PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA

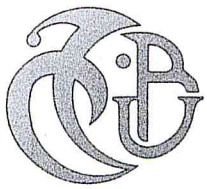
Ministry of Higher Education and Scientific Research

Saad DAHLEB University of Blida

Faculty of Sciences

Computer Science Department

Specialty: Software Engineering



MASTER'S THESIS

Identifying Depression in Tweets using Deep Learning

Author: Ibrahim Cheurfa

Supervisor: Fatima Boumahdi, University of Blida

President of the Jury: Amina Madani, University of Blida

Member of the Jury: Imene Cherfa, University of Blida

Academic Year: 2017/2018

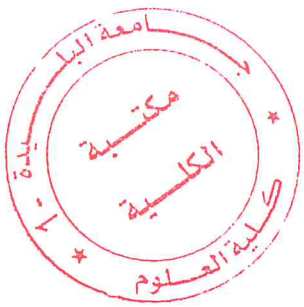
Acknowledgements

I would first like to thank my thesis advisor Mrs. Fatima Boumahdi for first, believing in me and trusting my capabilities, and second, for all her support, patience, and help during the whole duration of the research, and even before.

I would also like to thank the teachers who offered their help and the experts that have paved the way for this project to become a reality through their tremendous efforts to contribute to the computer science community.

Finally, I must express my very profound gratitude to my parents and to my best friend Hibat El Rahmene for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Ibrahim Cheurfa



Abstract

Mental health is considered as one of today's world's most prominent plagues. Therefore, our work aims to use the potential of social media platforms to solve one of mental health's biggest issues, which is depression identification. We propose a new deep learning model that we train on a depression-dedicated dataset in order to detect such mental illness from an individual's tweets. Our main contributions with this work lie in the three following points: (1) We trained our own word embeddings using a depression-dedicated dataset. (2) We combined a CNN model with the MSA model in order to improve the feature extraction process and enhance the model's performance. (3) We analyzed through different experiments the performance of three deep learning models in order to provide more perspectives and insights for depression researches. Our model achieved a 99% accuracy, outperforming any statistical or deep learning models found in literature currently.

Keywords: Sentiment Analysis, Deep Learning, Mental Health, Text Classification.

Table of Content

List of Figures

List of Tables

Introduction

Context

The aim of this study is to use artificial intelligence's deep learning techniques to overcome one of mental health's most prominent challenges, which is identifying depression automatically from an individual's behavior. We use tweets as a medium to track such behavior due to the thought-expression culture of the platform and the availability of the data.

Problematic

Despite the efforts invested in it, mental health still remains one of the most life-threatening health issues in the world. What makes such matter worth more attention is the threats that comes from neglecting it. Research has shown that individuals suffering from one or more of mental illnesses will likely experience a snowball effect towards other disorders, leading to life-degrading consequences, and in some cases, to fatal ones.

One of the most popular mental disorders amongst the world's populations is Major Depressive Disorder, commonly known as clinical depression, with nearly 300 million individuals suffering from it globally [127]. Studies have shown that 3 – 5% of males and 8 – 10% of female from the total world's population are likely to experience a major depressive episode within a period of one year [127]. What makes depression the most known disorder in the world, is its likelihood of being triggered by other health issues as it often co-occurs with other illnesses and mental conditions. With that being said, it has been reported that it's one the major causes of suicide, something that shows why it deserves more attention.

The reason why depression is considered as life-threatening, is because of its methods of identification. Diagnosis is extracted from the patient's self-reported experiences, behavior questionnaires, and surveys, which makes it prone to manipulation. Moreover, individuals suffering from depression tend to hide what they're going through and never seek out for help in most of the cases, something that can cause their state to worsen, and sometimes lead to suicide.

Objectives

Once identified and treated, depression has been proven to cured. Therefore, our main objective with this study is to propose a new way for identifying depression, a way that is based on concrete data and tracked natural behavior. For that, we strive to use one of today's most advanced technologies, which is deep learning.

With that being said, since people are using social media more and more to express their feelings and share their innermost thoughts and desires, we take Twitter as a source of data, as it records people's self expressions in their most naturalistic way.

Thesis structure

This thesis is composed of 5 chapters:

- The first chapter introduces sentiment analysis, its domains of application and related fields, its different types, and finally some of its most used techniques.
- In the second chapter, we start by presenting deep learning and its basic notions, after that, we move into the different deep learning techniques used in sentiment analysis.
- The third chapter is dedicated to depression detection related works that tackled the same or a similar problematic as ours.
- In the fourth chapter we present our proposed model and we explore each of its layers while explaining its origins and the different models it is built upon.
- In the last chapter, we go through the step-by-step process that we followed to build, train, and evaluate our model. By the end we share the results we got and we compare it to other works that used the same dataset we did.

Finally, we will end our thesis with a general conclusion, in which we will share the key learning points of this research initiative and future perspectives for our proposed model and how can it be improved and extended to possibly provide better results.

Part I: Literature Review

Chapter 1 Sentiment Analysis

1.1 Introduction

In our daily life, our behaviors and decisions are highly influenced by other people's' opinions, in Social Psychology this phenomenon is called social compliance. Whether consciously or not, every choice we make is based upon a previous opinion. Our beliefs and perceptions of reality are shaped according to what the world has influenced us, hence, everything we come across has a direct or an indirect influence on our actions.

With that being said, and with the birth of web 2.0, people began to express their opinions freely, publicly, and in different forms, something that increased the level of interest of organizations and companies in such data because of its precious value. Such insights can be very profitable to businesses and have a great political and economical impact on society. From that, a new research field birthed and became the center of attention of computer science's research community, and it is called *sentiment analysis*.

In this chapter, we will introduce sentiment analysis (also known as opinion mining) and its application domains, its different levels and methods, and we will conclude with the main challenges faced in this field.

1.2 Sentiment Analysis and Opinion Mining

Before moving to its scientific definition, let's analyze the word "sentiment analysis" itself and see its literal definition. In the dictionary, 'sentiment' is defined as *a thought, opinion, or idea based on a feeling about a situation, or a way of thinking about something* [1] and 'analysis' is defined as *the act of analysing something* [2]. If we combine the two definitions, we can define sentiment analysis as the act of analysing thoughts, opinions, or ideas that are based on feelings, or way of thinking.

Bing Liu, a pioneer in the field, defines sentiment analysis as the following: *Sentiment analysis, also called opinion mining, is the field of study that analyzes people's opinions, sentiments,*

evaluations, appraisals, attitudes, and emotions towards entities such as products, services, organizations, individuals, issues, events, topics, and their attributes [3].

Although researchers use more often the term "sentiment analysis" than "opinion mining", they can be used both interchangeably because they basically refer to the same field of study. Both terms appeared first around 2003 [4][5] but the research on this field started around 2000 [6][7][8][9][10][11] and since then, there has been many publications addressing the subject without really giving it a name [3].

1.2.1 Sentiment Analysis Applications

We, humans, tend to think that we are free in our choices