refoufi, A., Drif, A., and Giordano, S., "Sentiment Analysis-Based Model for Bot Detection on Social Media." 1st National Conference on Applied Science and Advanced Materials (NCASAM-2021) December 20-22, 2021 – ENSET–Skikda [6].

Chapter 5: This chapter addresses relevant research questions, such as: - what is the difference between the bot's writing style and the human writing style? - Is there a possibility by employing only a few pertinent linguistic features, achieve a competitive detection performance? Unlike most bot detection work that has integrated a larger feature set without considering that language is one of the most complex human faculties, we explored linguistic features for bot detection. On the one hand, we study the importance of lexical measures and syntactic indicators for the bots detection, on the other hand, we develop a new hybrid architecture for linguistic functionalities based on auto-encoders (Hybrid-MELAu). This semi-supervised framework is composed of two essential elements: learner functionalities and predictors. Learning functionality is provided by two powerful frameworks: a) the first is a dense Deep Autoencoder powered by lexical and syntactic content (DALS) which represents the higher order of lexical and syntactic features in latent space, b) A Glove-BiLSTM autoencoder, which is the second, creates semantic characteristics. Then, the fusion of the two previous structures uses transfer learning to produce elite elements from each latent space's trained encoder portion. Our proposal has been compared to related works to show the importance of discerning the differences between the writing styles of humans and bots by implementing an effective deep linguistic architecture. Our contribution has been published in: **3- Ferhat Hamida Z., Refoufi, A., Drif, A., & Giordano, S. (2022). Hybrid-MELAu: A Hybrid Mixing Engineered Linguistic Features Framework Based on Autoencoder for Social Bot Detection. Informatica, 46(6),** `https://doi.org/10.31449/inf.v46i6.4081` [7].

Finally, we conclude the thesis and introduce possible future directions.

Chapter 2

Deep learning for natural language processing (NLP): Fundamental theories and basics.

*****

2.1 Introduction

One of the most well-known and quickly developing areas of computer science research is artificial intelligence (AI). Through learning, reasoning, and self-correction, AI seeks to integrate human intelligence in machines, especially computer systems. As a result, it has expanded to encompass a variety of fields, from robotics to machine learning (ML) and deep learning (DL), where we may use this latter technique to address many different problems, including those relating to natural language processing (NLP) and text analytics.

In this chapter, we give a brief theoretical background on NLP, its linguistic tools and the powerful relationship between this field and the deep learning techniques especially the word representation methods.

2.2 Natural Language Processing

2.2.1 What is NLP ?

2.2.1.1 Definition

Language understanding is innate in humans, but computers have long struggled to do the same. As a result, one of the most significant and challenging areas of artificial intelligence has emerged: **Natural Language Processing (NLP)**. NLP is defined as a computer system's ability to analyze or synthesize spoken or written language using software or hardware. Computer analysis of speech and text is an exciting project, but it is not without difficulties. **"Natural Language Understanding" (NLU)** consist of making a computer system actually comprehend natural language like a human being (NLU) [8].

2.2.1.2 A Brief History of Natural Language Processing

Research on NLP started in 1940. Typically, its history can be divided into two major periods: appearance before and during deep learning [9] (Figure 2.1 and Figure 2.2).

2.2.1.2.1 NLP before the Deep Learning: Following the Second World War, Weaver's note [10] presented the idea of the first computer-based natural language application, machine translation (MT) in order to develop a system that could translate between languages automatically. It wasn't until 1957 that generative grammar, a rule-based system of syntactic structures, was initially presented to enhance machine translation [11].

Between 1957 and 1970, researchers divided into two distinct groups to study NLP: symbolic and stochastic. Many linguists and computer scientists focused on formal languages and developing syntax in symbolic, or rule-based. While, the statistical and probabilistic applications of NLP focus on optical character identification and pattern recognition between texts. Another NLP concepts also arose in the 1970s, including the development of conceptual ontologies that arranged real-world information into data [12–15].

Between 1983 and 1993, researchers tended to converge more on empiricism and probabilistic models as they discovered that many of Chomsky's arguments were written well but were not supported by empirical data [16–24]. Therefore, by 1993, statistical and probabilistic models were the most often used for natural language processing [25, 26].

2.2.1.2.2 NLP after the Deep Learning: Bengio *et al.* in 2003 [27] developed the first neural language model, which consisted of a one-hidden-layer feed-forward neural network [27]. Collobert and Weston [28] in 2008, added multi-task learning to neural networks for NLP, handling multiple learning tasks concurrently. They utilized a single convolutional

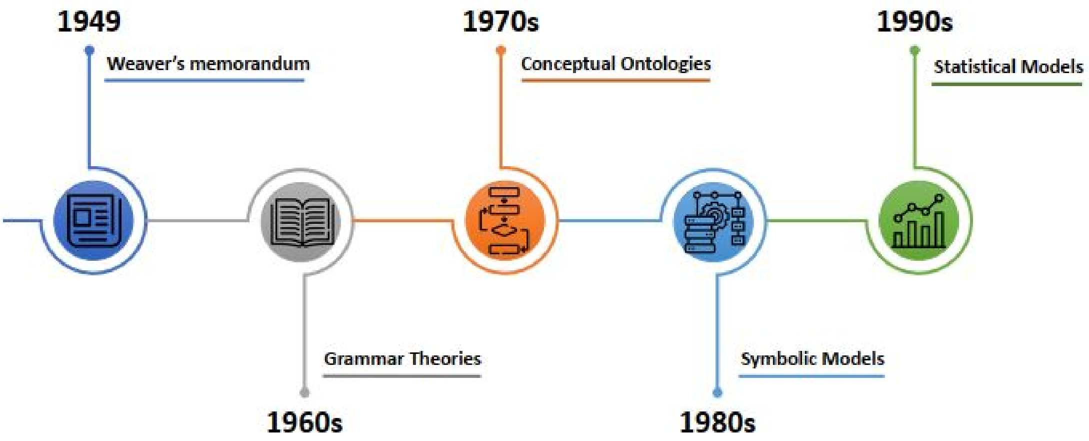


Figure 2.1: The big stages of NLP before the deep learning era [9].

neural network design [29] that generate variety of language processing predictions (including a part-of-speech tags, and semantic roles).

Bengio *et al.* [27] used dense vector of word representations and removed the hidden layer to approximate the objective. Mikolov *et al.* [30,31] presented an effective improvement to the training process in 2013. These simple improvements allowed for extended word embedding training on large unstructured text corpora, which is called Word2Vec. Additionally, three clearly defined types of neural networks were used for NLP (the recurrent neural networks [32], convolutional neural networks (CNNs), and recursive neural networks [33]). Sutskever *et al.* [34] suggested sequence-to-sequence learning, a comprehensive end-to-end method for mapping one sequence to another using a neural network, in 2014. They use an encoder neural network to parse a text before compressing it into a vector . Then, a decoder neural network predicts the output sequence.

In 2015, the introduction of the principle of attention by Bahdanau *et al.* [35] was one of the main developments in Neural Machine Translation (NMT) and the core idea that allowed NMT models to outperform conventional sentence-based MT systems. The most recent large pre-trained language models [36] significantly outperform state-of-the-art techniques on a range of tasks. Pre-trained language model embeddings can be used as features in a target model, or a pre-trained language model [37] can be fine-tuned using target task data, to enable efficient learning with substantially less data [38–41].

2.2.2 NLP and linguistics

This section introduce terminology and notions that are used often in the NLP field. From the figure 2.3, generally, NLP has 4 phases in its language learning process.

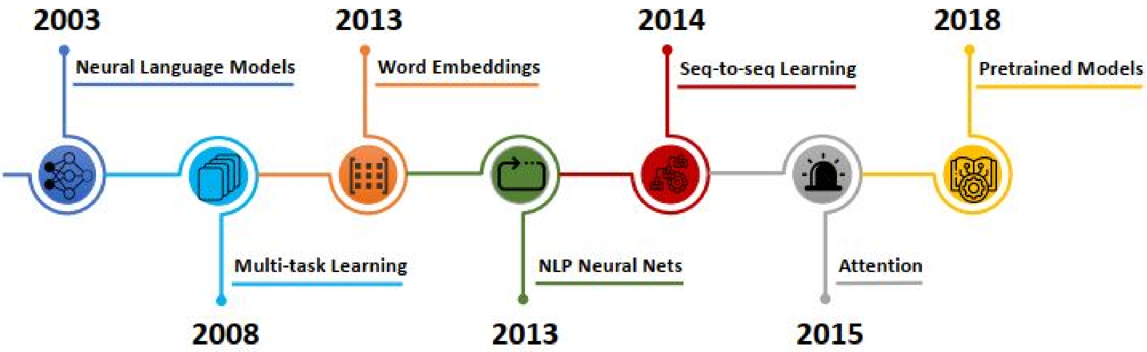


FIGURE 2.2: The big stages of NLP in the deep learning era [9].

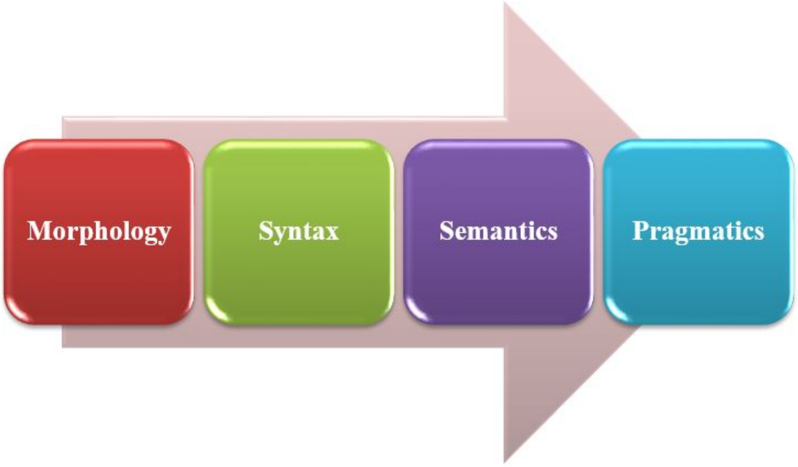


FIGURE 2.3: Natural Language Processing Process.

2.2.2.1 Morphological Level

In traditional linguistics, morphology investigates the origins of words, their history, and how their shape changes depending on the situation. A word is viewed as a series of characters at the word level, where most operations happen [8].

2.2.2.2 Syntax and semantics

The proper word relationships in a sentence is a syntax-related issue. The study of syntax focuses on how sentences are put together and how to identify correct sentences. Syntax is similar to what we usually refer to as grammar. On the other hand, semantics makes use of all of the aforementioned to assess sentence structure and comprehend the significance of words in texts so that computers may grasp language like humans do [8].

2.2.2.3 Pragmatics and context

In order to understand various context, words and sentences are forcefully grasped in the context of other information or prior knowledge that may not be present at the moment. This gives the AI the capacity to continuously connect recent and historical information, similarly to what a human being do [8].

2.2.2.4 Two views of NLP

NLP is often referred as ”symbolic” because it is largely concerned with manipulating symbols, for instance, grammatical norms that assess if a phrase is well-formed. Due to its heavy reliance on symbolic computation, traditional artificial intelligence currently outperforms all preceding approaches by a considerable margin.

The second methods are the statistical analysis of language known as ”empirical” since it gathers language data from sizable text corpora like those on the internet and in news feeds. Symbolic NLP frequently works top-down, imposing preconceived grammatical structures and meaning connections on texts. In most cases, empirical NLP begins with the texts themselves and works top-down, looking for patterns and associations.

The methods that uses only symbols must remove uncertainty by proposing additional context-specific factors. This methodology is knowledge based since it depends on human experts to identify regularities in the subject. The empirical approach is more quantitative since it frequently links probabilities to various evaluations of textual data using statistical techniques [8].

2.2.2.5 Tasks and supertasks

The principal application of language processing in the Web nowadays is document retrieval. The 1990’s saw a trend toward more sophistication in the indexing, identification, and presentation of significant texts. A related operation called document routing involves automatically forwarding things in a document stream to a user, for instance, one who fits a specific profile.

Document routing involves the task of document classification. This task involves categorizing documents into classes, usually according to their content. The document indexing task consist on assigning automatically particular words or phrases to a document, for instance, to generate an index that looks like the back of a book. Furthermore, the process of extracting the relevant information from a document and display it as a substitute document is the information extraction.

Super-tasks is the combination of the previous tasks in novel ways. By combining these tasks in novel ways, super-tasks can be produced. As an illustration, software could classify documents from a stream based on their content, choose the documents from the stream and then extract some useful information from each document. These super-tasks are currently being investigated under ”text mining” [8].

2.2.3 Linguistic tools

Textual linguistic analysis ofen proceeds in layers. Sentences, paragraphs, and individual words are the three divisions used in documents. A sentence’s words are then classified by part of speech and other characteristics that are subject to grammatical analysis before being processed. As a result, the basic building blocks of parsers are sentence delimiters, tokenizers, stemmers, and part of speech (POS) taggers [8].

2.2.3.1 Sentence delimiters and tokenizers

It is challenging to accurately identify sentence boundaries since punctuation marks that indicate a sentence’s end might occasionally be ambiguous. Instead, regular expressions, exception rules, or other information like part-of-speech frequencies are needed. Tokenizers (lexical analyzers) divide a stream of characters into recognizable tokens such as words, numbers, identifiers, or punctuation [8].

2.2.3.2 Stemmers and taggers

In reality, stemmers are morphological analyzers that connect several ways of writing the same word to a root form. The root is regarded as the shape that would generally appear as an entry in a dictionary. For instance, the terms ”go,” ”goes,” ”going,” ”gone,” and ”went” will all share the root form ”go.”

The foundation of part of speech taggers, which assign the appropriate tag to each word in a phrase, are tokenizers and sentence delimiters. We identify whether a word is a noun, verb, adjective, etc [8].

2.2.3.3 Stop word removal

In NLP, stop word reduction is a key tactic for minimizing the massive raw input space (swr). Like auxiliary verbs or articles, certain words are used more frequently than others or don’t convey a lot of information about the substance of texts in the majority of languages. Because of this, it is frequently correct to ovoid further analysis of the given stop words [8].

2.2.3.4 Sentiment analysis

Sentiment analysis is one of the most active research areas in natural language processing. In sentiment analysis, the three granularity levels of document, phrase, and aspect have attracted the most study interest. By assigning an overall sentiment orientation/polarity, the first level concern is to determine whether an opinion document (like a comprehensive online review) reflects an overall positive or negative impression. The sentiment of a sentence can be ascertained using the subjectivity classification and polarity classification.

Sentiment categorization of sentences in present deep learning models is often structured to predict whether a statement is positive, neutral, or negative. Usually, aspect-level sentiment classification consider both the sentiment and the target information. Inferring the sentiment polarity/orientation of the sentence towards the target aspect from a phrase and a target aspect is the goal of aspect-level sentiment classification [42].

2.2.3.5 Lexical diversity measures

Lexical diversity is a measure of how many different words appear in a text and it can be calculated in several different ways. We might consider the total number of words in the text, or consider only the number of clauses in each sentence, or we might focus on the lexical words.

2.3 Deep Learning for NLP

Over the past ten years, deep learning has been the current trend in AI. Results have continuously redefined the state of the art for a variety of data analysis operations across a number of fields. Deep learning-based NLP presently outperforms all prior techniques by a significant margin [43].

2.3.1 Machine learning methods for NLP

Machine learning is now massively used in natural language processing. In this subsection, we begin with a brief description of certain machine learning models [43].

2.3.1.1 The perceptron

The learning method of a perceptron (Rosenblatt's perceptron) was based on a simple one-layer neural network, which successfully served as the prototype for all succeeding neural networks (see figure 2.4). Based on a threshold θ and a bias b, the single-layer perceptron is capable to produce a binary output y (0 or 1) from a combination of input values that are weighted $x_1 x_n$,:

$$f(x) = \begin{cases} 1 \text{ if w.x+b} > 0 \\ 0 \text{ else} \end{cases} \tag{2.1}$$

The training set of labeled data, consists of input vectors with output labels. It is used to compute the weights $w_1, ...w_n$. A neuron is the unit that has reached the threshold. It is given the weighted and summed input v.

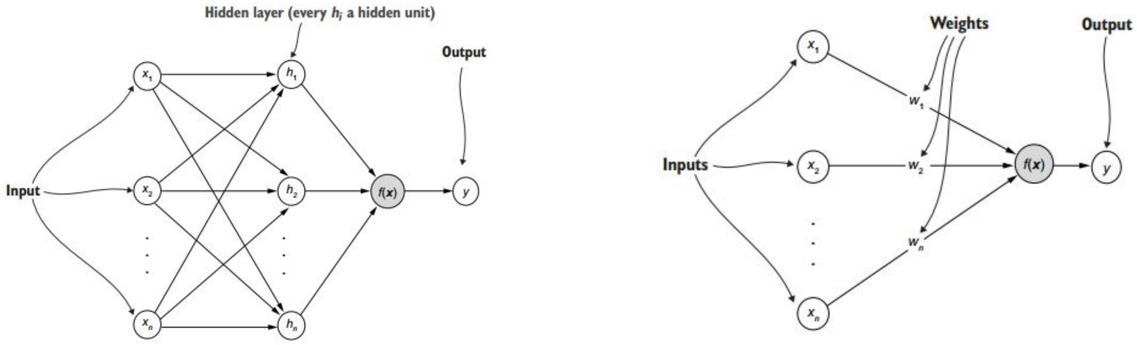


Figure 2.4: The right part represents Rosenblatt's perceptron that composed with a single neuron receiving several inputs and generating (by applying a threshold) a single output value. The left part is a multilayer perceptron (MLP) with an input layer, one hidden layer ($h_1 \cdots h_n$), and an output layer [43].

This imperfect network may train a specific set of functions that deal with the class of linearly separable problems. The single-layer perceptron is no longer widely utilized in NLP since, in reality, linear algorithms have difficulty telling apart highly entangled material with a common lexicon. The original perceptron's single-layer model is transformed into a model with at least three layers by the multilayer perceptron (MLP), which include an input layer, one or more hidden representational layers, and an output layer [43].

2.3.1.2 Support vector machines

Input from feature space is automatically translated to higher dimensions where it can be divided by a hyperplane, or a straight plane, by a binary classifier known as an SVM. This implicit mapping is performed with the aid of a kernel function.

This function transforms the original input space to an alternate representation with implicitly higher dimensionality in order to untangle the data and make it linearly separable. The distance between two feature vectors is only calculated when a similarity function is applied to them, hence this transformation is implicit (called The kernel). In general, a kernel function takes two vectors, adds a constant (a kernel parameter), and then adds extra kernel-specific elements to produce a specialized form of the dot product of two vectors. Two classes are, at most, separated with borders that are as wide as possible (called maximum margins. figure 2.5) in the modified space produced by the kernel trick.

Support vectors are the data points that compute the slope of these boundaries. An SVM's task for training is to determine weights that minimize the error margins. The model is composed of the support vectors, different weights, and biases after training. A positive or negative label is assigned to fresh input based on which side of the support vectors it lands on (recall that SVMs are binary classifiers). As a result, SVMs discard the majority of their training data and only retain the support vectors.

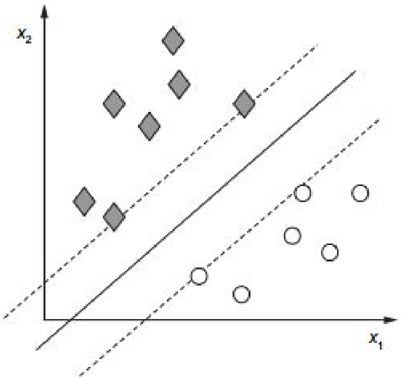


Figure 2.5: Maximum margins of an SVM. The support vectors are the points on the dashed lines [43].

2.3.1.3 Memory-based learning

MBL or what is called Memory-based learning is a kind of lazy machine learning, in contrast to eager varieties of machine learning that construct condensed and representative models of their training data. It maintains all training data available in memory rather than compressing it into generalizations. The real processing of the training data occurs during classification: input data and training data are matched using similarity or distance metrics. Distance functions between vectors compute similarity in a manner akin to SVMs. However, we don't use any dimensionality trickery in this case because we are working with explicit vectors.

Based on feature value overlap, this measure calculates the separation between two feature vectors: 100% similarity for non-numerical (symbolic) values. Most of these algorithms add feature weighting (such as information-gain-based weighting) or exemplar weighting to these distance measurements. They divide the matching search space into groups of training items that are equally spaced from the present test item. For instance, the procedure can find sets with distances $d_1, d_2, ...$ before computing the most prevalent class in those sets. The most likely label for the test item is then determined by casting votes across all classes. The k parameter controls the amount of distance sets that must be considered, this is why MBL is frequently classified using k-nearest distances [43].

2.3.2 Vector representations of language

Vectors are frequently used on machine learning approaches, which are fixed-size collections of numerical values and correlate to points in multidimensional spaces. Calculating the distance between points in these spaces is the core task of machine learning. which are extremely high-dimensional for typical machine learning applications like text mining and, as a result, defy our human sense regarding geometry. Several methods can be used to transform texts into vectors. There are essentially two categories of linguistic vector representations [43]:

2.3.2.1 Representational vectors

These vectors can be precisely and directly computed from the data. The simplest form of such a vector for text is to represent words with characters [43].

2.3.2.1.1 Bag-of-words: Each dimension in a bag-of-words representation can be thought of as representing a distinct feature dimension, which is the presence of a certain word in an index lexicon, where 0 and 1 respectively represent the absence or presence of the ith word in the phrase at a specific location i [43].

2.3.2.1.2 One-hot vectors: Words are represented as a sparsely filled N-dimensional vector in one-hot vectors, where a lexicon has N size. Only one of the dimensions—the dimension associated with a particular word—has a value of 1. For example, each word in a 50,000 word text is displayed by a cumbersome 50,000-dimensional vector with the single digit ”on” [43].

2.3.2.2 Operational vectors

These types of vectors, in contrast to representational vectors, are estimated from data using statistics or machine learning. Operational vector representations show a derived representation of the data that an algorithm has generated. Since they are created by irreversible algorithms that yield numerical vectors, these vectors are typically not interpretable by humans [43].

2.3.2.2.1 TF.IDF: With this approach, words are given numerical weights based on a term frequency and inverse document frequency product. Less salient words, such as stopwords, are those with lower weights. These representations facilitate the focus of significant words by machine learning algorithms. Almost all machine learning algorithms compare two vectors, and highlighting some dimensions while underplaying others can help produce estimations of similarity that are more precise. A word’s frequency in the document to be conveyed is expressed by the term frequency quantity [43]:

$$tf(w|d) = |w \in d| \tag{2.2}$$

The inverse document frequency in a collection of documents D indicates how often a word appears in other documents d:

$$idf(w|d, D) = \log \frac{|D|}{|d \in D : w \in d|} \tag{2.3}$$

A number called TF.IDF, which combines these two numbers, strikes a balance between a word's frequency and the number of documents it appears in:

$$tf.idf(w|d,D) = tf(w|d) \times idf(w|d,D) \tag{2.4}$$

The log ratio will approach 1 when d appears in every document in the collection D since log(1) = 0, the latter value will approach zero whenever it is high (indicating that a specific word is common). The contribution of the word's frequency to the TF.IDF weight is thereby effectively suppressed by the idf factor.

2.3.2.2.2 Neural word embedding: Undoubtedly, one of the most significant developments in text mining over the past ten years has been neural word embeddings. These embeddings create operational vector representations of words and are also referred to as Word2Vec embeddings (for more details see section 2.4.3).

2.3.3 Vector sanitization

Vectors can be autoclaved or optimized utilizing a variety of postprocessing techniques, whether they be representational or operational. The next two are normalization and dimensionality reduction by hashing.

2.3.3.1 The hashing trick

In order to reduce the wide feature vectors dimensionality, a hashing function is applied to the features in those vectors, where every feature is mapped by this function to an index, and only those indices are updated by the algorithm assuming there is an inverted lexicon that associates a word with an integer (rather than mapping words to indices). The indices with positive values in the input vector with binary values, which is a hash value given a hashing function, are recovered by this inverse lexicon. The value is limited to the range of 0 to the size of the output vector. The indexing is carried out by the hashing function as follows: similar input values will cause similar number indices to be increased. The particular hash function selected controls the degree of similarity [43].

2.3.3.2 Vector normalization

Vectors with numerical data can be normalized, including the TF.IDF vectors we encountered. As a result, the variance among their dimensions is reduced since they are compressed into a subspace. The magnitude of each vector has been normalized to be 1. By dividing each

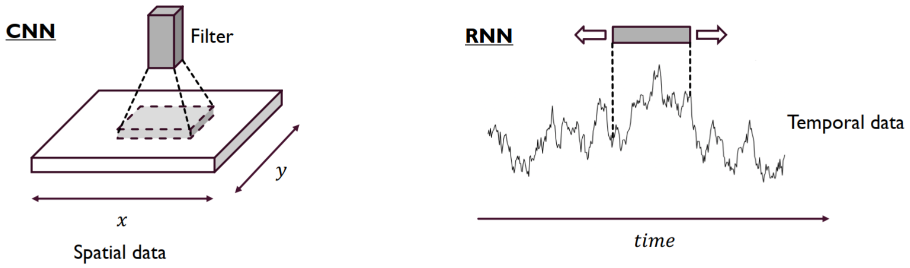


Figure 2.6: Spatial and temporal operators [43].

vector's component by its magnitude. Vectors have all been normalized in this situation:

$$\hat{v} = \frac{v}{||v||}; ||v|| = \sqrt{\sum_{i}^{n} v_i^2} \tag{2.5}$$

Any such normalized vector is referred to as a unit vector. This is the normalized form of a vector v. Any machine learning method that is sensitive to outlier data will benefit from the normalization, which compels vectors to lie within the same data range [43].

2.3.4 Deep learning

One of the most popular AI techniques of the last five years is deep learning. It is a neural network that features numerous internal or hidden layers as well as particular filtering techniques. Here, we examine the fundamental structures of deep learning: multilayer perceptrons and various input-filtering methods, including spatial and temporal filters.

2.3.4.1 Deep multilayer perceptrons

A multilayer perceptron is the archetypal deep learning network (MLP) (see figure 2.4). Artificial neurons, which are essentially mathematical functions that accept input through weighted connections to other neurons, are layers that make up MLPs. They use a range of mathematical processes to obtain output values. Deep neural networks can manipulate a large number of weights and numerous neurons. Typically, a deep network has a lot of hidden levels (more than two) between their input and output layer [43].

Two basic operators: Spatial and temporal

The interaction between spatial and temporal information filtering is frequently seen in deep learning networks (see figure 2.6). Spatial filters remove unimportant patches while allowing valuable ones to pass. They do this by addressing characteristics of the input data's structure. Similar processes are carried out by temporal filters, but they operate on memory state sequences that contain information from the past. They are frequently employed to process sequences.

2.3.4.1.1 Spatial filtering (convolutional neural network): Many of the major breakthroughs in image processing have been made possible by the convolutional neural network (CNN). It can also be used for text analysis. When processing input data, a CNN applies a group of weighted filters (known as convolutions) and then learns the weights of these filters using training data. A CNN architecture is formed by a stack of several types of processing layers. The following are among the most important:

Convolutional layer: The foundational component of a CNN is the convolution layer. Its function is to identify whether a certain collection of features is present in the input. The idea is to ”drag” a window representing the feature on the input that represented by a matrix, and then figure out how much of each segment of the scanned input the feature convolutions with. The three concepts are equivalent, therefore a feature is then considered as a filter. A feature map is generated, for each pair (input, filter), showing where the features are in the input. The greater the value, the more the associated location in the input resembles the feature.

Pooling layer: By aggregating the outputs of neuron clusters at one layer into a single neuron in the subsequent layer, layer pooling reduces the dimensionality of the data. It can also compute a maximum or an average. The largest value from each cluster of neurons at the preceding layer is used in max pooling. Each neuronal cluster’s average value from the preceding layer is used in average pooling.

Fully connected layer: This layer is used by the neural network’s high-level reasoning after numerous layers of convolution and pooling. A layer that is fully connected has links to every output of the layer below.

Dropout layer: Technically, at each training stage, individual neurons are either removed from the network with probability p (often p=0.5) until only a small portion of the original network is remained. Since the majority of the parameters are taken up by a fully connected layer, codependency between neurons during training reduces each neuron’s own power and results in over-fitting of the training data. To avoid this, dropout is used.

Text can also be analyzed using CNNs. Textual objects like strings are typically 1D objects, which are horizontally oriented streams of characters that extend into one dimension. Consequently, when used on text, a CNN can find intriguing words or other qualities that are pertinent for a particular NLP task.

2.3.4.1.2 Temporal filtering (reccurent neural network): A universal approximator of dynamical systems is a recurrent neural network (RNN) [44]. It can be trained to accurately replicate any target dynamics up to a certain level. An RNN utilizes sequential information by simulating temporal dependencies in the inputs, such as the necessity to know the words that came before a word in order to forecast what will come next in a sentence. The current input and the value of the previous internal state, which keeps track of the history of all previous inputs, are both factors that affect the network’s output. RNN (Recurrent Neural Network) uses many iterations of the same sub-network or cell to interpret inputs from various sources. The new hidden state and the output at time step t are defined as follows given input x_t and the hidden state of the preceding step h_{t-1}:

$$h_t = \sigma_h(W_h x_t + U_h h_{t-1} + b_h) \tag{2.6}$$

$$y_t = \sigma_y(W_y h_t + b_y) \tag{2.7}$$

Where:
x_t is the input vector at time step, h_t is hidden layer vector, y_t is the output vector at time step t.
W, U, b are parameter matrices and vector.
σ_h, σ_y are the activation functions either a sigmoid function or tanh.

Handling sequencial data is the main purpose of this network creation in which inputs are split up into smaller units and then sent one at a time into network cells rather than being sent to the network all at once. Although they have had some success, rudimentary RNNs are still fairly basic temporal networks. They have short, limited-capacity memories, which explains why they struggle with extended sequences, and they indiscriminately reuse hidden states in their entirety without distinguishing between useless and valuable information. In order to overcome this restriction of conventional RNNs, the Long Short Term Memory Network, a modified form of RNN incorporating gating mechanisms, is developed.

Long Short-Term memory Networks (LSTMs): By adding gating operations to the transmission of previous network information into the present, LSTM [45] networks make an effort to address the shortcomings of simple RNNs. Each LSTM is made up of a number of sequential cells that all take the same input—time steps made up of discrete linguistic units like words—as input. The value of the hyperparameter ”number of cells” based on validation data. These cells' underlying data can be read out either for the full sequence or for each time step (for example, a word) by using the last cell. It is crucial to understand that LSTM cells encode contextual information; in the case of time-distributed data, they make this data available at local positions, whereas in the case of non-distributed data, they make it available globally (for instance, for an entire string of words). architecture composed with four gate: forget (f), input (i), memory (c) and output gate (o). Given an old memory C_{t-1}, the new cell memory C_t is computed as:

$$C_t = f_t \times C_{t-1} + i_t \times C_{tt} \tag{2.8}$$

Forget gate: selects the data that will be removed from the current memory. It is calculated from an input x_t at time step t as follows:

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{2.9}$$

$Ct - 1$ then gets multiplied with this f_t to transform it with some information removed.
Memory gate: where a new candidate memory is produced. It is calculated using an input x_t as:

$$C_{tt} = \tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{2.10}$$

Input gate: The amount of candidate memory information that will be inserted into the updated memory is decided by this gate. It is calculated using an input x_t as:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{2.11}$$

C_{tt} then gets multiplied by it to get the new added memory into the new memory cell.
Output gate: determines how much of the cell memory is extracted out. It is computed as:

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{2.12}$$

the new hidden state is then updated as:

$$h_t = o_t \times \sigma_c(C_t) \tag{2.13}$$

Having internal memory and having the capacity to update it progressively is believed to solve the issue of lengthy dependencies.

The gated recurrent units (GRU): Like a long short-term memory (LSTM), the GRU [46] has gating units, t but is much simpler to compute and implement.
The update gate: The update gate assists the model in deciding how much historical data from earlier time steps should be transmitted to the future. This gate is defined as:

$$z_j = \sigma([w_z x]_j + [U_z h_{t-1}]_j) \tag{2.14}$$

where σ is the logistic sigmoid function, and $[.]_j$ denotes the $j-th$ element of a vector. x and h_{t-1} are the input and the previous hidden state, respectively. W_z and U_z are weight matrices which are learned.
The reset gate: In essence, the model uses this gate to determine how much past data should be forgotten. It is calculated as follows:

$$r_j = \sigma([w_r x]_j + [U_r h_{t-1}]_j) \tag{2.15}$$

The actual activation of the proposed unit $h_j$is then computed as follows

$$h_j^{<t>} = z_j h_j^{<t-1>} + (1 - z_j)\tilde{h}_j^{<t>} \tag{2.16}$$

Where

$$\tilde{h}_j^{<t>} = ([W_x]_j + [U(r \odot h_{<t-1>})]_j) \tag{2.17}$$

The Bi-directional: Bi-directional RNNs [47] utilize a finite sequence, forecasting or identifying each element based on its previous and upcoming contexts. By combining the outputs of two RNNs, one of which processes the sequence from left to right (past to future), the other one from right to left(future to past),this is how it is accomplished [48].

2.3.4.2 Deep learning and NLP: A new paradigm

As language is made up of words that are arranged in sequences that move from the past into the present, the combination of spatial and temporal filtering opens up a variety of fresh, exciting possibilities for NLP. It is difficult to see how an abstract, intermediate layer representation would be processed spatially. Similar to this, we can learn to forget or keep specific parts of these intricate representations by using the time dimension to gate previous knowledge into this abstraction process.

2.4 Text embeddings

This section represents a review of most basic and popular text embedding algorithms in NLP. The procedures called embeddings are used to transform input data into vector representations. Every vector exists as a single point in a multidimensional vector space, where each value is interpreted as a value along a particular dimension. A systematic, well-designed process for projecting incoming data into such a space produces embeddings. Depending on how they are made, there are two main categories of vector encodings: procedural and representational.

2.4.1 Embedding by direct computation: Representational embeddings

For instance, they can be calculated directly from data using statistical techniques like straightforward counting. One-hot embedding is the most straightforward representational embedding for converting text to vectors. In such a vector space, related words should be located close to one another. This closeness is quantified by a distance function like the Euclidean distance, which applies the Pythagorean algorithm to determine the length of a straight line connecting two points in a Euclidean space. This category of spaces is having a fixed number of dimensions. Coordinates for each dimension are used to describe points in this space [43].

2.4.2 Learning to embed: Procedural embeddings

Using machine learning or statistics, procedural encodings are learned and calculated from data. For example the embedding layer that is trainable creates weights matrix which is tuned

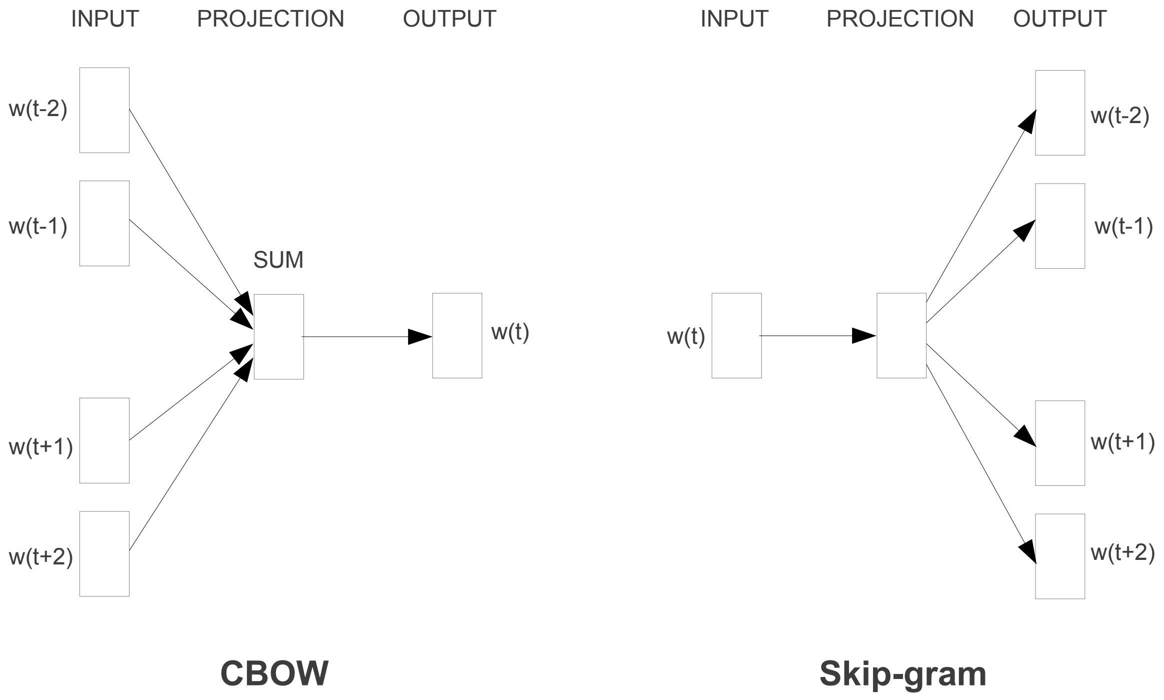


FIGURE 2.7: The CBOW architecture predicts the current word based on the context, and the Skip-gram predicts surrounding words given the current word [30].

during training. They are hence small, shallow mininetworks (one hidden layer). A loss function is implicitly minimized by embeddings. In other words, they tailor their representations to a particular criterion. Maximizing the uniqueness of the vector representations so that the confusability of any two vectors kept to a minimal is the implied default criterion [43].

2.4.3 From words to vectors: Word2Vec

Word2vec [30] creates a vector space from a large corpus of text as input, assigning a corresponding vector to each distinct word. If two word vectors in a corpus share the same context, they are positioned close to one another in the vector space. The Continuous Bag-of-Words (CBOW) form of the Word2Vec algorithm predicts words from contexts, whereas the contexts from words variant predicts words from contexts (the skipgram variant) , as illustrated in figure 2.7.

2.4.4 From documents to vectors: Doc2Vec

There are no word-level restrictions on embeddings. Larger language elements like sentences, paragraphs, and even whole publications can be embedded. By using vectors to describe texts, we may search for similarities in the generated vector space, much to how words have semantic

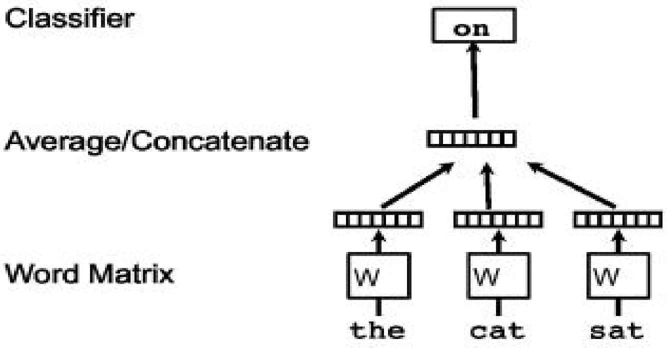


Figure 2.8: A framework for learning word vectors. Context of three words (“the,” “cat,” and “sat”) is used to predict the fourth word (“on”). The input words are mapped to columns of the matrix W to predict the output word [49].

similarity. Le and Mikolov’s original research [49] on paragraph vectors suggests a stylish expansion of Word2Vec to encompass complete manuscripts. Their strategy is referred to as Doc2Vec. Simple is the idea, we start by giving each document a special identification, such as a filename or an integer. The plan is to combine these document identifiers with a word-based embedding of their content in a second embedding (see figure 2.8).

2.4.5 GloVe

Global Vectors for Word Representation [50] is the full name of the GloVe. The GloVe model is a log- bilinear model in which the probability of the following word is determined when the preceding words are given, indicating that the statistics of word occurrences in a corpus is the all unsupervised approaches for learning word representations’ primary source of knowledge. Using statistics from the entire text corpus, this model creates an explicit word-context or word co-occurrence matrix. The outcome is a learning model that might lead to more effective word embeddings overall.

In Table 2.1, the researchers [50] provide a simple illustration based on the words ”ice” and ”steam” to demonstrate their point. The ratio of these terms’ co-occurrence probability with different probing words can be used to determine how these words are related. $P(solid|ice)$ will be relatively high, and $P(solid|steam)$ will be relatively low. Therefore, the ratio of $P(solid|ice)/P(solid|steam)$ will be large. The ratio of $P(gas|ice)/P(gas|steam)$ will be modest if we consider a word like gas that is connected to steam but not to ice. We anticipate the ratio to be close to one for a word like ”water” that is connected to both ice and steam.

The GloVe is a pre-trained model, where a 2014 dump of the English Wikipedia was used to generate a dataset of one billion tokens (words) for the training, which had a vocabulary of 400 thousand words. The GloVe has embedding vectors with dimensions of 50, 100, 200, and 300. The 300-dimensional one typically produces good results. If we go within the file,

Table 2.1: Pennington *et al.*'s Example [50].

Probability and Ratio	K = solid	K = gas	K = water	K = fashion
P(K|ice)	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
P(K|steam)	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
P(K|ice)/P(K|steam)	8.9	8.5×10^{-2}	1.36	0.96

we will notice that each line starts with a token (a word), then the weights (300 numbers)(e.g. dismissal 0.35632 -0.15902 0.27487 -0.14592 0:022154 -0.78008 0.53658 -0.34398 ....). The GloVe model has many advantages: Greater accuracy is done for the same amount of training, faster training, improved RAM/CPU efficiency (can handle larger texts), and more effective data utilization (helps with smaller corpora) [51].

2.4.6 FastText

The last initiative is FastText, which is an extension to Word2vec for text representation and classification, it proposed by Facebook in 2016 after regrouping the results for two methods represented in [52] and [53]. Simply, it divides words into several sub-words (n-grams) rather than providing individual words to the Neural Network. This approach is helpful for multiple reasons like getting better word embeddings for rare words and having a vector for words from its character n-grams even if the word doesn't appear in the training corpus. Unlike Word2vec, it does not simply anticipate the words around it; it also predicts the surrounding n-character gram. The word "whisper," for instance, would produce the 2- and 3-character gram wh, whi, hi, his, is, isp, sp, spe, pe, per, er quickly. For every n-character gram, including words, misspelled words, incomplete words, and even single characters, text teaches a vector representation [51].

2.4.7 BERT

Bidirectional Encoder Representations from Transformers, or BERT, is a deep learning technique that was first introduced by Google researchers in a study published in late 2018 [38]. Similar to Word2Vec, BERT seeks to derive word embeddings from raw textual input. However, it is performed in a far more sophisticated and effective way: when learning vector representations for words, BERT takes into account both the left and right contexts (figure 2.9.). Word2Vec, in comparison, only makes use of one context element. However, this is not the sole distinction. Unlike Word2Vec, BERT is based on attention and uses a deep network (recall that Word2Vec uses a shallow network with just one hidden layer.)

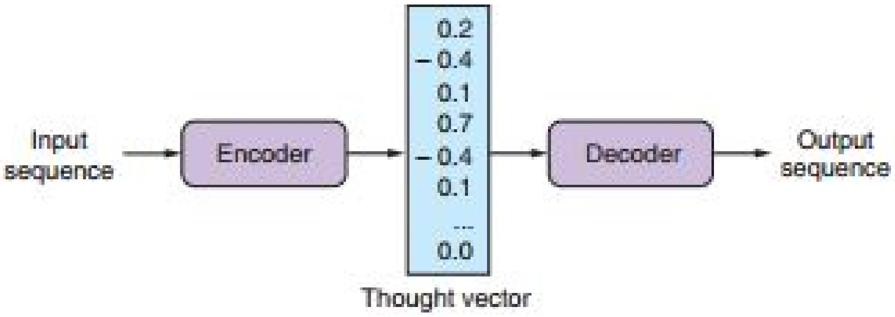


FIGURE 2.9: Autoencoder architecture [51].

2.5 Sequence-to-sequence models

Neural Networks (NNs) are only useful for problems whose inputs and targets can be coherently expressed using vectors of fixed dimensionality, in spite of their flexibility and power. Given that many essential problems are best stated with sequences whose length are unknown beforehand, this constraint is significant. For instance, developing a chatbot discussion, translating from one language to another, or summarizing a text are all related to sequential issues. It follows that a method that learns to map sequences to sequences and is domain independent would be helpful.

Sequences provide an issue since they need the dimensions of the inputs and outputs to be known and fixed. To handle this language challenges, seq2seq, a special type of recurrent neural network topologies, is often used. The Encoder-Decoder or autoencoder architecture is the most popular type of architecture used to construct Seq2Seq models (see figure 2.9).

2.5.1 Autoencoder architecture

Due to the significant lag in time between the inputs and their associated outputs, LSTMs are excellent at managing sequences and have the capacity to learn from data including long-range temporal relationships. The idea behind the autoencoder is to employ two components—an encoder and a decoder—to achieve massive fixed-dimensional vector representations of the input and output sequences, respectively. The first half of the autoencoder is an LSTM where at a timestep, it scans the input sequence, and to extract the output sequence from that vector using another LSTM acting as a decoder.

2.5.1.1 The encoder part

The encoder, which is a network that analyzes the input sequence, such as natural language text, and compresses the information into something known as the thought vector, hidden state, or context vector, is the initial component of an autoencoder model (these are known as the hidden state and cell state vectors in the context of LSTM). Only the internal states of the encoder are kept; its outputs are ignored. This context vector aims to incorporate data for all input items to help the decoder make accurate predictions. The following formula is used to calculate the hidden states h_t:

$$h_t = f(W^{hh}h_{t-1} + W^{hx}x_t) \tag{2.18}$$

The data are read by the LSTM one sequence at a time. We therefore state that LSTM reads the input in time steps of length 't' if the input is a sequence of that length.

1. X_i = Input sequence at time step i.

2. h_i and c_i = At each time step, the LSTM maintains two states ('h' for hidden state and 'c' for cell state). These represent the LSTM's internal state at time step i when taken as a whole.

3. Y_i = Output sequence at time step i. Y_i is essentially a probability distribution created by a softmax activation across the whole vocabulary. Thus, each Y_i is a vector expressing a probability distribution of size "vocab_size".

2.5.1.2 The thought vector

In order to represent the substance of the input text, Any natural language sentence can be compressed with information using a neural network into a fixed length vector or what is called a thought vector. It is a numerical representation of the thought within a document to drive some decoder model.

2.5.1.3 The decoder part

The sequence decoder is the other component of an encoder-decoder design. This LSTM has starting states that are initialized to the encoder LSTM's end states, i.e. the first cell of the decoder network receives the context vector from the encoder's final cell as input. The decoder decompresses the input again from the thought vector to predict (create) output sequences. These starting states serve as the basis for the output sequence that the decoder generates. Subsequent outputs also take into account the initial outputs.

Given time step t, an output y_t is predicted by a group of LSTM component. Each recurrent unit produces both an output and its own hidden state after receiving a hidden state from the preceding unit. We compute any hidden state h_i using the following formula:

$$h_t = f(W^{hh}ht - 1) \tag{2.19}$$

The output y_t at time step t is computed using the formula:

$$y_t = softmax(W^S h_t) \tag{2.20}$$

The current time step's hidden state and the appropriate weight are used to calculate the outputs. Using Softmax, We could create a probability vector that would allow us to forecast the outcome (e.g. the question-answering issue words).

The initial states (h_0, c_0) of the decoder are set to the encoder's final states, which is the most crucial factor. This naturally leads to the decoder being instructed to start creating the output sequence using the data that the encoder has encoded.

Through time, the errors are back propagated and the loss is computed using the anticipated outputs from each time step in order to update the network parameters. The network can produce quite accurate predictions after being trained for a longer time with sufficient data.

The learning ability of autoencoders is limited since they must construct and deconstruct the input rather than just replicate it. It must decide which elements of the input to learn in order of importance because of this restriction in its options.

2.6 Transfer learning

2.6.1 Transfer learning definition

One of the famous definition of this concept is: ”Situation where what has been learned in one setting is exploited to improve generalization in another setting” [54]. Given the time and effort required to label data points, obtaining large quantities of labeled data for supervised models can be very challenging. This is because the majority of traditional deep learning models that handle complicated problems require a lot of data. It is difficult to find such a dataset for every domain, though. In addition, the majority of deep learning models are highly tailored to a certain field or even task. This serves as the driving force behind transfer learning, which looks beyond particular tasks and domains and investigates how to employ information from previously trained models to address brand-new issues.

2.6.2 Transfer learning understanding

The information (features, weights, etc.) can be used from older models that have already been trained to train newer models and even work around issues like the new task requiring less data.

As a result of transfer learning, we ought to be able to use the knowledge from earlier activities and apply it to more recent. If task T1 has considerably more data, we can use that learning to generalize the features and weights for task T2 (it has fewer data). Edges, shapes, corners, and intensities are a few low-level attributes that can be shared throughout computer vision tasks and aid in knowledge transfer. Additionally, prior experience with one activity serves as supplementary input while learning a new target task.

So, we can give the following formulation of transfer learning: It seeks to enhance the learning of the target predictive function$f_T()$ in the target domain D_T using the information in the source domain D_S and learning task T_S, where $D_S \neq D_T$, or $T_S \neq T_T$ [55].

2.6.3 Transfer learning strategies

Depending on the domain, the task at hand, and the data availability, multiple transfer learning strategies and approaches may be used. Transfer learning techniques can be grouped according to the type of conventional machine learning algorithms used [55]:

- Learning through inductive transfer: the learning data are labeled and fall within the same domain, and the tasks to be performed are similar (example: recognizing a cat and a dog).

- Unsupervised Transfer Learning: This environment is comparable to inductive transfer by concentrating on unsupervised tasks in the target domain. Although the tasks are distinct, the source and target domains are comparable. In this case, neither of the domains has access to labeled data.

- The source and target tasks in this case are similar, but the corresponding domains are distinct, which is known as transductive transfer learning. There is a lot of labeled data in the source domain for this circumstance but none in the destination domain. This can be further broken down into scenarios where the marginal probabilities of the feature spaces diverge.

2.6.4 Applications of transfer learning

It is still a new method, yet it is already used in many machine learning applications. Transfer learning is already used in a variety of real-world situations, whether it's enhancing computer vision or natural language processing.

2.6.4.1 Transfer learning for natural language processing

Employing transfer learning, natural language processing-focused machine learning models can be improved in a number of ways. The use of pre-trained layers that understand specific dialects or vocabularies is one example, as is simultaneously training a model to detect a variety of linguistic components.

Transfer learning can also be used to alter language translation models. For similar activities or languages, models that were created and trained using English can be adjusted in several ways. Models can be trained on a huge dataset before having their components translated to a model for a foreign language because of the ubiquity of digitized English language content. NLP domain contains many pre-trained word embedding model such as: Word2Vec (section 2.4.3), GloVe(section 2.4.5), FastText(section 2.4.6). Excellent developments in transfer learning for NLP have recently been made. Most importantly: Universal Sentence Encoder by Google, Bidirectional Encoder Representations from Transformers (BERT) by Google(section 2.4.7).

2.6.4.2 Transfer learning in computer vision

The ability of systems to comprehend and derive meaning from visual representations like films or photos is known as computer vision. Large image collections are used to train machine learning algorithms to identify and classify image subjects. In this example, To apply transfer learning to a new model, reusable parts of a computer vision algorithm will be utilised.

Transfer learning can assist in applying the precise models created from huge training datasets to smaller sets of images. This includes transferring the model's more universal features, such as the method for spotting objects' edges in pictures. The model's more detailed layer, which is responsible for classifying different kinds of objects or shapes, can then be trained. The most pre-trained models known in this field are: VGG-16, VGG-19, Inception V3.

2.7 Conclusion

Deep learning and machine learning continue to spread across a variety of industries and have transformed NLP topic. Computer science's ”natural language processing” (NLP) field aims to make it possible for machines to comprehend language similarly to how people do. This usually refers to duties like deciphering text's sentiment, speech recognition, and coming up with questions to answer. This chapter presents how NLP becomes a large portion of artificial intelligence (AI) breakthroughs. It walks through some introductory NLP techniques, such as word tokenization, cleaning text data, term frequency, inverse document frequency, and more. And It describes the main NLP implementations using deep learning.

Chapter 3

A Survey of Fake News Detection Models on Social Networks.

*****

3.1 Introduction

Nowadays, the Internet offers a wealth of knowledge through a variety of documents. In fact, the social media has become a preferred source of information for the internet users. The users need a simple method for filtering these online documents to identify the ones that are best suited to their desideratum, preoccupations, and competences. Furthermore, the documents classification and indexing, the detection of false information or unlawful Web activity may also be required by the users. In our work we focus on detecting and decreasing the false information dissemination on social media documents which is a challenging problem. This chapter gives a definition of social media document, its properties. Also it provides the definition of fake news, its types and its sources, beside the different detection methods of both fake news and social bots. Hence, we point out the basic concepts and discuss the key ideas and the issues that motivate our work.

3.2 Documents and NLP Areas

The document is any information-supporting, information-communicating, and long-lasting object that is probably to be used for reference, research, study and experiments for example: charts, illustrations, textual documents (papers, newspaper...etc) [56,57] (see figure 3.1). The document can be non-digital which physically saved in file folder, or digital (electronic). The

FIGURE 3.1: Documents Type.

digital document is kept as one or more files in a computer or on other digital devices and it can then be included in a database. The process or group of processes used to gather, organize, and use documents is known as documentation.

3.2.1 Social media documents

The social media is a computer-based technology and it appeared in 1970's. It enables viewing, producing, or distributing concepts, ideas, and knowledge through online groups and networks, and it became an integrated part of billions of individuals worldwide daily lives. In industries like education, shopping, and politics, social media is also employed by businesses, governments, and other entities. Blogs, wikis, and various forms of digital social networks (Twitter, Facebook, Snapchat, Instagram..) were all included in social media and Web 2.0 categories, which still have a tight relationship.

All throughout the world, social media networks produce data constantly and nonstop. As a result, there has been a massive multiplication of documents, which might take the form of text, photographs, videos, audio, or Geo-locations.

3.2.2 Social media documents proprieties

The utilization of social media generates both structured and unstructured types of data. Text in social media posts is unstructured data and it can be short like the tweets or long as the online newspaper articles [58], whereas friendships, followers, groups, and networks are structured types of data. The potential for deep insights on attitudes, behavior, news, and more lies within the complete range of social media data.

The access to the social media documents can be public, private or secret. Most of the time, anyone who wants to access or read public documents can do so without charge. A few of these are books, press articles public records, and information stored and kept by government organizations. In several nations, laws regulate public record usage and access.

Only users who have been given permission by a ruling party to access private papers can do so. A company's IT department, senior management, the original author, or another party could be this one. Every authorized user must enter a password in order to open or edit a digital document, which can be protected in this way. These documents can also be encrypted or kept in a secure digital place that is only accessible to authorized users after completing some type of verification, like a password or multi-factor authentication. A very small number of authorized people often have access to secret or classified documents. These records may be owned by a business, a government, a military unit, etc. Governmental classified material is always safeguarded by encryption, access control, and security clearances. Additionally, the law may restrict access to certain documents, typically on a need-to-know basis. Finally, whether intentional or unintentional, mishandling, losing, or compromise of classified documents can result in criminal charges against the responsible party.

3.3 Fake news: Concepts and definition

Consumers are the norm for users in conventional media. From the publisher to the user is the only direction in which information can circulate. Social media deviates from this paradigm by enabling simultaneous publication and consumption for all its users [59] and that has prompted the false information proliferation, which has become the disease of the age because of the real danger it poses to the whole world.

Fake news has a long history, almost going back to the invention of the printing press in 1439 [60], and the first recorded negative impact to this news it dates back approximately to 1782 when in order to advance the cause of American independence, Benjamin Franklin claimed that British forces had recruited Native Americans to slaughter and scalp American soldiers, women, and children in Boston, and caused panic among a million Americans [61]. Over time, the meaning of ”Fake News” has changed, making it challenging to come up with a common definition of the issue. Overall, experts classify false news into three primary categories: hoaxes perpetrated on a big scale, intentionally false news, and false news that is taken seriously [62].

It's hard to believe that before Donald Trump became president of the United States, the phrase ”fake news,” which he popularized, was hardly ever uttered. When hundreds of websites published stories that were either false or blatantly biased throughout the 2016 campaign for the United State presidency, and many of them did so in an effort to profit from social

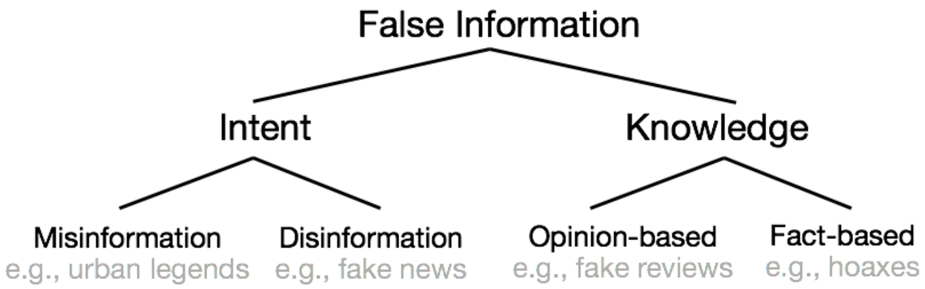


FIGURE 3.2: Categorization of false information based on intent and knowledge [63].

media advertising revenue, giving rise to this term, which has since gained widespread acceptance. Furthermore, there are many reasons why the prevalence and significance of false information are increasing. Some are written from scratch with an alluring headline to increase traffic and the number of visitors to the website, while others try to deceive the reader or affect his view on a specific subject. The ease with which websites may now be created or even altered to include specific content, the suitability of social media for spreading fake news, and the proliferation of online social media are a few examples of these goals.

3.3.1 Fake News Categories

The intention and knowledge content of false information might be used as the main concepts to identify their classification [63] (see Figure 3.2).

3.3.1.1 Categorization Based on Intent

Based on intent, information is divided into two main categories: misinformation and disinformation [64, 65]. Without intending to deceive, the first information category is shared. The person's lack of comprehension or cognitive biases lead to inaccurate representations of the true information, which are then unintentionally shared with others through tweets, blogs, and other channels. Disinformation is created with the purpose to mislead for a variety of objectives, such as swaying public opinion or sending web traffic to specific websites in order to generate revenue from advertising.

3.3.1.2 Categorization Based on Knowledge

Here, false information is divided into two categories: fact-based and opinion-based [63,66]. In cases where there is no absolute truth, the first category expresses the individual opinion. Such viewpoints may be inaccurate and careless, which may influence readers and their choices. The second type of information comprises material that runs counter to fundamental truths with the intention of making it more difficult to distinguish between true and false information. This type also includes created lies, rumors, and false news.

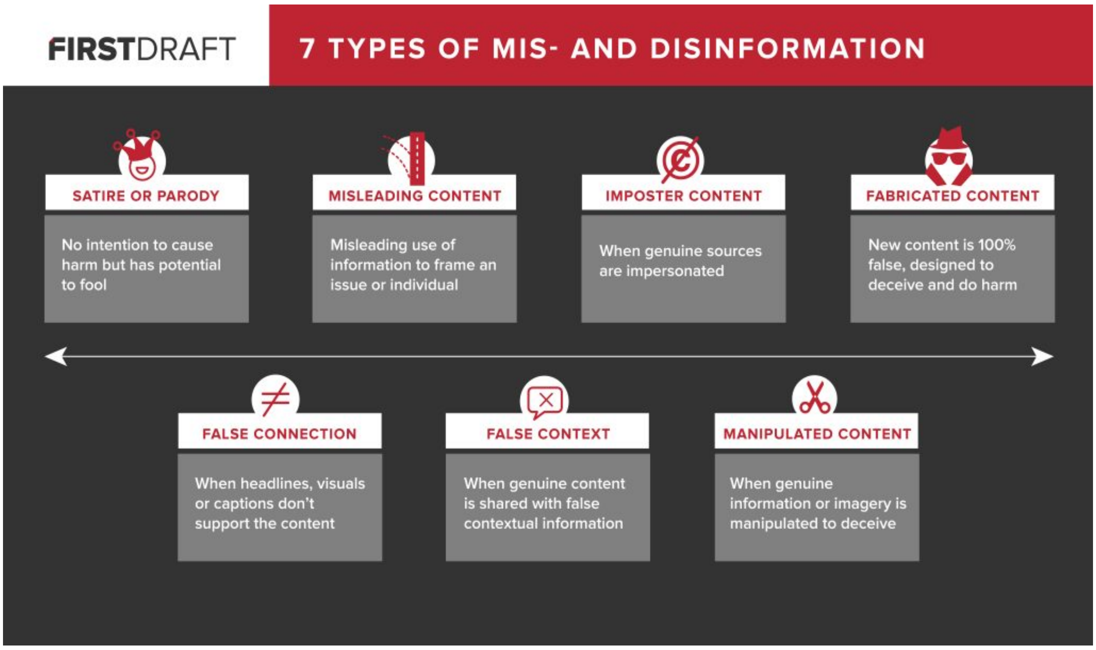

FIRSTDRAFT
7 TYPES OF MIS- AND DISINFORMATION
SATIRE OR PARODY
No intention to cause harm but has potential to fool
MISLEADING CONTENT
Misleading use of information to frame an issue or individual
IMPOSTER CONTENT
When genuine sources are impersonated
FABRICATED CONTENT
New content is 100% false, designed to deceive and do harm
FALSE CONNECTION
When headlines, visuals or captions don't support the content
FALSE CONTEXT
When genuine content is shared with false contextual information
MANIPULATED CONTENT
When genuine information or imagery is manipulated to deceive

FIGURE 3.3: The seven common forms of fake news [67].

Due to their use of criticism or cultural commentary, there are other seven frequent kinds of false information (see Figure 3.3) that are examined individually in addition to precedent categories [68–70]:

- **Satire or parody:** no malicious intent, yet has the potential to be deceptive.
- **Misleading content:** using false facts to model a case of a situation or person.
- **Imposter content:** related to mimic legitimate sources.
- **Fabricated content:** the novel content, which is wholly misleading and intended to mislead and cause damage.
- **False connection:** when the information is not supported by the masthead, visuals, or notes.
- **False context:** when accurate content is provided with incorrect context.
- **Manipulated content:** when real facts or images are modified to spread false information.

3.3.2 Fake News Proliferation Sources

It is quite difficult to monitor and investigate the sources of information being disseminated as well as the characteristics of its consumers because to the Internet's wide range and lack of its users identity. That is because nobody wants to expose their genuine identify, which is

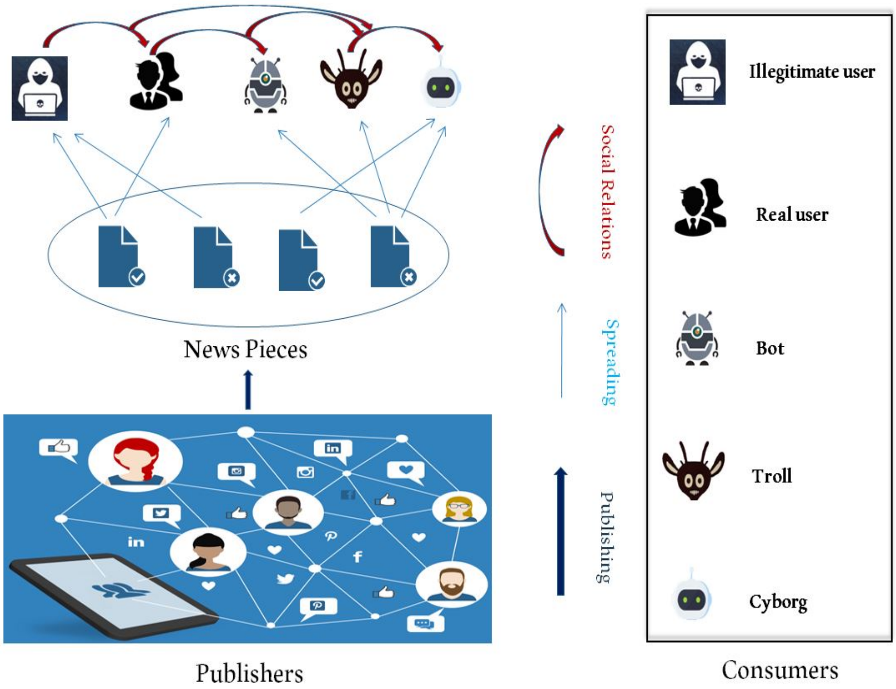


FIGURE 3.4: Fake news proliferation sources: Figure shows that during the news spreading process, many news pieces were published by several publishers from different social platforms, these news are shared by consumers who have engaged in social media and related with each other by social relations [5].

illegal in this situation. These users may even not be human at all. All of these circumstances contributed to the problem of malicious accounts, which emerged as a significant source of the spread of fake news [5]. We illustrate a news dissemination process in Figure 3.4.

3.3.2.1 Social Bots

It is a computer algorithm built especially to refer accounts that produce material automatically and communicate with people on social media by attempting to mimic their online behavior, such as time activity, propagation patterns, and emotional expression. Some of these bots were created to offer valuable services, while others may be destructive, such as when they help to sway public opinion, manipulate the stock market, steal people's personal data, and propagate false information [71].

3.3.2.2 Cyborg Users

After registering for a social media account, some users have the option to establish automatic programs to amplify statements in his absence; these users should be regarded as cyborgs. Contrary to bots, which heavily rely on automation, cyborgs combine the qualities of both manual and automated activity [72].

3.3.2.3 Trolls

A user that intentionally incites retaliatory, abusive, or provocative comments from others is known as a troll. Its objective is to cause distress and elicit an emotional response (preferably anger). To accomplish this, a troll may frequently engage in conversation unrelated to the suspended part, launch advertising-related attacks, issue death threats, or use derogatory language [73].

3.4 Fake News Detection Methods

From newspapers to radio to television, or what is known as conventional false news, the news environment has changed over time. Social media has just taken off and is now a major player in the spread of false information [60]. The spread of these stories on social media depends on a number of variables, including the information's content and the actions of its users.

Fundamentally, and in order to understand the two sides of both fake and real news, valuable features can be extracted from the news content, such as linguistic features, which include lexical features like the frequency of a particular word in the text, exploiting syntactic features (like the degree of clausal embedding, the presence of coordination, the type of speech act, etc.), and word similarity measures, also known as semantic features. Various visual indicators, such as the similarity distribution histogram and the clustering score, have also been derived from visual elements, such as pictures and videos, to capture the various aspects of news verification [74].

The social interactions that users have when consuming news on social media platforms can also yield a wealth of useful information. For instance, user profiles can be captured at two levels: (i) the individual level, such as registration age, number of followers/followees, and number of tweets the user has written [75]; and (ii) the group level, such as the percentage of verified users' and the average number of followers [76, 77].

Furthermore, as spotting fake news is crucial, the features extracted from social media networks that link individuals together should be taken into account. These networks were created based on relationships, interests, and various topics that cause social media users to group together like-minded individuals where they polarize their opinions, creating an echo chamber cycle effect. As a result, the network features were extracted by building specific networks among the users who published related social media posts. We specifically mention the friendship network, which shows the follower/followee structure of users who posted related tweets [76]. The diffusion network, which follows the trajectory of news dissemination, is an extension of this friendship network [76], where nodes stand in for users and edges for the information diffusion pathways between them.

If consumers have any suspicions about any information they receive, they can act right away. To ensure it is not a joke or a rumor, they can independently confirm the source, date,

HOW TO SPOT FAKE NEWS

CONSIDER THE SOURCE
Click away from the story to investigate the site, its mission and its contact info.

READ BEYOND
Headlines can be outrageous in an effort to get clicks. What's the whole story?

CHECK THE AUTHOR
Do a quick search on the author. Are they credible? Are they real?

SUPPORTING SOURCES?
Click on those links. Determine if the info given actually supports the story.

CHECK THE DATE
Reposting old news stories doesn't mean they're relevant to current events.

IS IT A JOKE?
If it is too outlandish, it might be satire. Research the site and author to be sure.

CHECK YOUR BIASES
Consider if your own beliefs could affect your judgement.

ASK THE EXPERTS
Ask a librarian, or consult a fact-checking site.

IFLA
International Federation of Library Associations and Institutions
With thanks to www.FactCheck.org

Figure 3.5: The International Federation of Library Associations and Institutions infographic on how to spot fake news [78].

and author or consult professionals or trustworthy fact-checking websites. Reputable organizations in numerous sectors, including the International Federation of Library Associations and Institutions, routinely recommend following this guidance (see Figure 3.5).

In other side, numerous methods have been used by experts to analyze the aspects of fake news; we classify these methods into three groups: unsupervised method, semi-supervised method and supervised method.

3.4.1 Approaches Based on Unsupervised Learning

In addition to supervised learning, unsupervised learning is one of the main areas of machine learning. It is a self-organized learning method based on finding hidden characteristics in unlabeled data sets. Here, we point out and discussed the relevant works that deal with this challenge.

Hosseinimotlagh *et al.* [79] proposed an approach to group fake news into various groups. They first used a multiple tensor decomposition method to refine clusters into a single, high-quality, and high-coherence set of documents by clustering the documents in tensors based on the appearance positions of each term in an article and its correlations with other terms (Spatial relation extraction). The findings attain greater coherence and pinpoint every type of fake news within actual data.

To address the challenges of unsupervised detection of fake news with unreliable social commitments, Yang *et al.* [80] proposed an unsupervised framework that examines users' social media engagements to determine their thoughts on news, and builds a graphical Bayesian probability model. This model captures the generation process of user opinions and news truths. In addition, the authors evaluated user credibility using a powerful collapsed Gibbs sampling method [81].

The main task in the claims verification process, recognizing evidence sentences, was investigated by an unsupervised approach in a recent study by Deka *et al.* [82]. A key task is stance identification, which supports numerous downstream tasks like predicting the spread of false news. Pick *et al.* [83] suggested a framework for stance detection. The framework is independent of domain and unsupervised. They build the interaction network from which they extract topological embedding for each speaker given a claim and a multi-participant conversation. These speaker embedding have the advantage that speakers with comparable stances are frequently represented by similar vectors, but speakers with opposite stances are typically represented by antipodal vectors. In order to detect fake news from various domains, Silva *et al.* [84] proposed a novel framework that simultaneously in news records, retains knowledge from several specific and cross domains. They then introduced an unsupervised technique to pick some important news records that haven't been labeled for manually labeling, which can then be utilized for the false news detection model training that excels across a wide range of domains. Li *et al.* [85] suggested an unsupervised autoencoder-based false news detection approach (UFNDA). This research initially considers a few types of news on social networks and includes the text content, photos, publication, and user information which relates to the news dissemination in order to increase the efficacy of false news identification. Next, to recover the intrinsic relationships between attributes and hidden information, the autoencoder is enhanced with a Bidirectional GRU(Bi-GRU) layer and a Self-Attention layer before reconstructing the residual to identify false news.

3.4.2 Approaches Based on Supervised Learning

Contrary to unsupervised techniques, supervised learning strategies have found extensive usage in the detection of misleading information.

The topic of phony followers was noticed by Cresci *et al.* [86]. They provided a list of Twitter accounts in the first step that served as a reference for both authentic users and false accounts. Following that, they used algorithms built on feature sets and a single categorization rule. According to the analysis's findings, algorithms based on categorization criteria are ineffective in spotting frauds. However, using a feature designed to detect spam-bots allows you to identify false followers with good accuracy.

Because of its transparency and instantaneous features, the microblog has emerged as one of the most significant news outlets in contemporary culture. But it's also a source of false information. Jin *et al.* [87] proposed a three-layer hierarchical propagation model to evaluate the veracity of news on micrologging. Their relationships and the process of establishing credibility can be reasonably modeled using the hierarchical structure of the message as a sub-event. The ability to recognize deeper semantic information for each event using a sub-event layer enhances the ability to identify fake news.

Ma *et al.* [77] emphasized the significance of the social surrounding elements' fluctuation over the message's long-term dissemination. Through the use of Support Vector Machines (SVM), they suggested a method to capture the temporal properties of these elements based on the time series of rumor's life cycle. The findings indicate an improvement in the rumors detection.

Ciampaglia *et al.* [88] suggested mapping the fact-checking problem to the well-known challenge of finding the shortest path in a graph utilizing the information provided by knowledge networks since the amount of information that is currently generated online makes the traditional fact-checking very challenging. In that situation, a shorter path denotes a higher likelihood of a true assertion. The results demonstrate that network analytics techniques in combination with extensive knowledge repositories present a new avenue for developing automated fact-checking techniques.

Lendavi *et al.* [89] created classifiers to deal with noisy text utilizing similarity features extracted from the string and part-of-speech level, based on Nearest Centroids (NC) and Random Forest (RF), since a textual divergence among social media posts can be a symptom of rumor. This study turned out to be an excellent foundation for categorizing contradictions. Additionally, one of the key tasks in computational journalism is the detection of contradiction and disagreement in micro-posts, which provides crucial indications to factuality and veracity assessment. Based on a comprehensive feature set, Hardalov *et al.* [90] suggested a language-independent method for automatically differentiating real news from fraudulent news. They specifically used (a) linguistic features, such as n-grams, (b) credibility features, such as the length of the article (number of tokens), the number of distinct punctuation, the percentage of plural pronouns, the number of URLs, and (c) semantic features utilizing embedding vectors. Then, based on self-generated data sets, each feature category was independently tested using the logistic regression classifier. According to the findings, it is quite accurate to distinguish between news that is reliable and news that is not.

The task of categorizing the stance of a news article's headline and the claim it makes was investigated by Ferreira *et al.* [91] using an emerging project data-set [92] that provides a rich source of labelled data (claims and related press articles). Each article title was given a stance label, indicating whether the piece is supporting, disputing, or merely reporting the assertion. They created a multiclass logistic regression-based stance classification method that used two different kinds of features: those taken solely from the article headline and those extracted by fusing the headline with the assertion. This method provides a high level of accuracy and shows how the paraphrase-based features, word alignment, and syntactic features all help to improve performances.

Using dimensional reduction, n-gram features, and a quick approximation of the softmax classifier, Joulin *et al.* [53] suggested a text classification model. The product quantization method is the foundation of this rapid text classifier [93], which minimizes the softmax loss l over N documents and provides accurate results with minimal training and evaluation time. A publicly accessible data set for the detection of fake news is presented by Wang [94] as LIAR [95]. The author created a hybrid convolutional neural network (CNN) to merge text and metadata in order to examine the automatic detection of fake news based on surface-level language patterns. In comparison to a deep learning model that uses solely text, the hybrid model showed good performance.

Additionally, Ruchansky *et al.* [96] proposed a three-part architecture; the first module is a recurrent neural network to record the temporal pattern of user activity on articles, the source characteristic is taught in the second module based on user behavior, and the third module identifies whether an article is fake or not. Long *et al.* [97] also suggested a hybrid LSTM that operated on two independent LSTMs, the first is used to acquire the representation of press articles. Then, two attention factors are built using the speaker's profile (including party affiliations and speaker position title). One just makes use of the speaker profile, and the other makes use of press articles' subject information. To obtain the speaker vector presentations, the second LSTM merely employs the speaker profiles. The soft-max function is used for classification. Moreover, Volkova *et al.* [98] used Tweeter data to forecast if a news piece is suspect or verified and categorize it into fine-grain subsets of suspect news (satire, hoaxes, clickbait and propaganda). The linguistic neural networks with linguistic features were used by the authors. Their research provides the idea that whereas syntactic and grammatical features have no bearing on fine-grained classification, linguistic features do.

Numerous studies have shown that images play a significant role in the spread of news on microblogs. In order to enhance the verification performance, Jin *et al.* [74] focused on the visual content in tweets. In order to characterize the image distribution patterns from fake and real news events visually and statistically, as well as to predict the veracity of the corresponding articles, several visual features (Clarity Score, Clustering Score, Similarity Distribution Histogram, etc.) and statistical features (the number of all images in a news event, the ratio of image number to tweet number, the ratio of the most popular image in all distinct images, etc.) have been proposed. Additionally, they employed a technique for spotting fake news that is built on a reputation propagation network and incorporates conflicting opinions taken from tweets.

The multi-modal fake news detector, the fake news detector, and the event discriminator are the three primary parts of the Event Adversarial Neural Network (EANN) that Wang *et al.* [99] suggested as a solution to the problem of multi-modal fake news detection. The textual and visual features from postings are extracted by the multi-modal feature extractor. In order to learn the discriminate representation for fake news detection, it works in tandem with the fake news detector.

Through categorizing news propagation paths, Liu *et al.* [100] established a model for the early detection of bogus news on social media. To start, they created a multivariate time series model of how each news story propagated, with each tuple reflecting a set of user characteristics represented by a numerical vector. Along the propagation channel, to measure the global and regional differences in user attributes, the authors constructed a time series classifier that combines both recurrent and convolutional networks. The experimental findings show that bogus news may be quickly identified after its initial distribution points.

Additionally, to identify fake news, Shrestha [101] integrated sentiment analysis with metadata with language cues from the content. The author used a hybrid strategy based on a web-based application and a machine learning model. This project's initial phase involved creating a web interface that accepts a news URL as input and asks the back-end Flask server for a prediction. The backend fetches text content, metadata, and Facebook trend metrics from a URL using API calls (reactions, shares, and comments). The trained machine learning model, which consists of four sub-pipelines: text, sentiment, numeric, and hashing pipeline, is then given the accumulated information from the news source. To predict the news's reliability, each of these sub-pipelines extracts the features and passes them for training to a random forest classifier. Following the prediction, the web user interface returns the outcome of the forecast.

Qian *et al.* [102] suggested an efficient approach based just on the texts in the detection phase for the early detection of false news. The model is a Two-Level Convolutional Neural Network (TCNN) with a User Response Generator (URG), additionally, it can successfully absorb news articles characteristics by condensing information from the word level to the sentence level in order to effectively capture semantic information from longer article texts. A user's prior replies are used by the URG to understand how users react to article material, which aids in the identification of fake news.

Another method for spotting fake news was put up by Tschiatschek *et al.* [103], who used the strength of crowd signals (users' flagging activities) to choose a small subset of k items, send them to a specialist for examination, and then suppress the false information. Due to the investigative Bayesian algorithm's ability to detect fake news with high confidence and learn over time about users' accuracy, this method helps to reduce the spread of false information.

Recently, By combining macro-level (which comprises nodes from: retweets, news and tweets) and micro-level information (it denotes the discussion tree that reply nodes represents), Shu *et al.* [104] created hierarchical propagation networks, then they used this method on the FakeNewsNet, an online database for fake news detection [105, 106]. The authors developed Gaussian Naive Bayes, Decision Tree, Logistic Regression, and Random Forest models for each type of network, extracting and comparing various aspects from structural, temporal, and linguistic perspectives for false and real news. The authors assessed how well the extracted features work. The results of the experiments demonstrate that (1) macro-level and micro-level features can significantly aid in the detection of fake news, (2) these features are generally resilient to various learning algorithms, (3) linguistic and structural traits are less discriminative than temporal ones. By examining the hierarchical propagation network architectures, this approach also offers the ability to learn if a person would propagate fraudulent news or not.

Geometric deep learning is a novel class of deep learning techniques created to operate on graph-structured data [107]. Monti *et al.* [108] suggested a learning model for false news based on propagation technique. The underlying techniques enable the merging of heterogeneous data including content, user profile and activity, social graph, and news propagation by generalizing traditional convolutional neural networks to graphs. The model was developed and tested using news articles that had been circulated on Twitter and had been fact-checked by reputable organizations.

Convolution layers that extract unlabeled features and LSTM layers that capture long-term dependencies between the sequences make up the hybrid CNN-LSTM model that has been presented by Drif *et al.* [109]. With the goal of improving predictions, this deep architecture learns a regulatory grammar. Despite the intriguing results, the performance of false news identification can be further enhanced by using all the other metadata (statement, author, title, and subject). To conduct the fake news identification, Belhakimi and Drif [110] have incorporated various metadata (text, author, and title). They implemented a Word2Vec-based word embedding algorithm, and combined two convolutional neural networks (CNNs). The model has therefore achieved great accuracy using the text and author inputs.

A labeled dataset comprising 7,000 social media posts, Persian data, and articles of true and misleading news has been compiled and published by Parvizimosaed *et al.* [111]. Fake news in Covid 19 has also been found in Hindi, Chinese, Arabic, and English. On the labeled dataset, they performed a multi-label task (real vs. made-up) and compared the results of six machine learning baselines: logistic regression, support vector machine, decision tree, naive bayes, k-nearest neighbors, and random forest. Additionally, Lee's contribution [112] consists of using PCA visualization and linear assessment, TF-IDF, Word2Vec, GloVe, and BERT analyze representation spaces, then using of CAM (Class Activation Mapping) [113] for finding class-specific patterns. Finally straightforward BERT-based architecture is utilized for classification.

3.4.3 Approaches Based on Semi-supervised Learning

By including a confidence network layer, Xin Li and al [114] created a self-learning semi-supervised deep learning network that could automatically return and add accurate findings to assist the neural network in accumulating positive sample cases and so increase accuracy. To cluster similar news, Suben *et al.* [115] used a semi-supervised approach. They then labeled the centroid instance to obtain the best labels. They used five percent of this centroid instance to train a classifier, which resulted in accuracy that was just 5% poorer than utilizing all examples.

Frick *et al.* [116] have suggested a novel method for identifying tweets check-worthy that merit further investigation. Utilizing ensemble learning, it blends supervised and semi-supervised learning by utilizing cutting-edge transformer models like BERT and Bertweet. Cross-SEAN, a cross-stitch based semi-supervised end-to-end neural attention model that takes advantage of the significant amount of unlabeled data, was proposed by Paka *et al.* [117]. This model learns from pertinent outside knowledge, which allows it to generalize to newly arriving bogus news to some extent. Meel *et al.* [118] displayed a novel Convolutional Neural Organize semi-supervised system based on the self-ensembling thought to utilize the linguistic and stylometric data of clarified news articles whereas too investigating hidden patterns in unlabeled information. These analysts moreover proposed another technique [119] that employs a semi-supervised learning approach when managing with little sums of labeled information. They create a GCN-based semi-supervised fake news identifying strategy (Graph Convolutional Networks). The recommended engineering is made up of three key components: gathering word embeddings from news articles in datasets utilizing GloVe; making a similarity graph utilizing Word Mover's Remove (WMD); and at long last utilizing Graph Convolutional Networks (GCN) to classify news articles into two categories in a semi-supervised way.

A brand-new early semi-supervised detection model called ENDEMIC was introduced by Bnasal *et al.* [120]. To gather information on propagation, they constructed a heterogeneous graph comprising follower-followee, user-tweet, and tweet-retweet connections. While time-relative web scraped data is an example of an exogenous signal, graph embeddings and contextual features are examples of endogenous signals. Also, Mansouri *et al.* [121] developed a hybrid approach for spotting counterfeit news that combines convolutional neural networks and semi-supervised linear discriminant analysis. This method begins by utilizing CNN to extract numerous features from text and image data. In order to determine the classes of unclassified data, linear discrimination analysis (LDA) is then performed. An additional semi-supervised learning algorithm was developed by Konkobo *et al.* [122] to quickly identify bogus news on social media. They created a model to extract users' opinions from comments, using the CredRank Algorithm to determine the credibility of the users, and then created a tiny network of people involved in the dissemination of a particular news story. The outputs from these three processes are sent into SSLNews (Semi supervised News classifier), a news classifier. The three networks that make up SSLNews are a shared CNN, an unsupervised CNN, and a supervised CNN.

We categorize and compare the previous methods discussed based on the features of the fake information (see Table 3.1). Table 3.1 illustrates that the majority of fake news detection algorithms are feature-based, in that they rely on developing efficient features that individually or jointly are able to distinguish between real and fake information. The linguistic feature are popular and widely used on fake news detection methods in contrast to the visual and friendship-network characteristics, which are somewhat rare in regard to the other features.

3.5 Social Bots detection

Bots or what are known the internet robots, which is a programs that have been created for specialized missions like ”chatbot”. They are used by advertisers, marketing agencies, and companies to mingle with users across messaging to offer services and response to the habitual requests and questions of the customers. In other side, it exists another type of bots that are really dangerous and could cause tremendous damages. Some of its purposes: stealing the account information like password, grabbing financial data and especially proliferating fake news.

On social media particularly Twitter, the bots are very active and try to imitate the genuine user behaviour on sharing information to be exactly like him. There are different methods to reach that objective such as tweeting posts, retweeting, following other accounts, or interacting in comments, and their comportment is already developed till many years which made the distinction between them and human extremely difficult.

Nowadays, many researchers dedicate their efforts and research for the social bots detection as it became an efficient mechanism to minimize the fake news propagation.

3.5.1 Social Bots detection Methods Based on Supervised and Unsupervised learning

The bot detection methods consider usually many features and information [3, 71, 123, 124] to obtain relevant classification. Lee *et al.* [125] tested the performance of 30 classification algorithms based on different features group, and concluded that the Tree-based supervised classifiers produced the highest accuracy. Based on spammers account, Cresci *et al.* [126] recommended comparing their behavior to look for patterns amongst automated accounts in order to discover spammers unsupervisedly. They introduced a method dubbed ”Digital DNA” that was inspired by biological processes for modeling online user behavior. Another work proposed by Cresci [127] implement an evolutionary algorithms to improve social bot skills. Kosmajac *et al.* [128] has created a technique for identifying bots on Twitter by employing a user behavior fingerprint and a collection of statistical techniques defining different facets of user behavior. Although, these methods reach an accurate detection, many studies on a particular set of features are always still needed. On the other hand, Pakaya *et al.* [129] have constructed a classification model based solely on account tweets. Logistic Regression, ADA Boost, XGBoost, and Random Forest were the models employed. The suggested strategy was joining the tweets together to create a single document. The tf-idf, bigram, and Word2Vec NLP feature extraction techniques were employed. Additionally, they constructed a multi-class model based on tweet attributes to illustrate the many forms of malicious accounts (spambots and fake followers). Results indicated that the XGBoost algorithm was the most effective one.

Moreover, several approaches based on emotions analysis have been proposed. Wang *et al.* [130] studied sentiment lexicon expansion for a social media corpus. Ferrara *et al.* [131] studied several sentiments aspects such as the velocity of conversations polarity spread, the most typical types of emotions of popular conversations on social media, and the kind of feeling that is expressed in conversations.

In recent years, deep learning algorithms have generated a lot of research interest and achieved cutting-edge results in many fields of natural language processing (NLP). Socher *et al.* [33] proposed a series of recurrent neural networks (RNN) that can be used to study the compositional semantics of words and phrases of varying length. To identify bots, Kudugunta and Ferrara [132] proposed a contextual LSTM architecture allowing to use both content and metadata of tweet. Wei *et al.* [133] presented a BiLSTM model with word embedding with no handcrafted features or prior knowledge to distinguish between bots and humans accounts.

Table 3.1: Comparison of features-based fake news detection methods [5].

Methods	Content level		User level		Social level	
	Linguistic	Visual	User profile	Credibility features	Diffusion network	Friendship network
Cresci *et al.* (2014) [86]					✓	✓
Jin *et al.* (2014) [87]	✓			✓	✓	
Ma *et al.* (2015) [77]	✓					
Ciampaglia *et al.* (2015) [88]	✓				✓	
Lendavi *et al.* (2016) [89]	✓					
Hardalov *et al.* (2016) [90]	✓		✓			
Ferreira *et al.* (2016) [91]	✓					
Joulin *et al.* (2017) [53]	✓					
Wang (2017) [94]	✓					
Ruchansky *et al.* (2017) [96]	✓		✓			
Long *et al.* (2017) [97]	✓		✓			
Volkova *et al.* (2017) [98]	✓			✓		
Jin *et al.* (2017) [74]		✓		✓		
Hosseinimotlagh *et al.* (2018) [79]	✓					
Deka *et al.* (2022) [82]	✓					
Pick *et al.* (2022) [83]	✓					
Silva *et al.* (2021) [84]	✓				✓	
Li *et al.* (2021) [85]	✓	✓	✓	✓		
Wang *et al.* (2018) [99]	✓	✓				
Liu *et al.* (2018) [100]			✓	✓	✓	
Shrestha *et al.* (2018) [101]	✓		✓	✓	✓	
Qian *et al.* (2018) [102]	✓		✓			
Tschiatschek *et al.* (2018) [103]					✓	
Yang *et al.* (2019) [80]				✓	✓	
Shu *et al.* (2019) [104]	✓		✓	✓	✓	✓
Monti *et al.* (2019) [108]	✓		✓	✓	✓	✓
Drif *et al.* (2019) [109]	✓					
Belhakimi *et al.* (2019) [110]	✓		✓			
Parvizimosaed *et al.* (2022) [111]	✓					
Lee *et al.* (2022) [112]	✓					
Xil Li *et al.* (2022) [114]	✓					
Suben *et al.* (2022) [115]	✓					
Frick *et al.* (2022) [116]	✓					
Paka *et al.* (2021) [117]	✓					
Meel *et al.* (2021) [118]	✓					
Meel *et al.* (2021) [119]	✓					
Bansal *et al.* (2021) [120]	✓					✓
Mansouri *et al.* (2020) [121]	✓	✓				
Konkobo *et al.* (2020) [122]	✓		✓	✓		

3.5.2 Social Bots detection Methods Based on Semi-Supervised learning

In addition to works based on supervised and unsupervised approaches, various studies are based on semi-supervised method to identify social bots. Zhao *et al.* [134] proposed a semi-supervised model founded on an attention mechanism-based graph CNN, which spots spam bots by integrating many user characteristics and relational structures. To detect counterfeit accounts from a vast volume of Twitter data, BalaAnand *et al.* [135] presented an Enhanced Graph-based Semi-supervised Learning Algorithm (EGSLA). Another work of Shaabani *et al.* [136] presents a semi-supervised self-training architecture capable of capturing Pathogenic Social Media users. To identify single and batches of spam accounts, Alharthy *et al.* [137] used two semi-supervised techniques plus a set of specified features. A recent work of Guo [138] symmetrically involved BERT and GCN (Graph Convolutional Network), and a new architecture for bot identification that merged large-scale pre-training and transductive learning was proposed.

Numerous studies have considered the bot detection problem as a binary classification. However, only binary classifiers will be capable to differentiate bots and genuine users when bots are of the identical category as the ones used when training the model. To detect the bots, Rodriguez *et al.* [139] used a one-class classification strategy. This strategy has the advantage of not necessitating examples of anomalous activity. When the goal is to detect deviations from predicted behavior, one-class categorization is usually applied. The researchers select the account features (retweet, replies, inter-time, number of listed tweets, and friends-to-follower ratio) and illustrated that the one-class classifier distinguishes the bots and the legitimate users consistently. Eventually, the previous semi-supervised techniques are summarized and briefly compared in Table 3.2

3.6 Conclusion

The field of information dissemination has undergone a significant amount of development in the modern world., where countless numbers of individuals can instantaneously submit and receive unrestricted news and information. In this chapter, we studied fake news, its main sources, and the detection methods on social media in order to come up with new insights and novel models. More concretely, We first give a thorough overview of the supervised and unsupervised techniques for detecting fake news, and then describe the feature engineering employed while extracting features from fraudulent news. Finally, we shed the light on the social bots detection approaches as this research direction is flourishing and helps to reduce the dissemination of false information through online social media.

Table 3.2: Brief description of prior surveyed semi-supervised methodes for bot detection.

Methode	**Dataset used**	**Features selected**	**Accuracy**	**Precision**	**Recall**	**F1-score**
Zhao *et al.* [134]	Twitter 1KS-10KN dataset	user and network features	_	0.93	0.88	0.91
Guo [138]	cresci-rtbust, botometer-feedbak, gilani, cresci-stock-2018 and midterm dataset	tweet text	0.9026 (The best result achieved on midterm dataset)	0.8842 (The best result achieved on cresci-rtbust dataset)	0.7884 (The best result achieved on midterm dataset)	0.8089 (The best result achieved on midterm dataset)
BalaAnand *et al.* [135]	automated data collection by Python web-scraping	Fraction of retweets, Standard tweet length, Fraction of URLs, Average time between tweets	0.903	0.923	0.908	_
Rodriguez *et al.* [139]	Cresci-2017 dataset	account features	0.921	_	_	_
Shaabani *et al.* [136]	ISIS dataset	_	0.82	0.90	_	_
Alharthy *et al.* [137]	automated data collection by Twitter API	Tweet meta-data and Account metadata	0.91	0.88	_	_

Chapter 4

Sentiment Analysis-based BiLSTM Model for Bot Detection on Social Media

*****

4.1 Introduction

Twitter is a social network based on micro-blogging, where its users interact by posting tweets, which are texts of no more than 280 characters. This social tool has certain features such as mentions, hashtags, URL shorteners...etc. It also allows users to mention other users in their tweets, reply to an infinite number of messages, or retweet them. It also provides various ways to access its services, such as mobile applications and web pages. Twitter always represents an ideal target for exploitation by special automated programs known as bots, and this is due to the large number of its users as well as its open nature. These bots were created from special algorithms to automatically produce content and simulate human behavior while interacting with them, such as expressing feelings and the duration of activity in social networks. The creation of this type of bot is not always beneficial but can be very harmful when it comes to stealing personal information and spreading fraudulent information [71, 133, 139].

Most existing works identify bots through a multi-feature approach, including content features [140], features on the profile, user behavior, friendship networks, and the timeline of an account [125, 141–146]. In this chapter, we propose a Bidirectional long short-term memory(BiLSTM) model that integrates linguistic cues of Twitter text with sentiment analysis metadata (polarity, subjectivity, happy emoticons, sad emoticons and the interjections). Building on the BiLSTM prediction, the proposed architecture depends not only on the previous input but also on the future input.

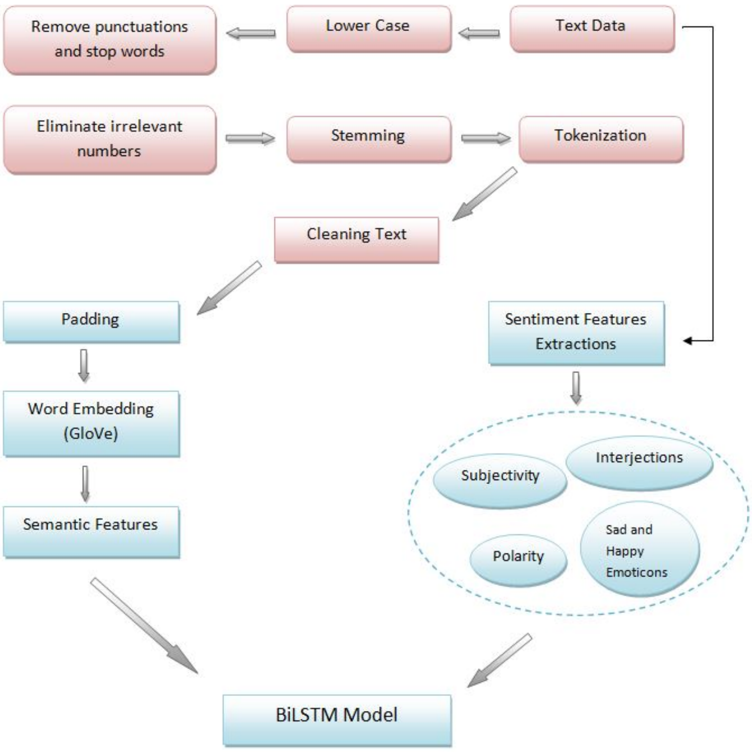


Figure 4.1: The proposed model based on semantic and sentiment features.

4.2 The proposed methodology

Based on BiLSTM (Bidirectional long short-term memory) architectures and a set of sentiment features we built our model.

The proposed deep learning model process workflow is shown in figure 4.1. First of all, we apply feature extraction techniques to identify the tweet polarity, subjectivity, interjections, and emoticon types, with a focus on understanding which features are important for this task. It is a time-consuming process, and not all the features provide the same amount of useful information. In order to extract the semantic features from the text, we apply the pre-processing phase, we remove the punctuation and the stop words to avoid feature noise in the classification process. We also eliminate the irrelevant numbers and carry out stemming and tokenization. After that, to accurately convert our words into a vector space where the same words are represented by the same vectors, we use the Word Embeddings technique. The final phase consists of exploiting both semantic features and sentiment features for the BiLSTM implementation.

4.2.1 Features extraction

To identify tweet-based bots, which is one of the common problems in language processing, recurrent neural networks are used to extract a certain type of quality input that is structured data, which necessarily needs a scalar vector to represent it [147]. For that, to transform the text of a tweet into a vector suitable for Long Short Term Memory (LSTM), we used as embedding a pre-trained set of GlobalVectors for word representation (GloVE) that uses the pre-trained word embeddings to allocate the vocabulary and create an embedding matrix. It enables the semantic and syntactic significance of a particular sentence to be captured. In fact, a predetermined vector space is used to represent each word as real-valued vectors.

The key aspect of sentiment analysis is to analyze a body of text to understand the opinion expressed by it. Typically, we quantify this sentiment with a positive or negative value, called polarity. The overall sentiment is often inferred as positive, neutral, or negative from the sign of the polarity score. Moreover, sentiment analysis typically performs better on text that contains subjective content rather than purely objective content. Objective text primarily focuses on presenting factual information, descriptions, or statements without any emotional or subjective language. Sentiment analysis algorithms rely on identifying patterns in language, including emotional cues, polarity, and subjective expressions, to determine the sentiment or emotional tone conveyed in the text. Subjective text, on the other hand, often involves language that reflects human emotions, opinions, attitudes, and feelings. This type of text contains subjective elements such as sentiment-laden words, expressions of emotions, personal viewpoints, or experiences. In our work, for the sentiment analysis phase, we focus on both the polarity and subjectivity of the tweets and the number of interjections. Polarity is defined as an output that lies between $[-1, 1]$, where -1 refers to negative sentiments and $+1$ refers to positive sentiments. Subjectivity is the output that lies within $[0, 1]$ and refers to personal opinions and judgments.

4.2.2 Model construction

We use TensorFlow [148]. Therefore, the deep learning model building is based on the following architecture:

First, the maximum number of words in one sentence is set at 50, where short sentences are padded to get the same length. Each word in the sentence is represented by a vector of length 300, so that the set of these vectors gives a matrix of size 50×300, which is what is known as embedding, that represents the first layer in the model, and so on semantic features. To connect these features with the sentiment features, we use a flatten layer that produces a single vector that we pass to a RepeatVector layer to produce a 3D output. This last will be the next input for a single BiLSTM layer of 10 cells, which can effectively understand the context

TABLE 4.1: Model Summary.

Layer	Output shape	Connected to (the layer before)
Input 1	(50)	
Embedding	(50,300)	Input 1
Flatten	(15000)	Embedding
Input 2	(5)	
Concatenate	(15005)	Flatten Input 2
RepeatVector	(1,15005)	Concatenate
BiLSTM	(20)	RepeatVector
Fully connected 1	(64)	BiLSTM
Fully connected 2	(1)	Fully connected 1

better due to its unique sequences analysis from forwards to backwards (past to future) and vice versa (future to past). This technique gives an effective possibility to “look forward” in the sentence to see if “future” tokens may influence the current decision by preserving information from the inputs using the hidden states. Eventually, our classification is binary so we use a first fully connected layer whith 64 neurons where the BiLSTM layer output is passed into it, then the second one based on sigmoid function [149] which produced a single output ranged between 0 and 1. Table 4.1 describes the connection between layers and the output shape of each one.

4.3 Experiments and Results

In this section, we discuss our experimental setup which contains three parts: a description of the dataset we have used to test the proposed classifier performance; our analysis of the extracted features that reveal the behavior of legitimate and bot accounts; and the various classifiers we implemented to measure the model accuracy.

4.3.1 Dataset

By utilizing the publicly available annotated dataset cresci-2017 [126], we assess our suggested models. 8,4 million tweets were sent from 3,474 human accounts in this dataset, while 3 million tweets were sent by 1,455 bot accounts. Table 4.2 reports the statistics of the dataset. According to Cresci *et al.* [127] A representative sample of genuine accounts is the real accounts (operated by humans). During the 2014 Rome mayoral election, Twitter’s social spambots 1 dataset was indexed. The #TALNTS hashtag (concerns a mobile phone application) was promoted by a gang of bots for several months under the name ”Spambots

Table 4.2: Cresci 2017 dataset statistics [150].

	Users	Tweets
Genuine	3,474	8,377,522
Spambots #1	991	1,610,176
Spambots #2	3,457	428,542
Spambots #3	464	1,418,626
Total	8,386	11,834,866

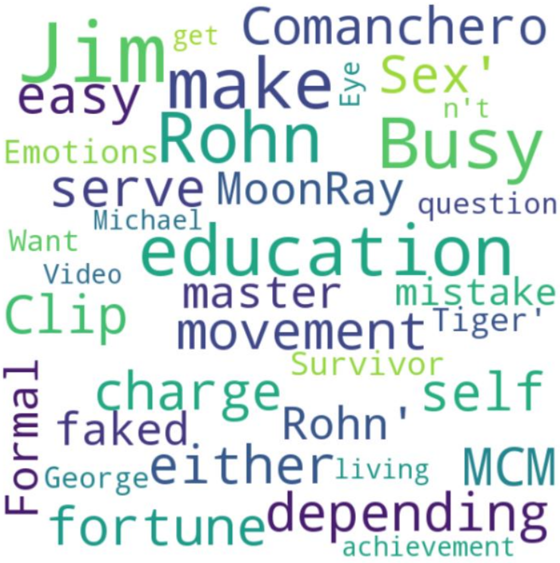


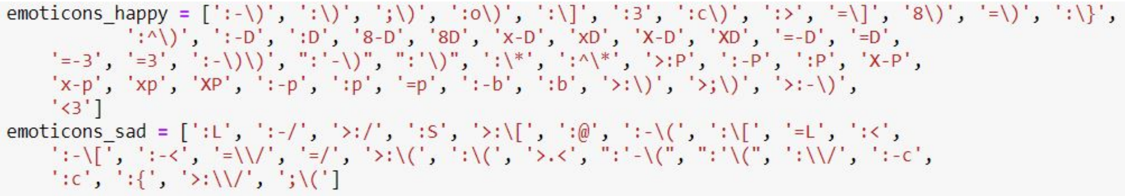
Figure 4.2: Social bots word cloud.

2 dataset", Social-bot-3, however, is about spammers that advertise things for sale on Amazon.com.

The data is cleaned, prepared, and investigated in the first phase for better performance. Figure 4.2 shows the bot's word cloud that we have extracted during the preprocessing phase.

Then, we proceed to a sentiment analysis phase. So, we focus on both the polarity and subjectivity of the tweets and the number of interjections. In addition, we extract the different emoticons that express happiness and sadness (see Fig. 4.3).

Our analysis results are shown in figure 4.4. As we can see, both humans and bots share



```
emoticons_happy = [':-\)', ':\)', ';\)', ':o\)', ':\]', ':3', ':c\)', ':>', '=\]', '8\)', '=\)', ':\}',
            ':^\)', ':-D', ':D', '8-D', '8D', 'x-D', 'xD', 'X-D', 'XD', '=-D', '=D',
    '=-3', '=3', ':-\)\)', ":'-\)", ":'\)", ':\*', ':^\*', '>:P', ':-P', ':P', 'X-P',
    'x-p', 'xp', 'XP', ':-p', ':p', '=p', ':-b', ':b', '>:\)', '>;\)', '>:-\)',
    '<3']
emoticons_sad = [':L', ':-/', '>:/', ':S', '>:\[', ':@', ':-\(', ':\[', '=L', ':<',
    ':-\[', ':-<', '=\\/', '=/', '>:\(', ':\(', '>.<', ":'-\(", ":'\(", ':\\/', ':-c',
    ':c', ':{', '>:\\/', ';\(']
```

Figure 4.3: Extracting the happy and sad emoticons.

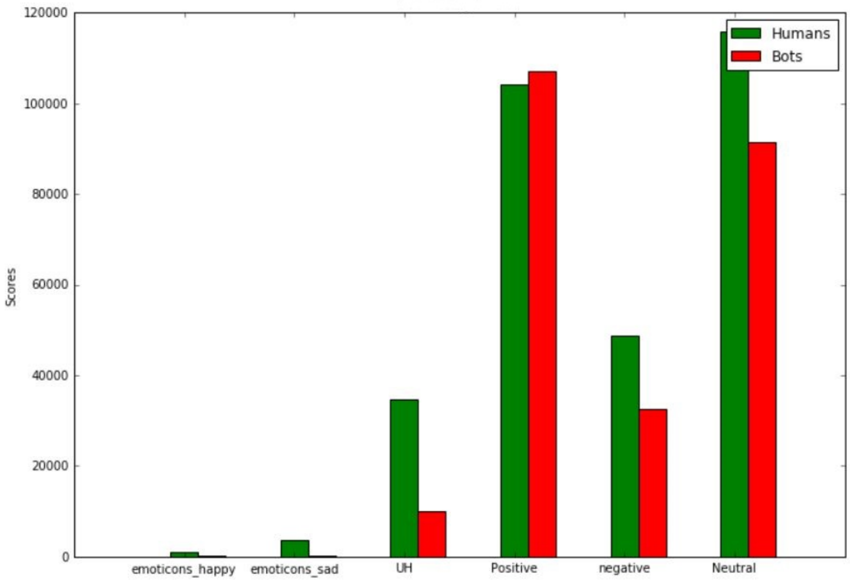


FIGURE 4.4: Sentimental Features Analysis Scores by Label (human and bot).

various sentiments. The human reached high scores in all sentiment features with no remarkable difference in terms of positive sentiments. The sentiment gap between humans and bots is more evident for humans, especially for neutral scores and interjection expressions. Human interjection conveys an emotion such as surprise, excitement, happiness, or anger. In other words, humans are capable of exhibiting significantly more complex statements that express emotion.

In the final phase, the tokenization technique was applied. With a vocabulary size of 100000, sequences were produced utilizing padding to produce sequences of identical size. After that, we implement word embedding to capture the semantic and contextual features.

To more accurately represent the text in dimensional space, we employ pre-trained GloVe word vectors in this study.

Baselines: We compared our model's performance with the following state-of-art methods (see Chapter 2 for more details):

- Cresci *et al.* [126] introduced a method dubbed ”Digital DNA” that was inspired by biological processes for modeling online user behavior. This method involves the extraction and analysis of various behavioral features exhibited by users during their interactions online. Leveraging machine learning algorithms and computational models, this approach identifies unique behavioral patterns and adapts to evolving behaviors over time.
- Kosmajac *et al.* [128] suggested a method to detect bots on Twitter by using a fingerprint of user behavior. The authors leverage diverse metrics and measures within the Twitter

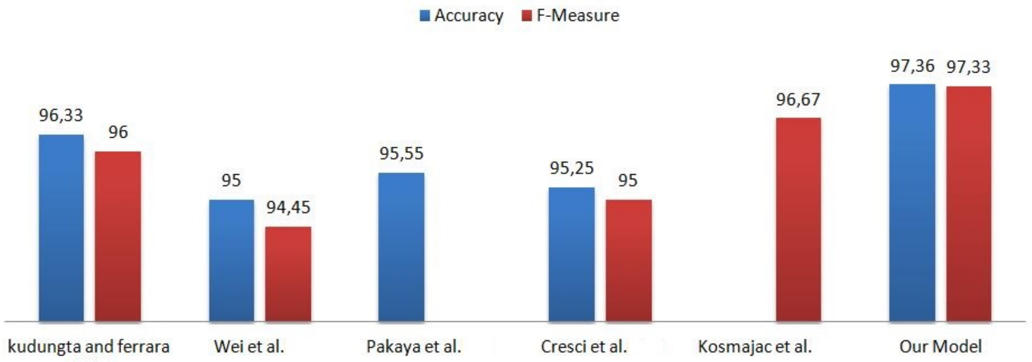


FIGURE 4.5: Performance Comparison [6].

ecosystem based on quantifying the variability and distinctiveness of bot-generated content compared to human-generated content. By analyzing factors such as posting frequency, language patterns, engagement dynamics, and content types, this method aims to detect anomalies in bot behavior.

- Pakaya *et al.* [129] have leveraged diverse account attributes such as posting behavior, temporal patterns, linguistic cues, engagement metrics, and account metadata, this method focuses on developing a classification model grounded on account tweets using Logistic Regression, ADA Boost, XGBoost, and Random Forest.
- Wei *et al.* [133] presented a BiLSTM model with word embedding with no handcrafted features or prior knowledge to distinguish between bots and human accounts. This work extracts the temporal sequence and semantic context of tweets and implements BiLSTM networks.
- Kudugunta et Ferrara [132] proposed a contextual LSTM architecture analyzing diverse attributes such as posting patterns, content semantics, temporal dynamics, and network interactions.

We have used a test-train split such that the dataset is separated into 20% for testing and 80% for training to experiment. We set the epochs to 40 training epochs. Our evaluation measures of choice were accuracy and F-measure. Cresci *et al.* [126] and Wei *et al.* [133] divided the dataset into two testing sets. The first mixed group, which included 50% social-bot-1 and 50% human accounts, and the second, which included 50% social-bot-3 and 50% human accounts, in this case, we calculate the average of their performance in the two sets. We outline our empirical results in Figure 4.5.

As shown in figure 4.5, our model outperforms the other models on several metrics. Where, our proposed BiLSTM model shows an accurate prediction expressed by 97.36% of accuracy and 97.33% of F-measure, which indicates a significant prediction. The deep bidirectional recurrent neural network armature's effective modeling ability with word embedding and emotive properties produced these encouraging experimental findings.

However, combining semantic and sentiment features could slightly boost the performance which is not very significant. It is clear that sentiment features have a positive influence on the results of our Bi-LSTM model and have ruled out the possibility of increasing the detection performance. This leads us to deduce that taking into consideration textual signs and the writing style can probably improve bot detection. Unlike the majority of works on bot detection, which has integrated a larger set of features without considering interesting linguistics, we will explore and dig deeply into the effect of accounting only on linguistic features for bot detection as language is one of the most complex human faculties.

4.4 Conclusion

In this chapter, we develop a Bi-LSTM approach where the neural network accepts text and sentiment data, which usually require being processed separately. Our approach gives insights into the strength of incorporating sentiment features for bot detection. As a result, the model has reached a significant prediction with 97.36% of accuracy. In the next chapter and according to the discussion we provide about the various model results, we aim to study the impact of more linguistic features on social bot detection.

Chapter 5

A Hybrid Mixing Engineered Linguistic Features Framework Based on Autoencoder for Social Bot Detection

*****

5.1 Introduction

Nowadays, many users on social media are performing various acts that can produce incorrect information that propagates easily through the internet for different purposes [151]. Some of this information tries to deceive the reader or sway his perspective on a topic. Others are created from scratch with a tempting caption to enhance website traffic and visits. Recently, there have been several works studying fake news features analysis [5, 60, 110, 152–154]. Supervising and investigating the diffusion's information sources and the nature of the users is a challenging task. These sources can be nonhuman (social bots and cyborg use). Therefore, we focus in this work on social bot detection as it has become an efficient mechanism for fake news propagation that can hurt individuals and society.

The main objective of this chapter is to recognize the social bots, particularly through their writing style. Firstly, we will explain the proposed methodology. Secondly, we will describe our proposed approach for social bot detection. Therefore, we will discuss the results.

5.2 The proposed Methodology

This research aims to design a linguistically oriented framework that combines the embedding-based strength with the advantage of the Autoencoder (AEs) and its ability to represent features in latent space. The workflow of our methodology is organized as follows: 1) Studying the feature extraction and the most prominent measures for natural language texts. 2) Explaining the proposed semi-supervised linguistic framework for building a high-performance bot detection model. 3) Conducting experiments on real-world datasets and discussing the results.

5.2.1 The features extraction study

First of all, we will examine the text content in the first part of this project to delineate bot behavior, as the language and phrase composition of bots and genuine people may differ. Although the techniques of Natural Language Processing have a redoubtable function in ensuring that bots grasp the language and are more human-like, it seems that the bots are surrounded by dissension due to their restrictions to communicate with people who speak the same language [142]. To find insights into this issue, we focus on three NLP steps:

- The first step is lexical analysis. The batch of sentences and words is a language lexicon. We will first analyze the text and separate it into sentences and words. Every word and punctuation mark is a separate unit.

- The second phase is syntactic analysis. We will explore the grammatical role of every word in a sentence by tagging each of it to indicate what type of token it is, for example, is a verb (in past, present, or future tense), a pronoun, an article, a stop word, adjective ..., and identifies the words relationship.

- In the third step, we perform the semantic analysis. To do this, we have to deploy the Word embedding techniques that mainly take words or phrases from the vocabulary to map them to real number vectors.

Machine learning algorithms can be used in these NLP phases to dynamically learn the rules by exploring a corpus. We start our approach by extracting features based on the previous three main analysis levels. The first process phase is the lexical analysis:

1. Divide tweets into sentences and words.

2. Elicit emojis, hashtags, both happy and sad emoticons.

3. Identify upper letters, numeric, and blank spaces.

4. Then, we calculate all these feature numbers, besides determining the whole number of characters and the average word in both human and bot tweets.

The next step is to analyze the tweet words in terms of syntax where an set of syntactic features was extracted:

1. The frequency of punctuations (commas, question mark and exclamation marks).
2. The frequency of stop words and URLs.
3. We also focus on identifying the grammatical role of each word in a sentence via speech tagging.

For the semantic approach, we realize it based on the GloVe embedding technique.

For the Features Extraction Based on Lexical Diversity, we have selected the ”Lexical Diversity” (LD). It is a key linguistic feature that aims at indicating how it is complex and difficult to read a text. The type-token ratio (TTR) is one of the methods for measuring LD [155, 156]. It is just the proportion of types (unique words) to the total number of words in a certain text [157].

$$TTR = V/N \tag{5.1}$$

V: Number of unique words in the text (types).
N: Number of total words in the text (tokens).

We take this sentence as an example: ”To live with untreated PTSD is to feel like you might die any moment. Again and again. Help costs money.”. There are 18 unique words (to, live, with, untreated, PTSD, is, feel, like, you, might, die, any, moment, again, and, help, costs, money) in this sentence which is composed of 20 words.

TTR: the number of different words (types)/all words produced (tokens). In this example, the TTR is 0.90 (i.e. 18/20).

But numerous research [158, 159] have demonstrated that text length significantly affects TTR. Because the amount of lexical items that can be activated at any given time is thought to be finite, it is less likely that a speaker will utter new words as the sample length grows [160].

The Measure of Textual Lexical Diversity (MTLD) [161, 162] separates the text into sections or components. Because the fragmentation is created based on the TTR values of the segments, these components can vary in length. When a segment reaches the 0.72 threshold known as the ”default TTR size value,” it is said to have ended. The mean of all the TTRs is eventually calculated. Because all of the variables reach the TTR’s stability point, this measurement appears to be accurate.

According to [162], the stabilization point is the point at which neither the introduction of repeated types nor even a significant number of new types can significantly change the TTR trajectory. The average of all the factors’ TTRs ought to provide a reliable and valid result independent of text length because the factors do not consist of a fixed amount of tokens and always reach the stabilization point individually. The total number of tokens (N) divided by the number of factors yields the MTLD’s final result [157].

$$MTLD = N/factors \qquad \text{(segments with the stabilization point of TTR)} \tag{5.2}$$

TABLE 5.1: MTLD Calculation Example.

	to	**live**	**with**	**untreated**	**PTSD**	**is**	**to**	**feel**	**like**	**you**
Type	1	2	3	4	5	6	6	7	8	9
Token	1	2	3	4	5	6	7	8	9	10
TTR	1	1	1	1	1	1	0.85	0.87	088	0.9
	might	**die**	**any**	**moment**	**again**	**and**	**again**	**help**	**costs**	**money**
Type	10	11	12	13	14	15	15	16	17	18
Token	11	12	13	14	15	16	17	18	19	20
TTR	0.90	0.91	0.92	0.923	0.93	0.937	0.88	0.888	0.89	0.9

After that, the entire language sample's text is inverted to estimate a new MTLD score. To calculate the final MTLD estimate, the forward and backward MTLD scores are averaged.

The MTLD value of the previous example is calculated in this way [161, 163]: From the text's beginning until it finishes, Count(x) the number of times the TTR is 0.72 or less. In Table 5.1, we compute the value of x by increasing the words successively to create a segment. Also, we calculate the TTR of each word: if the $TTR <=$ 0.72, then Count(x)=Count(x) +1. Notice that a segment can be created if and only if the length of its tokens is greater or equal to 10 tokens. As we can see from Table 5.1, the last value of 0.9 reaches 35% of the trajectory from 1.00 to 0.72 (i.e. [1.00-0.9]/[1.00-0.72] = 0.1/0.28). The segment that does not reach the threshold of 0.72 is taken into consideration to enhance the reliability of MTLD. Therefore, the rate of the trajectory is added to the count(x) and the mean number of words required is 57.14 (i.e. 20/0.35). Similarly, the calculation is made backward from the last word. The two values derived from the forward calculation and the backward one are averaged (in this case, we obtain MTLD= 57.14).

The Moving-Average TTR (MATTR) algorithm [164, 165] selects a window length of $x\prime$ tokens, and then, it computes TTR for tokens 1 to $x\prime$, 2 to ($x\prime$ + 1), 3 to ($x\prime$ + 2), ..., and so on for the entire sample. As a result, the final MATTR outcome is determined by averaging each individual TTR [166].

In the machine learning model training phase, the data characteristics have a big influence on its attainments. A bad choice of these features can injuriously influence model performance and decrease accuracy. There are many advantages of applying feature selection before shaping the data such as reducing overfitting, improving accuracy, reducing algorithm complexity, and algorithms' training faster.

For selecting features task, we used Extremely Randomized Trees Classifier (Extra Trees Classifier) [167] which is an updated Tree-Based Classifier that extracts the most relevant features. It attaches a score for every feature; if this score is high it indicates that this feature is pertinent for the model performance.

5.2.2 Mixing Engineered Linguistic features framework based on Autoencoders (Hybrid-MELAu)

We introduce Hybrid-MELAu: a novel semi-supervised deep framework oriented mixing engineered linguistic features based on autoencoder to improve the Twitter bot detection performance. Thus, we implement a feature extractor based on DALS (Deep Dense Autoencoder based on Lexical and Syntactic features) and GloVe-BiLSTM autoencoder (GloVe Word Embedding Bidirectional-LSTM Autoencoder) to learn better latent representations of the human linguistic behaviors. Also, there is a growing interest in bot detection to utilize one-class classifiers based solely on examples from a single class to learn its representations and determine whether a new example belongs to that class or not.
Therefore, the proposed architecture includes two parts: The feature learner, which relies on two autoencoders components [168–170] that pre-train the layers of the model. The feature learner associates the extracted lexical and syntactic features with their corresponding semantic features using the transfer learning technique. It consists in building the latent spaces from the pre-trained encoder part of the two autoencoders. These latent spaces present the most robust representation of the dataset. Whereas, we need to hold the concatenated encoders of the two autoencoders and fix their weights, to gain the advantage of their experience. Hence, the weight values of the encoders are frozen while we learn feed-forward deep learning network weights, following the architecture illustrated in Figure 5.1. Then the second part is the classification model, where we chain the features, that have been extracted, with several deep neural network classifiers.

Here, we will explain the Features learner and how it works. To extract features, we used one of the well-known deep representation learning algorithms; Autoencoders. The proposed approach comprises two different autoencoders: the first one is based on a deep-stacked autoencoder that reconstructs the input features, and the second one is a sequence-to-sequence LSTM autoencoder that learns vector representations of any unstructured text.
The first autoencoder DALS is composed of six layers to input the lexical and syntactic features. Its architecture is provided in Figure 5.2. The first three layers set up the encoder with 15, 10, and 6 neurons respectively, while the third layer is the latent space. The last three layers perform the decoder such as the initial two layers have 10 and 15 neurons successively, and the last layer is the output layer (it outputs the same neurons numbers as the inputs). The training procedure of this architecture is summarized in Algorithm 1.

At the semantic level, the sequence prediction remains a complex issue, not only because the input sequence length can vary, but the notes temporal scheduling can make it difficult to extract the appropriate features for use as input to supervised learning models. To capture the

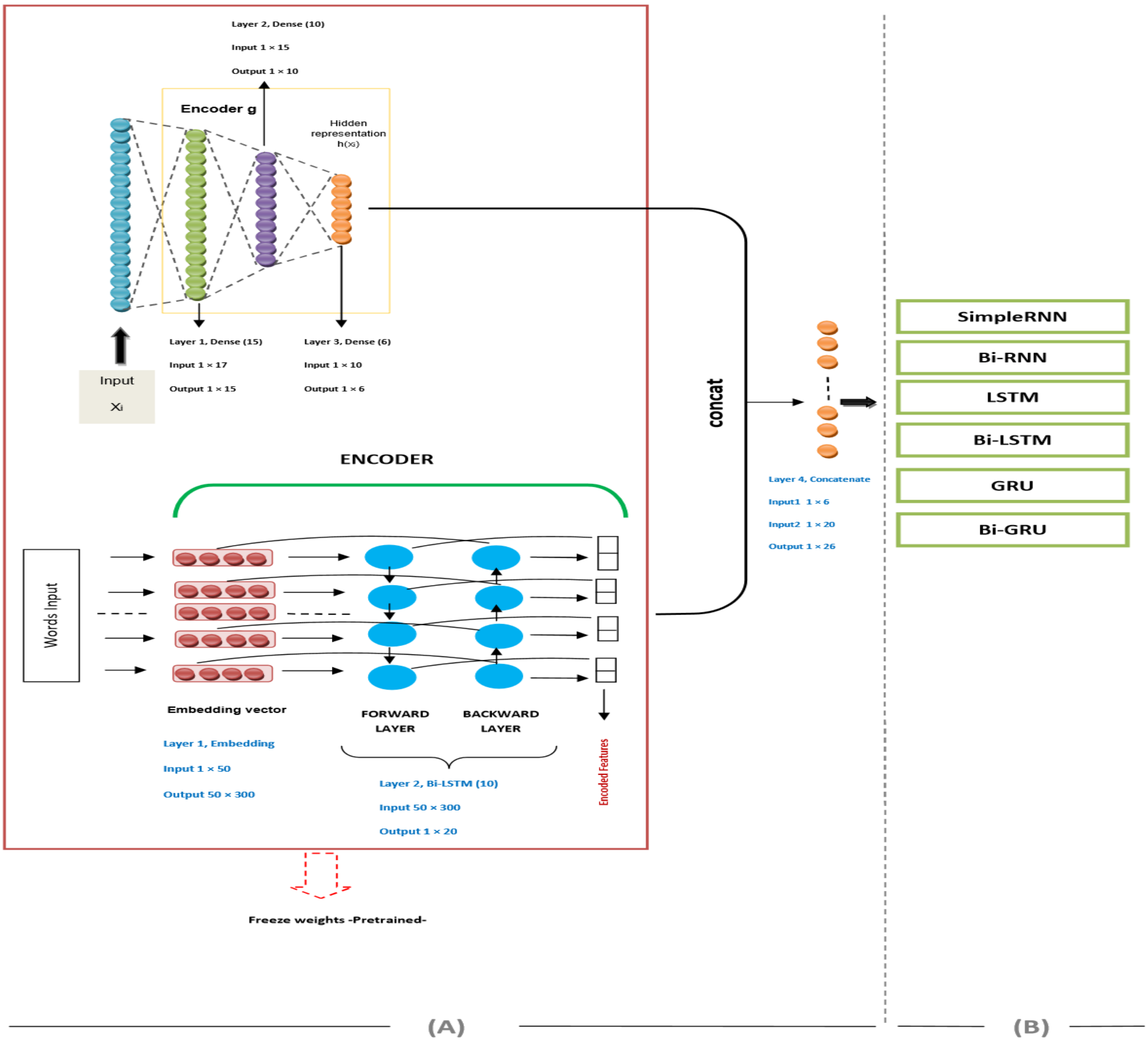


Figure 5.1: The Hybrid-MELAu for Twitter bots detection. Part (A) shows the freezing weights process for the two pre-trained encoders parts form DALS and Glove-BiLSTM autoencoder and their concatenation. Part (B) exhibits the classifiers.

temporal structure, we develop a GloVe-BiLSTM autoencoder model. In another word, the encoder part of the model can be used to compress tweets text that in turn may be used as a feature vector input to a supervised learning model.

For a better understanding, let's visualize the architecture in Figure 5.3. This figure shows the tweets flow across the GloVe-BiLSTM autoencoder network layers for one sample of data. The encoder is accountable for the source tweet reading and encoding it to an inner representation by capturing the meaning of these tweets. A simple model creation includes an embedding input ensued by a Bidirectional-LSTM hidden layer that generates a fixed-length representation. First, we input the tweet texts to the embedding layer, where each word is transformed into a distributed representation [171]. This layer is a matrix of size $m \times v$, where v is the vector length and it is equal to 300, in which we learned word embeddings

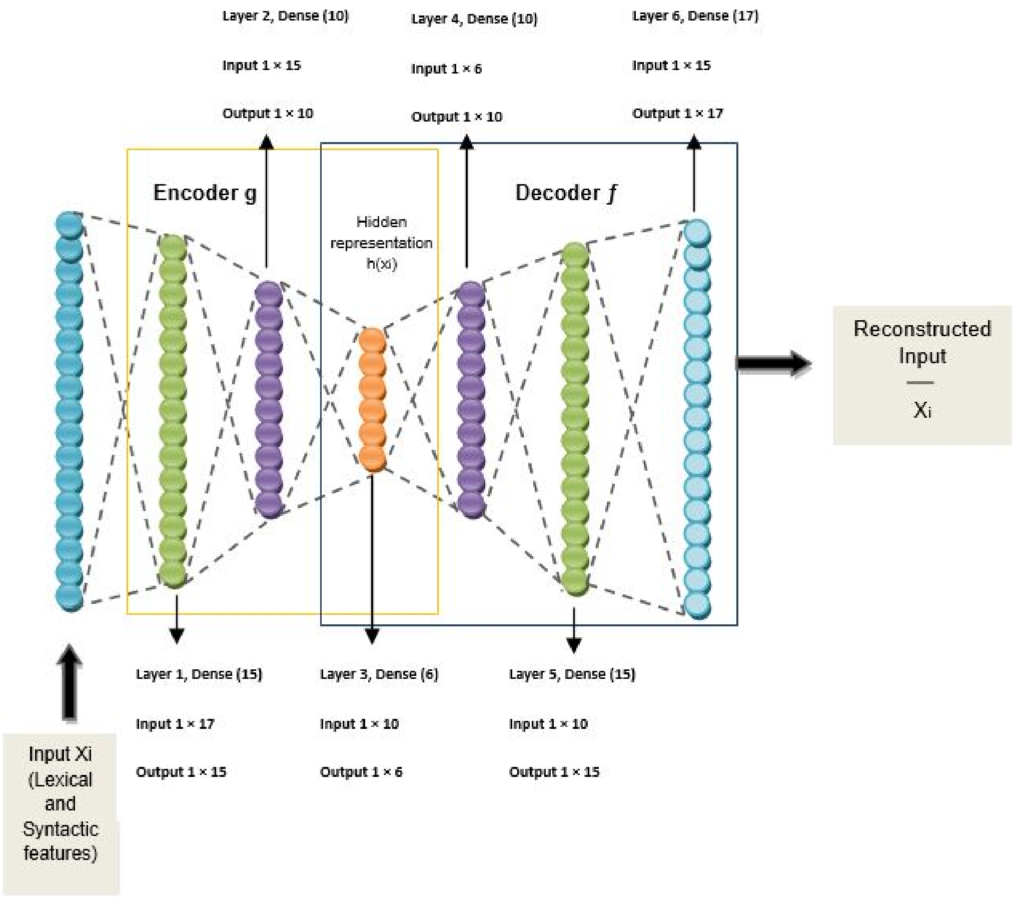


FIGURE 5.2: The feature extractor's architecture : Deep Dense Autoencoder based on Lexical and Syntactic features (DALS).

from text using a pre-trained 300-dimensional Google News Vectors approach (GloVe) [50], and m is the tokens number in the tweets which is fixed on 50.

The Bidirectional-LSTM layer [48] used the hidden states to maintain the inputs information and fed it in a forward way from past to future and backward from future to past. Moreover, Bidirectional LSTMs have the capacity to better understand the context [172]. After that, we add the decoder which is a Bidirectional-LSTM layer. It assumes a three dimensional input for creating a decoded sequence of various lengths determined by the problem. So we configure first the RepeatVector layer to create a three dimensional BiLSTM output. Then, like the encoder, one Bidirectional-LSTM layer with the same number of cells was utilized in the decoder model implementation. Finally, the dense layer generates the autoencoder output which is also a matrix of size $m \times n$ which n is the tweet corpus (50.000).

The training procedure of the GloVe-BiLSTM autoencoder is summarized in Algorithm 2.

In this part, we will explain the Predictor model and how it works. The key idea of our proposed framework (Hybrid-MELAu) is using transfer learning [173], by copying both the pre-trained encoder part of the first n layers of DALS and GloVe—BiLSTM to the n first layers of the deep learning classifiers. The implemented classifiers are 1)— a Recurrent Neural

Algorithm 1: Deep Dense Autoencoder based on Lexical and Syntactic content.

```
Input  : $X$: vector of unlabeled features
         $\lambda$: hyper-parameters
         $T$: the maximum number of iteration
Output : $\hat{X}$: reconstructed representation of the input
begin
    // Preparing data to be passed to the network
    Set $t$ to 1
    Initialize $w, w', b, c$
    repeat
        Encode the input $X$ into the latent space $h$.
        Decompress the original input from the latent space $h$.
        $E(X, \hat{X}) = ||X - \hat{X}||^2$ (the error rate).
        $t = t + 1$.
        Update $(w, w', b, c)$.
    until $t > T$;
    return $\hat{X}$;
end
```

Network (RNN) [44] classifier, which is a universal approximation of dynamical systems, 2)— Long short-term memory networks (LSTMs) [45] predictor, which is considered as an update of RNN that used on several works for example in the paper [174], 3)— Gated recurrent units (GRU) [175] classifier, 4)- three bidirectional architectures (BiRNN [47], BiLSTM and BiGRU [176]).

During the predictor models training, we set the mean squared error (MSE) as a cost function. It is defined below:

$$L(Y, f(X, s)) = L(Y, \hat{Y}) = \frac{1}{N}\sum_{i=1}^{N}(Y, \hat{Y})^2 \tag{5.3}$$

Where N is the feature dimensionality, X is the features vectors and s is a set of tweets, Y is the output ground truth, and $\hat{Y}$ is the predicted output (Human or Bot).

Using a pre-trained network that is trained on data with one class only ensures that the bot detection task is performed based on the most frequent characteristics of non-intrusion samples.

5.3 Experiments and results

5.3.1 Datasets

Several tweeter real-world datasets are used in our research. The first is defined in [150, 177]. According to [150]), genuine accounts consists of 3,474 real users accounts with 8, 377, 522 tweets. The bots accounts separated on three datasets. During the 2014 Romanian Mayoral election, the social spambots1 dataset was scraped from Twitter, it is composed of 991 accounts and 1, 610, 176 tweets. Spambots 2 dataset is a group of 3, 457 bots accounts who

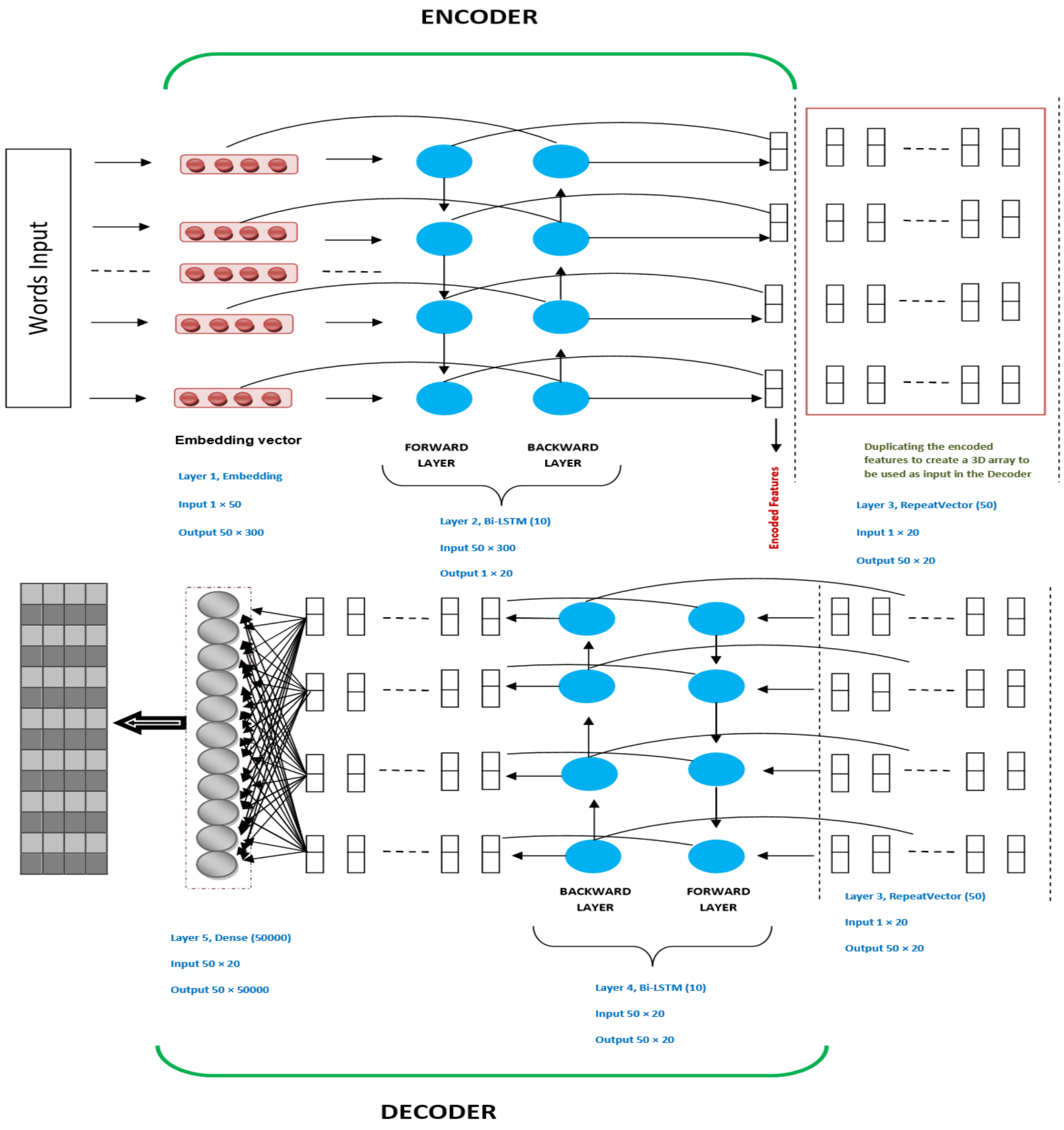


Figure 5.3: GloVe-BiLSTM autoencoder Flow Diagram.

Algorithm 2: GloVe-BiLSTM autoencoder

```
Input  : $S$: set of tweets
         $K$: set of tokens in one tweet: size $m$
         $C$: The tweet corpus: size $n$
         Batch: the number of training examples utilized in one iteration: size $z$
         $\theta$: hyper-parameters
Output : $\hat{P}$: reconstructed matrix: size $(m \times n)$
begin
    // Preparing data to be passed to the stack
    foreach $s \in S$ do
        $s \leftarrow$ nlp.prepossessing $(s)$
    end
    repeat
        foreach $Batch$ do
            // Calculating embeddings for each token
            foreach $k \in K$ do
                emb($k$) $\leftarrow$ glove($k$)
            end
            Encoder_input size = BuildModel(LSTM_Bidirectional.input
              $([m \times Embedding\_size], \theta)$)
            Encoder_output size $\leftarrow$ $[1 \times Thedoublenumberofcells]$
            // repeat Encoder_output size $m$ times to create 3D
               vector
            repeat($Encoder\_output\ size, m$)
            output_size $\leftarrow$ [$m$×Number of cells$*$2]
            Decoder_input size = BuildModel(LSTM_Bidirectional.input
              $(output, \theta)$)
            Decoder_output size $\leftarrow$ [$m$× Number of cells$*$2]
            // Generating the output using a fully connected layer
               with size $n$
            $\hat{P} = [m \times n]$
        end
    until Untill convergence;
    return $\hat{P}$ ;
end
```

passes many months promulgating the #TALNTS hashtag through 428, 542 tweets. Where this last concerns a mobile phone application for contacting and recruiting artists working in several fields. The immense generality of tweets were innocuous statements, sporadically scattered by tweets naming a specific human account and recommending that he purchases the VIP edition of the software from a Web store. The dataset of Spambots 3 is a set of 464 accounts and 1, 418, 626, this dataset announced products for selling on Amazon.com. The delusive activity is executed by spamming URLs referring to the publicized products.

The second one is the celebrity dataset which contains celebs' accounts [178]. The Center

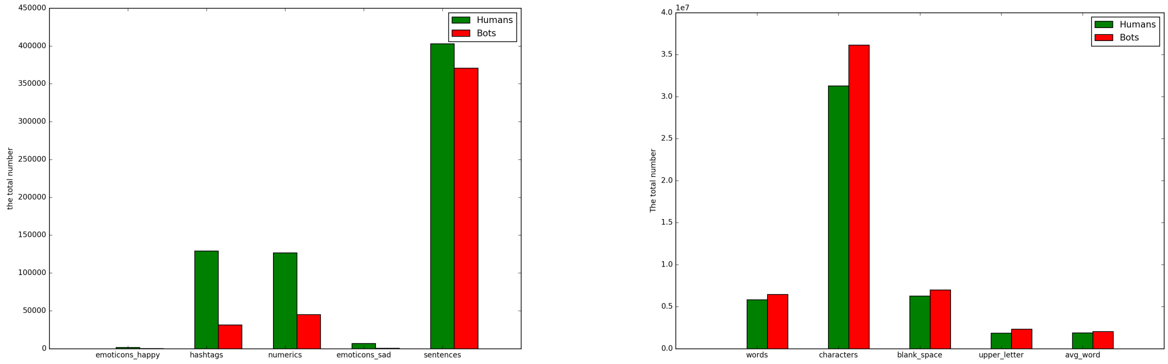


Figure 5.4: Comparison between the writing style of both humans and bots at the Lexical level.

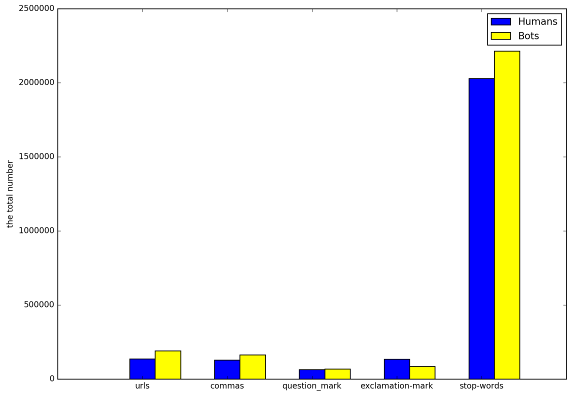


Figure 5.5: Comparison between the writing style of both humans and bots at the syntactic level.

for Complex Networks and Systems Research at Indiana University (CNetS team) collected 5,918 celebrity human accounts. We also add two other datasets: pronbots-2019 and political-bots [178]. Pronbots-2019 is a set of 21,963 bot accounts distributed by Andy Patel. Political is a set of 62 Automated political accounts.

5.3.2 Exploratory analysis results

To extract syntactic and lexical characteristics, we have applied NLP analysis approaches. Hither, we consider a sample of 1 000 000 data containing an equal number of human and bots tweets and compare the writing style of both at the lexical level. The findings of this comparison are illustrated in Figure 5.4. We observe that humans use a greater number of hashtags than bots. Also, they use different numbers of emotion types, numbers, sentences, words, blank spaces, and upper letters. It is because the human can easily diversify their lexical context.

Then, we compare the syntactic features analysis results based on URLs, punctuation, and stop words. As we can see in Figure 5.5, the number of different syntactic tokens could be different, especially since a successful bot can use a linguistic approach based on the linguistic structure. For syntactic analysis based on speech-tagging, there are many tags, so, we have

Table 5.2: Part of speech tags used.

	Tag	Description	Example
1.	CC	coordinating conjunction	and, but, or,..
2.	DT	determiner	a, the,..
3.	FW	foreign word (another language)	maître, sonrisa,..
4.	IN	preposition	of, in, by,..
5.	JJ	adjective	tall,..
6.	JJR	adj, comparative	bigger,..
7.	JJS	adj, superlative	wildest,..
8.	MD	modal	could, should,..
9.	NN	noun, singular	desk,..
10.	NNS	noun, plural	desks,..
11.	NNP	proper noun, singular	Harrison,..
12.	NNPS	proper noun, plural	Americans,..
13.	PRP	personal pronoun	I, you,..
14.	RB	adverb	expertly, inside,..
15.	RBR	adverb, comparative	faster,..
16.	RBS	adverb, superlative	fastest,..
17.	RP	particle	up,off,..
18.	UH	interjection	oops!, oh gosh!, wow!,..
19.	VB	verb base form	walk,..
20.	VBD	verb, past tense	ate,,..
21.	VBG	verb, present participle/gerund	eating,..
22.	VBN	verb, past participle	eaten,..
23.	VBP	verb non-3sg pres	eat,..
24.	VBZ	verb 3sg pres	eats,..

just focused on recognizing some tags, that essentially help in the interpretation of the given sentence. (see Table 5.2). Besides, we compare the different POS tagging features in the bots and human writing styles (Figure 5.6).

From Figure 5.6, we notice that bots used to write their tweets, the following features: proper plural noun, proper singular noun, plural noun, singular noun, prepositions, coordinating conjunctions, determiners, modal, verbs 3sg pres, verbs base form, adjective, comparative adjective, superlative adjective, superlative adverbs and comparative adverbs much more than humans. Because these characteristics are considered basic units (tokens) in the construction of the sentences and aren't difficult to simulate. Although this is a good bot imitation, they haven't been able to outperform humans in terms of features shown on the right side (personal

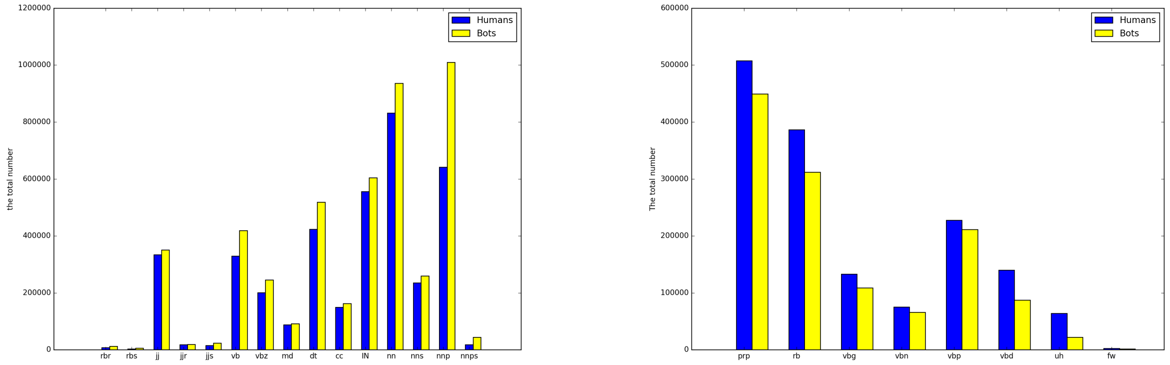


Figure 5.6: Comparison of different POS tagging features. the left part finds out the most characteristics employed by bots in comparison to humans, while the right side shows the features for which humans surpassed the bots.

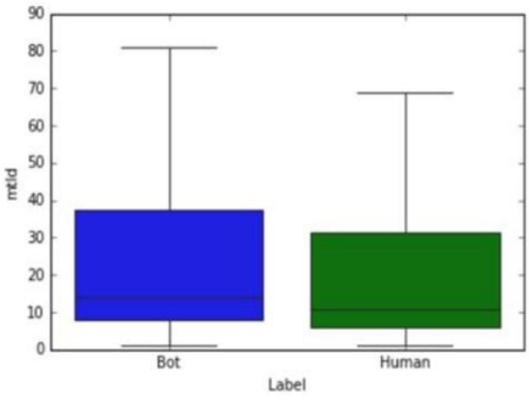


Figure 5.7: Variation of MTLD metric between the human and bot tweets.

pronoun, adverb, verb past tense, verb present participle, verb past participle, verb non-3sg pres, interjections, and foreign words from other languages). The key idea is that exploiting this feature set is more complicated and requires special conditions. For example, humans make use of various interjections with rich context. Therefore, we can conclude that human varies their tone in writing depending on their feelings, the reader, and the events by using empathy, encouragement, and astonishing events.

For the lexical richness task, We chose the MTLD metric due to the fact that it is a robust lexical diversity indicator that is unaffected by sample length [179].

First, we compute all the POS (part-of-speech) tagged to rich inflectional languages. After that, we compute the MTLD. Figure 5.7 shows how the MTLD metric varies between the human and bot tweets.

As we can observe from Figure 5.7 and Table 5.3, the range of the bot's MTLD values is bigger than the human range values. The maximum value of the bots' MTLD is higher than the humans' MTLD while both minimum values are equal. It can be seen that MTLD is a metric for analyzing the number of consecutive words supported by a specific type-token ratio. We observe that a well-automated bot relies on the NLP rules to generate a rich lexicon

Table 5.3: Summary values of Measure of Textual Lexical Diversity distribution across the two lables.

Measure of Textual Lexical Diversity (MTLD)							
	Min	Max	Mean	25th percentile	Median	75th percentile	IQR
Human	1.0	69.91	25.23	6.0	11.0	31.5	25.5
Bot	1.0	81.55	27.38	8.0	14.0	37.33	29.33

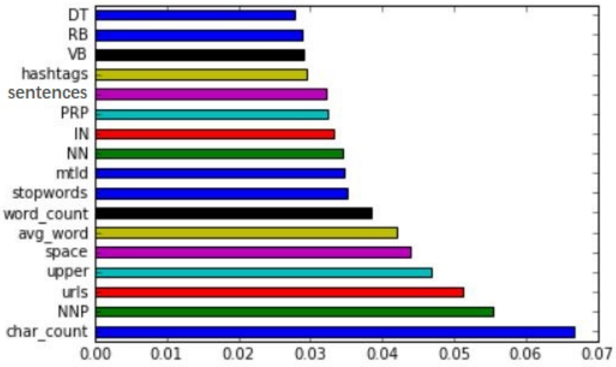


Figure 5.8: Top 17 most important features in the data using Extra Trees Classifier.

but human is using an odd approach as their writing skills outperform simple NLP rules.

5.3.3 Hybrid-MELAu evaluation results

For this evaluation phase, we have selected the 17 highest linguistic features that have a great impact on predictability (see Figure 5.8). We split Cresci datasets into approximately 80% training and 20% testing sets. As mentioned in the 5.2.2 subsection, the two autoencoders will be trained on data with one class only to ensure that the prediction task is performed based on the most frequent characteristics of human samples. So, the training group is divided again based on the dataset label (Human and Bot). The human and bot label rates are respectively 54% and 46% of the training set. Then, we rely on the training set of the human class. After dividing the dataset into 75% training set and 25% validation set, and for retrieving the best hyper-parameters of the two autoencoders, we used one of the optimization approaches that are provided in scikit-learn: "GridSearchCV" [180]. It evaluates all potential values of parameter composition and retains the best one. Table 5.4 shows the autoencoder hyper-parameters after using GridsearchCV.

The top hyper-parameters are:

- Utilization 256 as a batch size for the both autoencoders.

- The usage of "Adam" and "Nadam" separately as optimizer functions for the DALS

TABLE 5.4: GridsearchCV for the best hyper-parameters optimization.

	Optimizer	**Hidden Activation Function**	**Output Activation Function**	**Loss**	**Batch size**	**Learning rate**
1	SGD	softmax	softmax	mse	16	0.00001
2	RMSprop	softplus	softplus	sparse-categorical-crossentropy	32	0.0001
3	Adagrad	softsign	softsign	msle	64	0.001
4	Adadelta	relu	relu	categorical-crossentropy	128	0.01
5	Adam	tanh	tanh	kullback-leibler-divergence	256	0.1
6	Adamax	sigmoid	sigmoid	mae	512	-
7	Nadam	hard-sigmoid	hard-sigmoid	binary-crossentropy	-	-
8	-	linear	linear	hinge	-	-
9	-	elu	elu	squared-hinge	-	-
10	-	selu	selu	-	-	-
DALS	**Adam**	**relu**	**linear**	**mse**	**256**	**0.0001**
GloVe-BiLSTM autoencoder	**Nadam**	**tanh**	**softmax**	**sparse-categorical-crossentropy**	**256**	**0.001**

and GloVe-BiLSTM autoencoder.

- The DALS loss function is MSE and for GloVe-BiLSTM autoencoder is sparse-categorical-crossentropy.

- The learning rate values for the first autoencoder and the second one are 0.0001 and 0.001.

- For the hidden activation function the choice fell on “relu” for the DALS and “tanh” for the GloVe-BiLSTM autoencoder. And for the output activation function, linear and softmax functions were selected respectively for the two autoencoders.

First, the two autoencoders were trained on the dataset based on human class only using the selected linguistic features and the best hyperparameters. Then, the two encoder parts are frozen and cemented to make one feature vector. After this phase, six recurrent neural network models were built as follows: the feature vector was repeated once to create a 3D output utilizing RepeatVector, and it’s fed to the next layer of six classifiers: (1 — SimpleRNN classifier, 2 — BiRNN classifier, 3 — LSTM classifier, 4 — BiLSTM classifier, 5 — GRU classifier, 6 — BiGRU classifier). The unit number of cells in each one is fixed at 300.

Afterward, to make a one-dimensional vector a flattened layer was added. To render the model more powerful, the output vector is passed to a fully connected layer. Then, the last layer transforms its input into one result using the sigmoid function [149].

The different recurrent neural classifiers are implemented on the whole labeled dataset with 55% of data for the training set, the previous preserved testing set (20% of data), and 25% of data for the validation set using the Google Colab environment. The runtime had configured to use Keras [181] API v2.4.3, Tensorflow v2.4.0, Python 3.6.9 - 64bit-, a GPU Hardware accelerator. The classifiers were trained for 150 epochs with 256 as a batch size using Adam as an optimization function and mse as a loss function. For the fully connected layer and output layer we used ReLu and sigmoid activation functions respectively. We employ different metrics : *Precision*, *Recall*, *F-Measure*, *Accuracy* and *Matthew Correlation Coefficient (MCC) [182]* to compare the classifiers performance.

- *Precision*: the percentage of classes that are correctly positive when expected to be positive.
- *Recall (or also Sensitivity)*: the ratio of right positive classes that are predicted positive.
- *F-Measure*: the recall and precision harmonic mean.
- *Accuracy*: the percentage of actual results that were anticipated in the samples.
- *Matthew Correlation Coefficient (MCC) [182]*: infers the correlation between the correct class and the one predicted.

Every metric apprehends various aspects of the prediction capacity. Accuracy scales the users' number is correctly classified in both classes, but it does not debrief whether the positive class is better identified than the other one. Moreover, there are cases in which some predictive models are implemented better than others, even if there is low accuracy [183]. Getting richly precision means that numerous relevant samples are recognized correctly but don't provide information about the relevant samples that are not identified. This information is provided by the Recall, which expresses the samples' numbers, in the full range of relevant samples, have been properly identified: a low recall means that many relevant samples are left anonymous. At last, F-Measure and MCC report the quality of the prediction. Moreover, The confusion matrix's four components are used by the MCC measure [183]. The prediction is indicated extremely accurate when the MCC ≈ 1, instead of MCC ≈ -1 which reveals that the prognosis is noticeably at odds with the actual class, MCC ≈ 0 shows that the prediction is no better than random guessing.

Experiments on the Cresci dataset show that it is possible to forecast with a high degree of accuracy. As we can see from Table 5.5, the Hybrid-MELAu+BiRNN classifier shows high performance for bot detection, and it is better than the other recurrent classifiers when the overall accuracy is 92.22%. All recurrent classifiers had closely comparable performance.

Table 5.5: Comparison among the various presented approaches in terms of performance [7].

Classifiers	**Precision**	**Recall**	**F1-score**	**Accuracy**	**Loss**	**MCC**
Hybrid-MELAu+SimpleRNN	0.92455	0.9102	0.91375	0.9154	0.0718	0.8347
Hybrid-MELAu+BiRNN	**0.9318**	**0.9169**	**0.92065**	**0.9222**	**0.0654**	**0.8486**
Hybrid-MELAu+LSTM	0.9231	0.908	0.91165	0.9134	0.0728	0.8310
Hybrid-MELAu+BiLSTM	0.93085	0.9168	0.92035	0.9219	0.0657	0.8476
Hybrid-MELAu+GRU	0.92195	0.90705	0.91065	0.9124	0.0740	0.8289
Hybrid-MELAu+BiGRU	0.9311	0.9166	0.92025	0.9218	0.0658	0.8476

After that, because our Hybrid-MELAu (with BiRNN) model falls under the semi-supervised techniques, we choose to compare its performance with the methods mentioned in Table 3.2, the results are presented in Figure 5.9.

As we can see from Figure 5.9, the Hybrid-MELAu outperformed the other models in terms of the different metrics.

In fact, in this work, we emphasize linguistic features without taking into account the users' features to discriminate the human's and bots writing style behavior. Hence, this result illustrates the ability of the feature learner based on autoencoders with transfer learning to generate elite features from latent spaces from the pre-trained encoder part.

Certainly, the linguistic features capture sentence level and word level complexity using different lexical and syntactic indexes to influence bot identification and show better results. It also showed that, when compared to human accounts, Bot accounts have a high non-homogenize in their discriminatory behavioral characteristics [184]. Designing a deep linguistic framework with transfer learning founded only on features of linguistic characteristics can define if a single tweet is being written by a human or a bot with good accuracy. Moreover, the generative ability of the part of the pre-trained encoder enhances the predictor to discern differences in the writing styles of both humans and bots.

5.4 Discussion

In this section, we will discuss the main findings of the manuscript and address its implications. Moreover, we will test the proposed model's robustness by introducing further experiments. Whilst the majority of work on bot detection has focused on investigating various sets of features, our first concern in this present research is the analysis of the bot writing style by using Natural Language Processing (NLP) to find insights about how linguistic features help in bot detection. Our research reveals that certain lexical and syntactic measures are

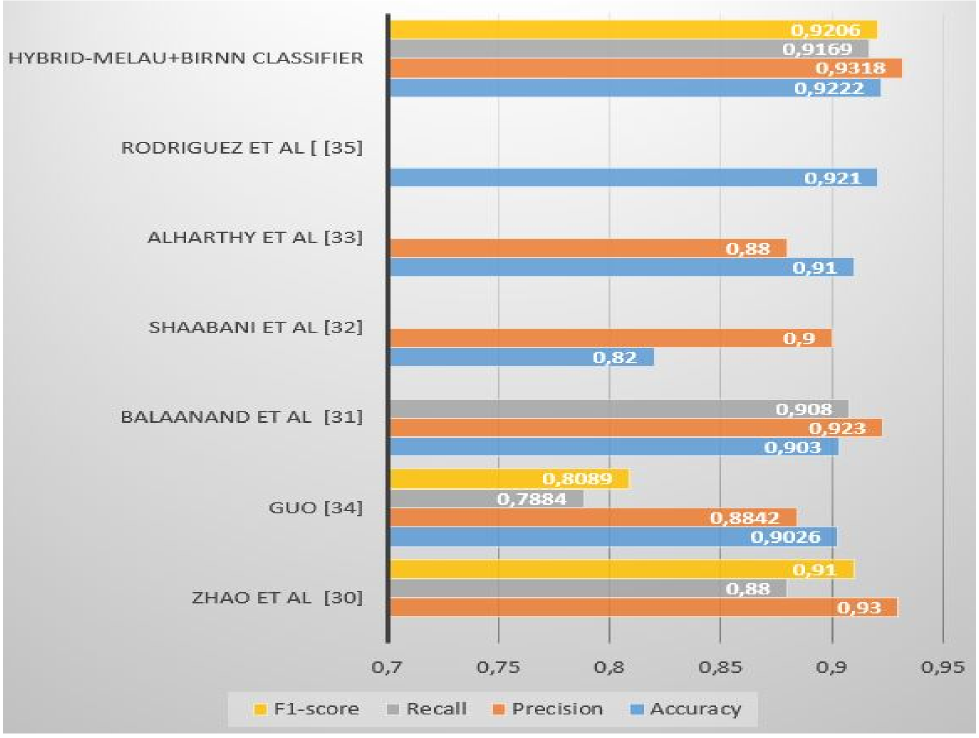


FIGURE 5.9: Experiments Results [7].

the most significant signs that contribute to distinguishing the writing style of both bots and humans. The exploratory analysis showed that humans could infer the relationship between different contexts by employing a context-related lexical level (as discussed in section 5.3.2). Unlike bots, humans intend to express and argue their ideas using numerics (digits, dates, real numbers). In addition, humans use more phrases in one tweet than the bots.

According to the syntactic analysis based on speech-tagging (see Figure 5.6), we find that although humans master the language's syntax, they show creative behavior in their writing style. Therefore, humans make use of interjections in a specific sentence related to their psychological state and their feelings. They might also express a position whether the latter is personal or related to another person. For example, in this human tweet "hmm fishy!!" an exclamation sentence existed, conveying that the person expresses arousing feelings of doubt or suspicion. As we can note there is no grammatical structure in this sentence, it's just composed of two words, an interjection (hmm) and an adverb (fishy). Furthermore, the human explains a specific statement taking profit from a variety of adverbs and personal pronouns. It means that they tend to diversify their writing styles according to a somewhat odd approach to tweeting their ideas, such as using words in foreign languages. They don't also focus on one tense to conjugate verbs. These results represent a strong conclusion to discern the difference between humans' and bots' writing styles.

Furthermore, computing the lexical diversity measures (see Figure 5.7) would further disseminate the writing style from humans and bots, which can be seen differently in a text

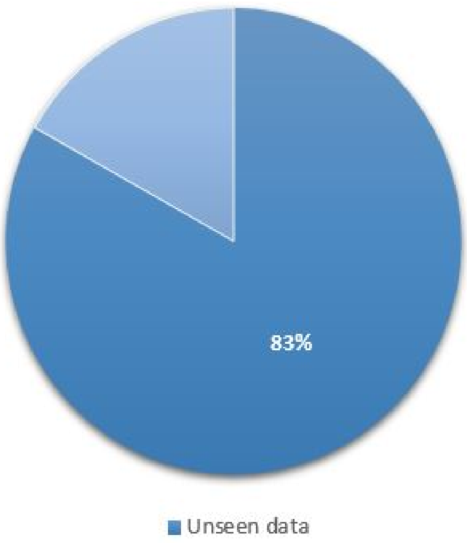


Figure 5.10: The prediction accuracy of Hybrid-MELAU+BiRNN classifier on an unseen data (Celebrity, pronbots-2019 and political bots dataset).

depending on specific type-token ratio and vocabulary knowledge. We conclude that a successful (well-automated) bot includes the NLP approaches to generate tweets and get a rich lexicon. Meanwhile, human includes their skills with the language to write in an intelligent way (”To live with untreated PTSD is to feel like you might die any moment. Again and again. Help costs money.”). Even though the machine learning techniques used in bots through NLP have improved their ability to generate content with high lexical diversity, as we can see from this bot tweet: ”Today’s Inspirational Quote Climb the mountains and get their good tidings. Nature’s peace will flow into you as...”, there is still a lot to do to imitate the smart human writing.

Our second concern was how to develop a hybrid deep learning approach based only on linguistic features that can improve detection performance. This can be achieved by building a framework in a hybrid fashion Mixing Engineered Linguistic features based on Autoencoders (Hybrid-MELAu). In fact, deep neural networks’ versatility allows them to integrate numerous neural building blocks to construct a more powerful hybrid model by complementing one another. The autoencoder has shown to be a useful model for modeling latent distributions since it allows you a lot of control.

To demonstrate the model’s sturdiness, we tested our framework’s prediction performance on a new unseen dataset that combined three datasets: celebrity, pronbots-2019, and political bots. The capacity of a predictive model to perform well over a variety of data sets determines its robustness. Therefore, the resilience of the models built in this study was tested on this new dataset after they had been trained with the Cresci dataset. Figure 5.10 illustrates a good prediction result when applied to an unseen dataset. Our framework ensures efficient detection because once the autoencoder model is trained, its results will be used directly for transfer learning without the need to resort to the two features of learners’ training.

The fact that our framework is semi-supervised with one-class authorization benefits from the myriad of unlabeled training data for learning task performance amelioration because the amount of unlabeled samples is generally greater and more accessible than the number of labeled samples.

Finally, the findings of this work also show that pre-trained models based on transfer learning can improve the accuracy of bot detection. Surprisingly, a set of linguistic features, such as those obtained from our exploratory study, are effective in distinguishing social bots. In future works, since we have found that the linguistic deep framework with a transfer learning model is discernible of the bot's writing style, we are going to incorporate the different sets of features in our framework. This could help with social bot detection performance.

5.5 Conclusion

We develop the Hybrid-MELAu: a semi-supervised framework to model different mixing engineered linguistic features based on autoencoder and use the transfer learning to take profit from its strong ability to generalize to unseen samples, which improves the social bots detection. The framework is composed of two essential parts: the features learner and the predictor. The features learner combines two encoder parts: i) the DALS (Deep Dense Autoencoder based on Lexical and Syntactic features and ii) The GloVe-BiLSTM. The DALS maps the content features to higher-order features, which enables the lexical richness to be encompassed. The GloVe-BiLSTM trains two LSTMs instead of one on the input sequences. This work combines the concept of transfer learning and one-class classification. This can provide reliable semantic features and result in accurate learning on the detection. The proposed approach captures different lexical and syntactic indexes that influence bot detection and shows significant results. Our new mechanism for detecting bots based on a mining writing style effectively detects bots with a 92.22% accuracy rate.

Chapter 6

Conclusion

*****

Considering the negative implications of the Fake news diffusion on social media, the importance of detecting and fighting this phenomenon rises continuously. In this Phd thesis, We clarify the phenomena of false news, its primary sources, and examine various current ways to identify fake news from the standpoint of feature extractions and modeling techniques. Numerous automatic detection methods have been created utilizing common machine learning techniques and many experiments have been conducted on real world dataset collected from Twitter's content. In fact, Twitter is a popular web application and online social network, where a large part of its accounts is managed by automated programs called Bots. Some of these bots are legitimate and have great purposes, such as delivering a piece of useful information and providing assistance during emergency cases. While the malicious bots work to spread malware to influence the perception of the public about a topic. To help human users to identify with whom they interact, our works focuses on the detection of human and bot accounts on Twitter. Therefore, we chose to dig deeply the existing approaches to improve the bots detection by proposing two principal contributions from the perspective of features engineering.

The first contribution is a Bidirectional long short-term memory (BiLSTM) model that integrates linguistic cues of twitter text. Here, we present a Deep Bi-LSTM model based on sentiment features. Building on the BiLSTM prediction, the suggested architectures rely on both the upcoming and prior inputs. Here, the following is a summary of our contributions.:

- We extracted a set of features which include sentiment polarity and subjectivity, the number of the happy emoticons and the sad emoticons, the number of the interjections (i.e. ah!, ooh!..) in the tweets. Besides, we extracted the semantic features using embedding vectors.

- The deep BiLSTM model incorporates sentiments features (polarity, subjectivity, happy emoticons, sad emoticons and the interjections). The BiLSTM model ties two hidden layers facing different directions together to create one output. The output layer can simultaneously receive data from the past (backwards) and the future (forwards) states with this type of generative deep learning. This method can efficiently capture semantic using word embedding technique.

- We conducted our experiment on real-world data sets. Experiment on the cresci-2017 dataset shows that the proposed approach achieves a competitive classification accuracy. As a result, the model has reached a significant prediction with 97.36% of accuracy. The experiment with real-world data revealed slightly improve of the detection results and illustrated how it is complex to identify the impact of each feature in bot detection problem.

As there has been fairly lower work exploring lexicon measurement and linguistic indicators to detect bots, our second contribution is focusing on the recognition of the social bots through their writing style. Thus, we carried out an exploratory study on the effectiveness of only a set of linguistic features (17 features) exploitable for bot detection, without the need to resort to other types of features. And we develop a novel framework in a hybrid fashion of Mixing Engineered Linguistic features based on Autoencoders (Hybrid-MELAu). The semi-supervised Hybrid-MELAu framework is composed of two essential constituents: the features learner and the predictors. We establish the features learner innovated on two powerful structures: a) the first is a Deep dense Autoencoder fed by the Lexical and the Syntactic content (DALS) that represents the high order lexical and syntactic features in latent space, b) the second one is a Glove-BiLSTM autoencoder, which sculpts the semantic features; subsequently, we generate elite elements from the pre-trained encoder part from each latent space with transfer learning. We consider a sample of 1 Million from Cresci datasets to conduct our linguistic analysis comparison between the writing style of humans and bots. With this dataset, we observe that the bot's textual lexical diversity median is greater than the human one and the syntactic analysis based on speech-tagging shows a creative behavior in human writing style. Finally, we test the model's robustness on several public dataset (celebrity, pronbots-2019, and political bots). Here, we summarized our contributions as follows:

- We extracted various feature sets that indicate the writing style for both humans and bots, to be able to compare and evaluate the performance of the social bot detection model enhanced by the distinct writing patterns in estimating the human and bot classes. The different writing style features are lexical features relying on text richness and diversity, syntactic features based on Pos-tagging, and semantic features that are extracted with word embeddings techniques.

- We developed Hybrid-MELAu: novel semi-supervised framework in a hybrid fashion mixing engineered linguistic features based on autoencoder. Therefore, we trained two autoencoders. The frst is a deep dense autoencoder fed by the lexical and the syntactic features (DALS). The second one is a Glove word embedding BiLSTM autoencoder

(GloVe-BiLSTM autoencoder), which effectively captures the semantic or the contextual features across tweets. Then, we stapled the trained encoder building blocks to generate elite characteristics from both latent spaces. The idea behind this combination is to complement one another; they successfully model the lexical, syntactic, and semantic knowledge. In low-dimensional spaces, this representation will become very efficient.

- We have benefited from the profound features attained from encoders for transfer learning to discern differences in the writing styles of both humans and bots. The initialization of the classifiers with transferred features has improved the performance when modeling the bot detection task.

- We have chained the Hybrid-MELAu output with six Recurrent Neural Network classifers: SimpleRNN, BiRNN, LSTM, BiLSTM, GRU (Gate Recurrent Units), and Bi-GRU.

- Experiments were carried out with a real-world data set show that the introduced framework significantly achieves an accurate social bot detection. Our new mechanism for detecting bot based on a mining writing style effectively detects bots with a 92.22% accuracy. The proposed framework achieves a good accuracy of 92.22%. Overall, the results shown in this work, and the related discussion, contend that an effective linguistic deep framework can be used to distinguish between the writing styles of humans and bots.

Finally, in the future work, we will discuss open-ended questions and future research directions relating to fake news detection paradigms such as:
1- The study of Arabic natural language processing (NLP) is a challenging task. This is caused by a variety of things, including Arabic's complex and rich morphology, its high level of ambiguity, and the existence of several dialects with a wide range of differences. For this reason, In order to provide in-depth knowledge about the impact of text diversity on the detection process for the Arabic languages, we want to apply our approach to data sets with Arabic corpus.
2- Furthermore, we aim at highlighting the human behavioral trends. Actually, considering their activity and dynamics, which are connected to linguistic characteristics, would be a fruitful path for future research.

Bibliography

*****

[1] M. Alyukov, "Propaganda, Authoritarianism and Russia's Invasion of Ukraine," *Nature Human Behaviour*, vol. 6, no. 6, pp. 763–765, Jun 2022. [Online]. Available: https://doi.org/10.1038/s41562-022-01375-x

[2] A. Badawy, E. Ferrara, and K. Lerman, "Analyzing the Digital Traces of Political Manipulation: The 2016 Russian Interference Twitter Campaign," in *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 2018, pp. 258–265. [Online]. Available: https://doi.org/10.1109/ASONAM.2018.8508646

[3] A. Bessi and E. Ferrara, "Social Bots Distort the 2016 U.S. Presidential Election Online Discussion," *First Monday*, vol. 21, 11 2016. [Online]. Available: http://dx.doi.org/10.5210/fm.v21i11.7090

[4] T. Nelson, N. Kagan, C. Critchlow, A. Hillard, and A. Hsu, "The Danger of Misinformation in the Covid-19 Crisis," *Missouri medicine*, vol. 117, pp. 510–512, 11 2020. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7721433

[5] Z. Ferhat Hamida, A. Refoufi, and A. Drif, "Fake News Detection Methods: A Survey And New Perspectives," in *Advanced Intelligent Systems for Sustainable Development (AI2SD'2020)*, J. Kacprzyk, V. E. Balas, and M. Ezziyyani, Eds. Cham: Springer International Publishing, 2022, pp. 123–141. [Online]. Available: https://doi.org/10.1007/978-3-030-90639-9_11

[6] Z. Ferhat Hamida, A. Refoufi, A. Drif, and S. Giordano, "Sentiment Analysis-Based Model for Bot Detection on Social Media," in *1st National Conference on Applied Science and Advanced Materials (NCASAM-2021)*. Skikda, Algeria: ENSET, 20–22 Dec. 2021.

[7] ——, "Hybrid-MELAu: A Hybrid Mixing Engineered Linguistic Features Framework Based on Autoencoder for Social Bot Detection." *Informatica*, vol. 26, no. 6, 2022. [Online]. Available: https://doi.org/10.31449/inf.v46i6.4081

[8] P. Jackson and I. Moulinier, *Natural Language Processing for Online Applications: Text Retrieval, Extraction and Categorization*, 2nd ed., ser. Natural language processing. John Benjamins Publishing, 2007, vol. 5. [Online]. Available: https://benjamins.com/catalog/nlp.5

[9] A. Louis, "NetBERT: A Pre-Trained Language Representation Model for Computer Networking," Master's thesis, Université de Liège, Liège, Belgique, 2020. [Online]. Available: https://matheo.uliege.be/handle/2268.2/9060

[10] C. E. Shannon and W. Weaver, *The Mathematical Theory of Communication.* Urbana: University of Illinois Press, 1949.

[11] R. B. Lees, "Review of Syntactic Structures, by N. Chomsky," *Language*, vol. 33, no. 3, pp. 375–408, 1957. [Online]. Available: https://doi.org/10.2307/411160

[12] R. C. Schank and R. P. Abelson, "Scripts, Plans, and Knowledge," in *Proceedings of the 4th International Joint Conference on Artificial Intelligence - Volume 1*, ser. IJCAI'75. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1975, p. 151–157.

[13] R. E. Cullingford, "Script Application: Computer Understanding of Newspaper Stories." Ph.D. dissertation, Yale University, Connecticut, USA, 1977.

[14] R. C. Schank and R. Wilensky, "A Goal-Directed Production System for Story Understanding," *SIGART Bull.*, no. 63, p. 72, jun 1977. [Online]. Available: https://doi.org/10.1145/1045343.1045385

[15] J. G. Carbonell, "Subjective Understanding: Computer Models of Belief Systems." Ph.D. dissertation, Yale University, Connecticut, USA, 1979.

[16] E. Charniak, "Passing Markers: A Theory of Contextual Influence in Language Comprehension," *Cognitive Science*, vol. 7, no. 3, pp. 171–190, 1983. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S036402138380010X

[17] M. G. Dyer, "The Role of Affect in Narratives," *Cognitive Science*, vol. 7, no. 3, pp. 211–242, 1983. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1207/s15516709cog0703_3

[18] C. K. Riesbeck and C. E. Martin, "Direct Memory Access Parsing," in *Experience, Memory, and Reasoning*, J. L. Kolodner and C. K. Riesbeck, Eds. Psychology Press, 1986, ch. 13. [Online]. Available: https://www.taylorfrancis.com/chapters/edit/10.4324/9781315802169-16

[19] B. J. Grosz, D. E. Appelt, P. A. Martin, and F. C. Pereira, "TEAM: An Experiment in the Design of Transportable Natural-Language Interfaces," *Artificial Intelligence*, vol. 32, no. 2, pp. 173–243, 1987. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0004370287900117

[20] G. Hirst, "Semantic Interpretation and Ambiguity," *Artificial Intelligence*, vol. 34, no. 2, pp. 131–177, 1988. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0004370288900379

[21] L. Bahl, P. Brown, P. de Souza, and R. Mercer, "A Tree-Based Statistical Language Model for Natural Language Speech Recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 7, pp. 1001–1008, 1989. [Online]. Available: https://doi.org/10.1109/29.32278

[22] E. Brill, D. Magerman, M. Marcus, and B. Santorini, "Deducing Linguistic Structure from the Statistics of Large Corpora," in *Proceedings of the 5th Jerusalem Conference on Information Technology, 1990. 'Next Decade in Information Technology'*, 1990, pp. 380–389. [Online]. Available: https://doi.org/10.1109/JCIT.1990.128309

[23] M. V. Chitrao and R. Grishman, "Statistical Parsing of Messages," in *Proceedings of the Workshop on Speech and Natural Language*, ser. HLT '90. USA: Association for Computational Linguistics, 1990, p. 263–266. [Online]. Available: https://doi.org/10.3115/116580.116665

[24] P. F. Brown, J. Cocke, S. A. Della Pietra, V. J. Della Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Roossin, "A Statistical Approach to Machine Translation," *Computational Linguistics*, vol. 16, no. 2, pp. 79–85, 1990. [Online]. Available: https://aclanthology.org/J90-2002

[25] H. Tanaka, "Verbal Case Frame Acquisition from a Bilingual Corpus: Gradual Knowledge Acquisition," in *COLING 1994 Volume 2: The 15th International Conference on Computational Linguistics*, 1994. [Online]. Available: https://aclanthology.org/C94-2116

[26] I. Allmuallim, Y. Akiba, T. Yamazaki, A. Yokoo, and S. Kaneda, "Two Methods for Learning ALT-J/E Translation Rules from Examples and a Semantic Hierarchy," in *COLING 1994 Volume 1: The 15th International Conference on Computational Linguistics*, Kyoto, Japan, Aug. 1994. [Online]. Available: https://aclanthology.org/C94-1006

[27] Y. Bengio, R. Ducharme, and P. Vincent, "A Neural Probabilistic Language Model," in *Advances in Neural Information Processing Systems*, T. Leen, T. Dietterich, and V. Tresp, Eds., vol. 13. MIT Press, 2000. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2000/file/728f206c2a01bf572b5940d7d9a8fa4c-Paper.pdf

[28] R. Collobert and J. Weston, "A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning," in *Proceedings of the 25th International Conference on Machine Learning*, ser. ICML '08. New York, NY, USA: Association for Computing Machinery, 2008, p. 160–167. [Online]. Available: https://doi.org/10.1145/1390156.1390177

[29] Y. LeCun, P. Haffner, L. Bottou, and Y. Bengio, "Object Recognition with Gradient-Based Learning," in *Shape, Contour and Grouping in Computer Vision*, D. A. Forsyth, J. L. Mundy, V. Gesú, and R. Cipolla, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 319–345. [Online]. Available: https://doi.org/10.1007/3-540-46805-6_19

[30] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," in *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2013. [Online]. Available: http://arxiv.org/abs/1301.3781

[31] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and Their Compositionality," in *Advances in Neural Information Processing Systems*, C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, Eds., vol. 26. Curran Associates, Inc., 2013. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2013/hash/9aa42b31882ec039965f3c4923ce901b-Abstract.html

[32] J. L. Elman, "Finding Structure in Time," *Cognitive Science*, vol. 14, no. 2, pp. 179–211, 1990. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1207/s15516709cog1402_1

[33] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts, "Recursive Deep Models for Semantic Compositionality over a Sentiment Treebank," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Seattle, Washington, USA: Association for Computational Linguistics, Oct. 2013, pp. 1631–1642. [Online]. Available: https://aclanthology.org/D13-1170

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

[35] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1409.0473

[36] A. M. Dai and Q. V. Le, "Semi-Supervised Sequence Learning," in *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., vol. 28. Curran Associates, Inc., 2015. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2015/file/7137debd45ae4d0ab9aa953017286b20-Paper.pdf

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

[38] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds. Minneapolis, Minnesota: Association for Computational Linguistics, jun 2019, pp. 4171–4186. [Online]. Available: https://aclanthology.org/N19-1423

[39] J. Howard and S. Ruder, "Universal Language Model Fine-Tuning for Text Classification," *CoRR*, vol. abs/1801.06146, 2018. [Online]. Available: http://arxiv.org/abs/1801.06146

[40] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language Models are Unsupervised Multitask Learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[41] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "XL-Net: Generalized Autoregressive Pretraining for Language Understanding," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2019/file/dc6a7e655d7e5840e66733e9ee67cc69-Paper.pdf

[42] L. Zhang, S. Wang, and B. Liu, "Deep Learning for Sentiment Analysis: A Survey," *WIREs Data Mining and Knowledge Discovery*, vol. 8, no. 4, p. e1253, 2018. [Online]. Available: https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/widm.1253

[43] S. Raaijmakers, *Deep Learning for Natural Language Processing*. Manning, 2022.

[44] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning Representations by Back-Propagating Errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct 1986. [Online]. Available: https://doi.org/10.1038/323533a0

[45] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997. [Online]. Available: https://doi.org/10.1162/neco.1997.9.8.1735

[46] K. Cho, B. van Merrienboer, Ç. Gülçehre, F. Bougares, H. Schwenk, and Y. Bengio, "Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation," *CoRR*, vol. abs/1406.1078, 2014. [Online]. Available: http://arxiv.org/abs/1406.1078

[47] M. Schuster and K. K. Paliwal, "Bidirectional Recurrent Neural Networks," *IEEE Transactions on Signal Processing*, vol. 45, pp. 2673–2681, 1997. [Online]. Available: https://doi.org/10.1109/78.650093

[48] A. Graves, S. Fernández, and J. Schmidhuber, "Bidirectional LSTM Networks for Improved Phoneme Classification and Recognition," in *Artificial Neural Networks: Formal Models and Their Applications – ICANN 2005*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 799–804. [Online]. Available: https://doi.org/10.1007/11550907_126

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

[50] J. Pennington, R. Socher, and C. Manning, "GloVe: Global Vectors for Word Representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, 2014, pp. 1532–1543. [Online]. Available: https://doi.org/10.3115/v1/D14-1162

[51] H. Lane, C. Howard, and H. M. Hapke, *Natural Language Processing in Action: Understanding, Analyzing, and Generating Text with Python.* Shelter Island: Manning Publications Co., 2019.

[52] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching Word Vectors with Subword Information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 06 2017. [Online]. Available: https://doi.org/10.1162/tacl_a_00051

[53] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of Tricks for Efficient Text Classification," *CoRR*, vol. abs/1607.01759, 2016. [Online]. Available: http://arxiv.org/abs/1607.01759

[54] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[55] S. J. Pan and Q. Yang, "A Survey on Transfer Learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010. [Online]. Available: https://doi.org/10.1109/TKDE.2009.191

[56] S. Briet, *Qu'est-ce que la documentation?*, ser. Collection de documentologie. Éditions documentaires, industrielles et techniques, 1951, no. 1. [Online]. Available: https://ci.nii.ac.jp/ncid/BA68456042

[57] J. Meyriat, "Document, documentation, documentologie. L'écrit et le document," *Schéma et schématisation*, vol. 14, no. 1, pp. 51–63, 1981.

[58] A. Guillén, Y. Gutiérrez, and R. Muñoz, "Natural Language Processing Technologies for Document Profiling," in *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017.* Varna, Bulgaria: INCOMA Ltd., Sep. 2017, pp. 284–290. [Online]. Available: https://doi.org/10.26615/978-954-452-049-6_039

[59] A. M. Kaplan and M. Haenlein, "Users of the World, Unite! The Challenges and Opportunities of Social Media," *Business Horizons*, vol. 53, no. 1, pp. 59–68, 2010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0007681309001232

[60] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, "Fake News Detection on Social Media: A Data Mining Perspective," *SIGKDD Explor. Newsl.*, vol. 19, no. 1, p. 22–36, sep 2017. [Online]. Available: https://doi.org/10.1145/3137597.3137600

[61] S. Jayakumar, B. Ang, and N. Anwar, *Disinformation and Fake News*. Palgrave Macmillan Singapore, 01 2021, p. 158. [Online]. Available: https://doi.org/10.1007/978-981-15-5876-4

[62] Ö. Özgöbek and J. A. Gulla, "Towards an Understanding of Fake News," in *CEUR workshop proceedings*, vol. 2041, 2017, pp. 35–42. [Online]. Available: https://ceur-ws.org/Vol-2041/paper4.pdf

[63] S. Kumar and N. Shah, "False Information on Web and Social Media: A Survey," *CoRR*, vol. abs/1804.08559, 2018. [Online]. Available: http://arxiv.org/abs/1804.08559

[64] D. Fallis, “A Functional Analysis of Disinformation,” *IConference 2014 Proceedings*, 2014. [Online]. Available: https://doi.org/10.9776/14278

[65] P. Hernon, “Disinformation and Misinformation Through the Internet: Findings of an Exploratory Study,” *Government Information Quarterly*, vol. 12, no. 2, pp. 133–139, 1995. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0740624X95900527

[66] J. E. Thomas, “Statements of Fact, Statements of Opinion, and the First Amendment,” *California Law Review*, vol. 74, no. 3, pp. 1001–1056, 1986. [Online]. Available: https://doi.org/10.2307/3480402

[67] C. Wardle, “Fake News. It’s Complicated.” *First Draft*, 16 2017. [Online]. Available: https://firstdraftnews.org/articles/fake-news-complicated/

[68] C. Jack, “Lexicon of Lies: Terms for Problematic Information,” Data & Society Research Institute, Tech. Rep., 2017. [Online]. Available: https://apo.org.au/node/183786

[69] V. Rubin, *Misinformation and Disinformation: Detecting Fakes with the Eye and AI*. Springer International Publishing, 2022. [Online]. Available: https://link.springer.com/book/10.1007/978-3-030-95656-1

[70] C. Wardle and H. Derakhshan, “Information Disorder: Toward an Interdisciplinary Framework for Research and Policymaking,” Council of Europe Strasbourg, Tech. Rep., 2017. [Online]. Available: https://edoc.coe.int/en/media/7495-information-disorder-toward-an-interdisciplinary-framework-for-research-and-policy-making.html

[71] E. Ferrara, O. Varol, C. Davis, F. Menczer, and A. Flammini, “The Rise of Social Bots,” *Commun. ACM*, vol. 59, no. 7, p. 96–104, jun 2016. [Online]. Available: https://doi.org/10.1145/2818717

[72] Z. Chu, S. Gianvecchio, H. Wang, and S. Jajodia, “Detecting Automation of Twitter Accounts: Are You a Human, Bot, or Cyborg?” *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 6, pp. 811–824, 2012. [Online]. Available: https://doi.org/10.1109/TDSC.2012.75

[73] J. Cheng, M. Bernstein, C. Danescu-Niculescu-Mizil, and J. Leskovec, “Anyone Can Become a Troll: Causes of Trolling Behavior in Online Discussions,” in *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, ser. CSCW ’17. New York, NY, USA: Association for Computing Machinery, 2017, p. 1217–1230. [Online]. Available: https://doi.org/10.1145/2998181.2998213

[74] Z. Jin, J. Cao, Y. Zhang, J. Zhou, and Q. Tian, “Novel Visual and Statistical Image Features for Microblogs News Verification,” *IEEE Transactions on Multimedia*, vol. 19, no. 3, pp. 598–608, 2017. [Online]. Available: https://doi.org/10.1109/TMM.2016.2617078

[75] C. Castillo, M. Mendoza, and B. Poblete, "Information Credibility on Twitter," in *Proceedings of the 20th International Conference on World Wide Web*, ser. WWW '11. New York, NY, USA: Association for Computing Machinery, 2011, p. 675–684. [Online]. Available: https://doi.org/10.1145/1963405.1963500

[76] S. Kwon, M. Cha, K. Jung, W. Chen, and Y. Wang, "Prominent Features of Rumor Propagation in Online Social Media," in *2013 IEEE 13th International Conference on Data Mining*, 2013, pp. 1103–1108. [Online]. Available: http://dx.doi.org//10.1109/ICDM.2013.61

[77] J. Ma, W. Gao, Z. Wei, Y. Lu, and K.-F. Wong, "Detect Rumors Using Time Series of Social Context Information on Microblogging Websites," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, ser. CIKM '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 1751–1754. [Online]. Available: https://doi.org/10.1145/2806416.2806607

[78] IFLA, "How to Spot Fake News," The International Federation of Library Associations and Institutions. Retrieved from https://www.ifla.org/publications/node/11174, 13 Mar. 2017.

[79] S. Hosseinimotlagh and E. Papalexakis, "Unsupervised Content-Based Identification of Fake News Articles with Tensor Decomposition Ensembles," in *Proceedings of the MIS2 Workshop held in conjuction with 11th International Conference on Web Search and Data Mining. MIS2*, 02 2018.

[80] S. Yang, K. Shu, S. Wang, R. Gu, F. Wu, and H. Liu, "Unsupervised Fake News Detection on Social Media: a Generative Approach," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 5644–5651, Jul. 2019. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/4508

[81] S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-6, no. 6, pp. 721–741, 1984. [Online]. Available: https://doi.org/10.1109/TPAMI.1984.4767596

[82] P. Deka, A. Jurek-Loughrey, and D. P, "Evidence Extraction to Validate Medical Claims in Fake News Detection," in *Health Information Science*, A. Traina, H. Wang, Y. Zhang, S. Siuly, R. Zhou, and L. Chen, Eds. Cham: Springer Nature Switzerland, 2022, pp. 3–15. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-031-20627-6_1

[83] R. K. Pick, V. Kozhukhov, D. Vilenchik, and O. Tsur, "STEM: Unsupervised Structural Embedding for Stance Detection," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 10, pp. 11 174–11 182, Jun. 2022. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/21367

[84] A. Silva, L. Luo, S. Karunasekera, and C. Leckie, "Embracing Domain Differences in Fake News: Cross-Domain Fake News Detection Using Multi-Modal Data," *Proceedings of the*

AAAI Conference on Artificial Intelligence, vol. 35, no. 1, pp. 557–565, May 2021. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/16134

[85] D. Li, H. Guo, Z. Wang, and Z. Zheng, "Unsupervised Fake News Detection Based on Autoencoder," *IEEE Access*, vol. 9, pp. 29 356–29 365, 2021. [Online]. Available: https://doi.org/10.1109/ACCESS.2021.3058809

[86] S. Cresci, R. Di Pietro, M. Petrocchi, A. Spognardi, and M. Tesconi, "A Fake Follower Story: Improving Fake Accounts Detection on Twitter," *IIT-CNR, Tech. Rep. TR-03*, 2014.

[87] Z. Jin, J. Cao, Y.-G. Jiang, and Y. Zhang, "News Credibility Evaluation on Microblog with a Hierarchical Propagation Model," in *2014 IEEE International Conference on Data Mining*, 2014, pp. 230–239. [Online]. Available: https://doi.org/10.1109/ICDM.2014.91

[88] G. L. Ciampaglia, P. Shiralkar, L. M. Rocha, J. Bollen, F. Menczer, and A. Flammini, "Computational Fact Checking from Knowledge Networks," *PLOS ONE*, vol. 10, no. 6, pp. 1–13, 06 2015. [Online]. Available: https://doi.org/10.1371/journal.pone.0128193

[89] P. Lendvai and U. Reichel, "Contradiction Detection for Rumorous Claims," in *Proceedings of the Workshop on Extra-Propositional Aspects of Meaning in Computational Linguistics (ExProM)*, E. Blanco, R. Morante, and R. Saurí, Eds. Osaka, Japan: The COLING 2016 Organizing Committee, Dec. 2016, pp. 31–40. [Online]. Available: https://aclanthology.org/W16-5004

[90] M. Hardalov, I. Koychev, and P. Nakov, "In Search of Credible News," in *Artificial Intelligence: Methodology, Systems, and Applications*, C. Dichev and G. Agre, Eds. Cham: Springer International Publishing, 2016, pp. 172–180. [Online]. Available: https://doi.org/10.1007/978-3-319-44748-3_17

[91] W. Ferreira and A. Vlachos, "Emergent: a Novel Data-Set for Stance Classification," *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, June 2016. [Online]. Available: https://eprints.whiterose.ac.uk/97416/

[92] C. L. Silverman, "Lies, Damn Lies and Viral Content," Tow Center for Digital Journalism, Columbia University, Tech. Rep., 2015. [Online]. Available: https://doi.org/10.7916/D8Q81RHH

[93] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou, and T. Mikolov, "FastText.zip: Compressing Text Classification Models," *CoRR*, vol. abs/1612.03651, 2016. [Online]. Available: http://arxiv.org/abs/1612.03651

[94] W. Y. Wang, ""Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, R. Barzilay and M.-Y. Kan, Eds. Vancouver, Canada:

Association for Computational Linguistics, jul 2017, pp. 422–426. [Online]. Available: https://aclanthology.org/P17-2067

[95] ——, "LIAR," 2017. [Online]. Available: https://www.cs.ucsb.edu/~william/data/liar_dataset.zip

[96] N. Ruchansky, S. Seo, and Y. Liu, "CSI: A Hybrid Deep Model for Fake News Detection," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, ser. CIKM '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 797–806. [Online]. Available: https://doi.org/10.1145/3132847.3132877

[97] Y. Long, Q. Lu, R. Xiang, M. Li, and C.-R. Huang, "Fake News Detection Through Multi-Perspective Speaker Profiles," in *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Taipei, Taiwan: Asian Federation of Natural Language Processing, Nov. 2017, pp. 252–256. [Online]. Available: https://aclanthology.org/I17-2043

[98] S. Volkova, K. Shaffer, J. Y. Jang, and N. Hodas, "Separating Facts from Fiction: Linguistic Models to Classify Suspicious and Trusted News Posts on Twitter," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, R. Barzilay and M.-Y. Kan, Eds. Vancouver, Canada: Association for Computational Linguistics, Jul. 2017, pp. 647–653. [Online]. Available: https://aclanthology.org/P17-2102

[99] Y. Wang, F. Ma, Z. Jin, Y. Yuan, G. Xun, K. Jha, L. Su, and J. 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*, ser. KDD '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 849–857. [Online]. Available: https://doi.org/10.1145/3219819.3219903

[100] Y. Liu and Y.-F. Wu, "Early Detection of Fake News on Social Media Through Propagation Path Classification with Recurrent and Convolutional Networks," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, Apr. 2018. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/11268

[101] M. Shrestha, "Detecting Fake News with Sentiment Analysis and Network Metadata," *Earlham College, Fall*, 2018. [Online]. Available: https://portfolios.cs.earlham.edu/wp-content/uploads/2018/12/Fake_News_Capstone.pdf

[102] F. Qian, C. Gong, K. Sharma, and Y. Liu, "Neural User Response Generator: Fake News Detection with Collective User Intelligence," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*. International Joint Conferences on Artificial Intelligence Organization, 7 2018, pp. 3834–3840. [Online]. Available: https://doi.org/10.24963/ijcai.2018/533

[103] S. Tschiatschek, A. Singla, M. Gomez Rodriguez, A. Merchant, and A. Krause, "Fake News Detection in Social Networks via Crowd Signals," in *Companion Proceedings of the The*

Web Conference 2018, ser. WWW ’18. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2018, p. 517–524. [Online]. Available: https://doi.org/10.1145/3184558.3188722

[104] K. Shu, D. Mahudeswaran, S. Wang, and H. Liu, “Hierarchical Propagation Networks for Fake News Detection: Investigation and Exploitation,” *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 14, no. 1, pp. 626–637, May 2020. [Online]. Available: https://ojs.aaai.org/index.php/ICWSM/article/view/7329

[105] K. Shu, D. Mahudeswaran, S. Wang, D. Lee, and H. Liu, “FakeNewsNet: A Data Repository with News Content, Social Context, and Spatiotemporal Information for Studying Fake News on Social Media,” *Big Data*, vol. 8, no. 3, pp. 171–188, 2020, pMID: 32491943. [Online]. Available: https://doi.org/10.1089/big.2020.0062

[106] “FakeNewsNet.” [Online]. Available: https://github.com/KaiDMML/FakeNewsNet

[107] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, “Geometric Deep Learning: Going Beyond Euclidean Data,” *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 18–42, 2017. [Online]. Available: https://doi.org/10.1109/MSP.2017.2693418

[108] F. Monti, F. Frasca, D. Eynard, D. Mannion, and M. M. Bronstein, “Fake News Detection on Social Media Using Geometric Deep Learning,” *CoRR*, vol. abs/1902.06673, 2019. [Online]. Available: http://arxiv.org/abs/1902.06673

[109] A. Drif, Z. Ferhat Hamida, and S. Giordano, “Fake News Detection Method Based on Text-Features,” *France, International Academy, Research, and Industry Association (IARIA)*, pp. 27–32, 2019.

[110] B. M. Amine, A. Drif, and S. Giordano, “Merging Deep Learning Model for Fake News Detection,” in *2019 International Conference on Advanced Electrical Engineering (ICAEE)*, 2019, pp. 1–4. [Online]. Available: https://doi.org/10.1109/ICAEE47123.2019.9015097

[111] M. Parvizimosaed, M. Esnaashari, A. Damia, and R. Bahmanyar, “Using Supervised Learning Models for Creating a New Fake News Analysis and Classification of a Covid-19 Dataset: A Case Study on Covid-19 in Iran,” in *2022 8th International Conference on Web Research (ICWR)*, 2022, pp. 152–155. [Online]. Available: https://doi.org/10.1109/ICWR54782.2022.9786244

[112] D. Lee, “Better Reasoning Behind Classification Predictions with BERT for Fake News Detection,” *CoRR*, vol. abs/2207.11562, 2022. [Online]. Available: http://arxiv.org/abs/2207.11562

[113] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, “Learning Deep Features for Discriminative Localization,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2016.319

[114] X. Li, P. Lu, L. Hu, X. Wang, and L. Lu, “A Novel Self-Learning Semi-Supervised Deep Learning Network to Detect Fake News on Social Media,” *Multimedia Tools and Applications*, vol. 81, 06 2022. [Online]. Available: https://doi.org/10.1007/s11042-021-11065-x

[115] S. Saha and T. Hasan, “Improving Classification Efficiency of Fake News Using Semi-Supervised Method,” 12 2022, PREPRINT (Version 1) available at Research Square. [Online]. Available: https://doi.org/10.21203/rs.3.rs-1201074/v1

[116] R. A. Frick, I. Vogel, and I. N. Grieser, “Fraunhofer SIT at Checkthat!-2022: Semi-Supervised Ensemble Classification for Detecting Check-Worthy Tweets,” in *Proceedings of the Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum, Bologna, Italy, September 5th - to - 8th, 2022*, ser. CEUR Workshop Proceedings, G. Faggioli, N. Ferro, A. Hanbury, and M. Potthast, Eds., vol. 3180. CEUR-WS.org, 2022, pp. 500–510. [Online]. Available: http://ceur-ws.org/Vol-3180/paper-39.pdf

[117] W. S. Paka, R. Bansal, A. Kaushik, S. Sengupta, and T. Chakraborty, “Cross-SEAN: A Cross-Stitch Semi-Supervised Neural Attention Model for Covid-19 Fake News Detection,” *Applied Soft Computing*, vol. 107, p. 107393, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1568494621003161

[118] P. Meel and D. K. Vishwakarma, “A Temporal Ensembling Based Semi-Supervised Convnet for the Detection of Fake News Articles,” *Expert Systems with Applications*, vol. 177, p. 115002, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417421004437

[119] ——, “Fake News Detection Using Semi-Supervised Graph Convolutional Network,” *CoRR*, vol. abs/2109.13476, 2021. [Online]. Available: https://arxiv.org/abs/2109.13476

[120] R. Bansal, W. S. Paka, Nidhi, S. Sengupta, and T. Chakraborty, “Combining Exogenous and Endogenous Signals with a Semi-Supervised Co-Attention Network for Early Detection of Covid-19 Fake Tweets,” in *Advances in Knowledge Discovery and Data Mining*, K. Karlapalem, H. Cheng, N. Ramakrishnan, R. K. Agrawal, P. K. Reddy, J. Srivastava, and T. Chakraborty, Eds. Cham: Springer International Publishing, 2021, pp. 188–200. [Online]. Available: https://doi.org/10.1007/978-3-030-75762-5_16

[121] R. Mansouri, M. Naderan-Tahan, and M. J. Rashti, “A Semi-Supervised Learning Method for Fake News Detection in Social Media,” in *2020 28th Iranian Conference on Electrical Engineering (ICEE)*, 2020, pp. 1–5. [Online]. Available: https://doi.org/10.1109/ICEE50131.2020.9261053

[122] P. M. Konkobo, R. Zhang, S. Huang, T. T. Minoungou, J. A. Ouedraogo, and L. Li, “A Deep Learning Model for Early Detection of Fake News on Social Media,” in *2020 7th International Conference on Behavioural and Social Computing (BESC)*, 2020, pp. 1–6. [Online]. Available: https://doi.org/10.1109/BESC51023.2020.9348311

[123] F. Morstatter, H. Dani, J. Sampson, and H. Liu, “Can One Tamper with the Sample API? Toward Neutralizing Bias from Spam and Bot Content,” in *Proceedings of the 25th International Conference Companion on World Wide Web*, ser. WWW ’16 Companion. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2016, p. 81–82. [Online]. Available: https://doi.org/10.1145/2872518.2889372

[124] F. Morstatter, L. Wu, T. H. Nazer, K. M. Carley, and H. Liu, “A New Approach to Bot Detection: Striking the Balance Between Precision and Recall,” in *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 2016, pp. 533–540. [Online]. Available: https://doi.org/10.1109/ASONAM.2016.7752287

[125] K. Lee, B. Eoff, and J. Caverlee, “Seven Months with the Devils: A Long-Term Study of Content Polluters on Twitter,” *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 5, no. 1, pp. 185–192, Aug. 2021. [Online]. Available: https://ojs.aaai.org/index.php/ICWSM/article/view/14106

[126] S. Cresci, R. Di Pietro, M. Petrocchi, A. Spognardi, and M. Tesconi, “DNA-Inspired Online Behavioral Modeling and Its Application to Spambot Detection,” *IEEE Intelligent Systems*, vol. 31, no. 5, pp. 58–64, 2016. [Online]. Available: https://doi.org/10.1109/MIS.2016.29

[127] S. Cresci, M. Petrocchi, A. Spognardi, and S. Tognazzi, “On the Capability of Evolved Spambots to Evade Detection via Genetic Engineering,” *Online Social Networks and Media*, vol. 9, pp. 1–16, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S246869641830065X

[128] D. Kosmajac and V. Keselj, “Twitter Bot Detection Using Diversity Measures,” in *Proceedings of the 3rd International Conference on Natural Language and Speech Processing*. Trento, Italy: Association for Computational Linguistics, Sep. 2019, pp. 1–8. [Online]. Available: https://aclanthology.org/W19-7401

[129] F. N. Pakaya, M. O. Ibrohim, and I. Budi, “Malicious Account Detection on Twitter Based on Tweet Account Features Using Machine Learning,” in *2019 Fourth International Conference on Informatics and Computing (ICIC)*, 2019, pp. 1–5. [Online]. Available: https://doi.org/10.1109/ICIC47613.2019.8985840

[130] Y. Wang, Y. Zhang, and B. Liu, “Sentiment Lexicon Expansion Based on Neural PU Learning, Double Dictionary Lookup, and Polarity Association,” in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 553–563. [Online]. Available: https://doi.org/10.18653/v1/D17-1059

[131] E. Ferrara and Z. Yang, “Quantifying the Effect of Sentiment on Information Diffusion in Social Media,” *PeerJ Computer Science*, vol. 1, 06 2015. [Online]. Available: https://doi.org/10.7717/peerj-cs.26

[132] S. Kudugunta and E. Ferrara, "Deep Neural Networks for Bot Detection," *Information Sciences*, vol. 467, pp. 312–322, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025518306248

[133] F. Wei and U. T. Nguyen, "Twitter Bot Detection Using Bidirectional Long Short-Term Memory Neural Networks and Word Embeddings," in *2019 First IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*, 2019, pp. 101–109. [Online]. Available: https://doi.org/10.1109/TPS-ISA48467.2019.00021

[134] C. Zhao, Y. Xin, X. Li, H. Zhu, Y. Yang, and Y. Chen, "An Attention-Based Graph Neural Network for Spam Bot Detection in Social Networks," *Applied Sciences*, vol. 10, no. 22, 2020. [Online]. Available: https://www.mdpi.com/2076-3417/10/22/8160

[135] M. BalaAnand, N. Karthikeyan, S. Karthik, R. Varatharajan, G. Manogaran, and C. B. Sivaparthipan, "An Enhanced Graph-Based Semi-Supervised Learning Algorithm to Detect Fake Users on Twitter," *The Journal of Supercomputing*, vol. 75, no. 9, pp. 6085–6105, Sep 2019. [Online]. Available: https://doi.org/10.1007/s11227-019-02948-w

[136] E. Shaabani, A. Sadeghi-Mobarakeh, H. Alvari, and P. Shakarian, "An End-to-End Framework to Identify Pathogenic Social Media Accounts on Twitter," *2019 2nd International Conference on Data Intelligence and Security (ICDIS)*, pp. 128–135, 2019. [Online]. Available: https://doi.org/10.48550/arxiv.1905.01553

[137] R. Alharthi, A. Alhothali, and K. Moria, "Detecting and Characterizing Arab Spammers Campaigns in Twitter," *Procedia Computer Science*, vol. 163, pp. 248–256, 2019, 16th Learning and Technology Conference 2019Artificial Intelligence and Machine Learning: Embedding the Intelligence. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050919321453

[138] Q. Guo, H. Xie, Y. Li, W. Ma, and C. Zhang, "Social Bots Detection via Fusing BERT and Graph Convolutional Networks," *Symmetry*, vol. 14, no. 1, 2022. [Online]. Available: https://www.mdpi.com/2073-8994/14/1/30

[139] J. Rodríguez-Ruiz, J. I. Mata-Sánchez, R. Monroy, O. Loyola-González, and A. López-Cuevas, "A One-Class Classification Approach for Bot Detection on Twitter," *Computers & Security*, vol. 91, p. 101715, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404820300031

[140] H. Gao, Y. Chen, K. Lee, D. Palsetia, and A. N. Choudhary, "Towards Online Spam Filtering in Social Networks." in *The Network and Distributed System Security (NDSS) Symposium 2012*, vol. 12, no. 2012, 2012, pp. 1–16. [Online]. Available: https://www.ndss-symposium.org/ndss2012/ndss-2012-programme/towards-online-spam-filtering-social-networks

[141] C. Yang, R. Harkreader, and G. Gu, "Empirical Evaluation and New Design for Fighting Evolving Twitter Spammers," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 8, pp. 1280–1293, 2013. [Online]. Available: https://doi.org/10.1109/TIFS.2013.2267732

[142] C. A. Davis, O. Varol, E. Ferrara, A. Flammini, and F. Menczer, “BotOrNot: A System to Evaluate Social Bots,” in *Proceedings of the 25th International Conference Companion on World Wide Web*, ser. WWW ’16 Companion. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2016, p. 273–274. [Online]. Available: https://doi.org/10.1145/2872518.2889302

[143] M. Alsaleh, A. Alarifi, A. M. Al-Salman, M. Alfayez, and A. Almuhaysin, “TSD: Detecting Sybil Accounts in Twitter,” in *2014 13th International Conference on Machine Learning and Applications*, 2014, pp. 463–469. [Online]. Available: https://doi.org/10.1109/ICMLA.2014.81

[144] O. Varol, E. Ferrara, C. Davis, F. Menczer, and A. Flammini, “Online Human-Bot Interactions: Detection, Estimation, and Characterization,” *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 11, no. 1, pp. 280–289, May 2017. [Online]. Available: https://ojs.aaai.org/index.php/ICWSM/article/view/14871

[145] F. Ahmed and M. Abulaish, “A Generic Statistical Approach for Spam Detection in Online Social Networks,” *Computer Communications*, vol. 36, p. 1120–1129, 06 2013. [Online]. Available: https://doi.org/10.1016/j.comcom.2013.04.004

[146] Z. Miller, B. Dickinson, W. Deitrick, W. Hu, and A. H. Wang, “Twitter Spammer Detection Using Data Stream Clustering,” *Information Sciences*, vol. 260, pp. 64–73, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025513008037

[147] A. Derhab, R. Alawwad, K. Dehwah, N. Tariq, F. A. Khan, and J. Al-Muhtadi, “Tweet-Based Bot Detection Using Big Data Analytics,” *IEEE Access*, vol. 9, pp. 65 988–66 005, 2021. [Online]. Available: https://doi.org/10.1109/ACCESS.2021.3074953

[148] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, “TensorFlow: a System for Large-Scale Machine Learning,” in *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*, ser. OSDI’16. USA: USENIX Association, 2016, p. 265–283. [Online]. Available: https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf

[149] J. Han and C. Moraga, “The Influence of the Sigmoid Function Parameters on the Speed of Backpropagation Learning,” in *From Natural to Artificial Neural Computation*, J. Mira and F. Sandoval, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1995, pp. 195–201. [Online]. Available: https://doi.org/10.1007/3-540-59497-3_175

[150] S. Cresci, R. Di Pietro, M. Petrocchi, A. Spognardi, and M. Tesconi, “The Paradigm-Shift of Social Spambots: Evidence, Theories, and Tools for the Arms Race,” in *Proceedings of the 26th International Conference on World Wide Web Companion*, ser. WWW ’17 Companion.

Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2017, p. 963–972. [Online]. Available: https://doi.org/10.1145/3041021.3055135

[151] E. Kajan, N. Faci, Z. Maamar, M. Sellami, E. Ugljanin, H. Kheddouci, D. Stojanovic, and D. Benslimane, "Real-Time Tracking and Mining of Users' Actions over Social Media," *Computer Science and Information Systems*, vol. 17, pp. 403–426, 2020. [Online]. Available: https://zuscholars.zu.ac.ae/works/2886

[152] X. Zhou and R. Zafarani, "A Survey of Fake News: Fundamental Theories, Detection Methods, and Opportunities," *ACM Comput. Surv.*, vol. 53, no. 5, sep 2020. [Online]. Available: https://doi.org/10.1145/3395046

[153] L. Azevedo, M. D'aquin, B. Davis, and M. Zarrouk, "LUX (Linguistic aspects Under eXamination): Discourse Analysis for Automatic Fake News Classification," in *The Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2021)*, ser. Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021. Online, France: Association for Computational Linguistics, Aug. 2021, pp. 41–56. [Online]. Available: https://hal.science/hal-03659147

[154] P. Nakov, G. Da San Martino, T. Elsayed, A. Barrón-Cedeño, R. Míguez, S. Shaar, F. Alam, F. Haouari, M. Hasanain, N. Babulkov, A. Nikolov, G. K. Shahi, J. M. Struß, and T. Mandl, "The CLEF-2021 CheckThat! Lab on Detecting Check-Worthy Claims, Previously Fact-Checked Claims, and Fake News," in *Advances in Information Retrieval*, D. Hiemstra, M.-F. Moens, J. Mothe, R. Perego, M. Potthast, and F. Sebastiani, Eds. Cham: Springer International Publishing, 2021, pp. 639–649. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-72240-1_75

[155] J. W. Chotlos, "IV. a Statistical and Comparative Analysis of Individual Written Language Samples," *Psychological Monographs*, vol. 56, p. 75–111, 1944. [Online]. Available: https://doi.org/10.1037/h0093511

[156] M. Templin, *Certain Language Skills in Children: Their Development and Interrelationships*. Minneapolis, MN: University of Minnesota Press, 1957. [Online]. Available: https://www.jstor.org/stable/10.5749/j.ctttv2st

[157] P. Lissón and N. Ballier, "Investigating Lexical Progression Through Lexical Diversity Metrics in a Corpus of French L3," *Discours. Revue de linguistique, psycholinguistique et informatique. A journal of linguistics, psycholinguistics and computational linguistics*, no. 23, 2018. [Online]. Available: https://doi.org/10.4000/discours.9950

[158] N. Chipere, D. Malvern, and B. Richards, "Using a Corpus of Children's Writing to Test a Solution to the Sample Size Problem Affecting Type-Token Ratios," in *Corpora and language learners*, G. Aston, S. Bernadini, and D. Stewart, Eds. John Benjamins, 2004, pp. 139–147. [Online]. Available: https://doi.org/10.1075/scl.17.10chi

[159] K. Kettunen, "Can Type-Token Ratio be Used to Show Morphological Complexity of Languages?" *Journal of Quantitative Linguistics*, vol. 21, no. 3, pp. 223–245, 2014. [Online]. Available: https://doi.org/10.1080/09296174.2014.911506

[160] H. Heaps, *Information Retrieval: Computational and Theoretical Aspects*. New York: Academic Press, Inc., 1978.

[161] P. M. MacCarthy, "An Assessment of the Range and Usefulness of Lexical Diversity Measures and the Potential of the Measure of Textual, Lexical Diversity," Ph.D. dissertation, University of Memphis, Tennessee, USA, 2005.

[162] P. M. McCarthy and S. Jarvis, "MTLD, vocd-D, and HD-D: A Validation Study of Sophisticated Approaches to Lexical Diversity Assessment," *Behavior Research Methods*, vol. 42, no. 2, pp. 381–392, May 2010. [Online]. Available: https://doi.org/10.3758/BRM.42.2.381

[163] R. Koizumi, "Relationships Between Text Length and Lexical Diversity Measures: Can We Use Short Texts of Less Than 100 Tokens?" *Vocabulary Learning and Instruction*, vol. 1, no. 1, pp. 60–69, 2012. [Online]. Available: https://doi.org/10.7820/vli.v01.1.koizumi

[164] M. A. Covington, "MATTR User Manual," University of Georgia Artificial Intelligence Center, Tech. Rep., 2007. [Online]. Available: https://esploro.libs.uga.edu/esploro/outputs/9949316127802959

[165] M. A. Covington and J. D. McFall, "Cutting the Gordian Knot: The Moving-Average Type–Token Ratio (MATTR)," *Journal of Quantitative Linguistics*, vol. 17, no. 2, pp. 94–100, 2010. [Online]. Available: https://doi.org/10.1080/09296171003643098

[166] G. Fergadiotis, H. H. Wright, and T. M. West, "Measuring Lexical Diversity in Narrative Discourse of People with Aphasia," *American Journal of Speech-Language Pathology*, vol. 22, no. 2, pp. S397–S408, 2013. [Online]. Available: https://pubs.asha.org/doi/abs/10.1044/1058-0360%282013/12-0083%29

[167] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely Randomized Trees," *Machine Learning*, vol. 63, no. 1, pp. 3–42, Apr 2006. [Online]. Available: https://doi.org/10.1007/s10994-006-6226-1

[168] Y. Le Cun and F. Fogelman-Soulié, "Modèles connexionnistes de l'apprentissage," *Intellectica*, no. 2-3, pp. 114–143, 1987. [Online]. Available: https://www.persee.fr/doc/intel_0769-4113_1987_num_2_1_1804

[169] H. Bourlard and Y. Kamp, "Auto-Association by Multilayer Perceptrons and Singular Value Decomposition," *Biological Cybernetics*, vol. 59, no. 4, pp. 291–294, Sep 1988. [Online]. Available: https://doi.org/10.1007/BF00332918

[170] G. E. Hinton and R. Zemel, "Autoencoders, Minimum Description Length and Helmholtz Free Energy," in *Advances in Neural Information Processing Systems*, J. Cowan, G. Tesauro, and

J. Alspector, Eds., vol. 6. Morgan-Kaufmann, 1993. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/1993/file/9e3cfc48eccf81a0d57663e129aef3cb-Paper.pdf

[171] K. Lopyrev, "Generating News Headlines with Recurrent Neural Networks," 2015.

[172] A. Kulkarni and A. Shivananda, *Natural Language Processing Recipes: Unlocking Text Data with Machine Learning and Deep Learning Using Python*. Springer, 2019. [Online]. Available: https://doi.org/10.1007/978-1-4842-4267-4

[173] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How Transferable are Features in Deep Neural Networks?" in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, Eds., vol. 27. Curran Associates, Inc., 2014. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2014/file/375c71349b295fbe2dcdca9206f20a06-Paper.pdf

[174] S. Kalyoncu, A. Jamil, E. Karataş, J. Rasheed, and C. Djeddi, "Stock Market Value Prediction Using Deep Learning," *Data Science and Applications*, vol. 3, no. 2, pp. 10–14, 2020. [Online]. Available: https://www.jdatasci.com/index.php/jdatasci/article/view/42

[175] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, "On the Properties of Neural Machine Translation: Encoder-Decoder Approaches," *CoRR*, vol. abs/1409.1259, 2014. [Online]. Available: http://arxiv.org/abs/1409.1259

[176] C. Xiong, S. Merity, and R. Socher, "Dynamic Memory Networks for Visual and Textual Question Answering," in *Proceedings of The 33rd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. F. Balcan and K. Q. Weinberger, Eds., vol. 48. New York, New York, USA: PMLR, 20–22 Jun 2016, pp. 2397–2406. [Online]. Available: https://proceedings.mlr.press/v48/xiong16.html

[177] S. Cresci, "MIB Datasets," http://mib.projects.iit.cnr.it/dataset.html, 2017, accessed: 2021-01-12.

[178] K.-C. Yang, O. Varol, C. A. Davis, E. Ferrara, A. Flammini, and F. Menczer, "Arming the Public with Artificial Intelligence to Counter Social Bots," *Human Behavior and Emerging Technologies*, vol. 1, no. 1, pp. 48–61, 2019. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/hbe2.115

[179] G. Fergadiotis, H. H. Wright, and S. B. Green, "Psychometric Evaluation of Lexical Diversity Indices: Assessing Length Effects," *Journal of Speech, Language, and Hearing Research*, vol. 58, no. 3, pp. 840–852, 2015. [Online]. Available: https://pubs.asha.org/doi/abs/10.1044/2015_JSLHR-L-14-0280

[180] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and Édouard Duchesnay, "Scikit-Learn: Machine Learning in Python," *Journal*

of Machine Learning Research, vol. 12, no. 85, pp. 2825–2830, 2011. [Online]. Available: http://jmlr.org/papers/v12/pedregosa11a.html

[181] F. Chollet, "Keras: Theano-Based Deep Learning Library," Code: https://github.com/fchollet. Documentation: https://keras.io/, 2015.

[182] P. Baldi, S. Brunak, Y. Chauvin, C. A. F. Andersen, and H. Nielsen, "Assessing the Accuracy of Prediction Algorithms for Classification: an Overview," *Bioinformatics*, vol. 16, no. 5, pp. 412–424, 05 2000. [Online]. Available: https://doi.org/10.1093/bioinformatics/16.5.412

[183] D. M. W. Powers, "Evaluation: from Precision, Recall and F-Measure to Roc, Informedness, Markedness and Correlation," *CoRR*, vol. abs/2010.16061, 2020. [Online]. Available: https://arxiv.org/abs/2010.16061

[184] M. Sayyadiharikandeh, O. Varol, K.-C. Yang, A. Flammini, and F. Menczer, "Detection of Novel Social Bots by Ensembles of Specialized Classifiers," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, ser. CIKM '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 2725–2732. [Online]. Available: https://doi.org/10.1145/3340531.3412698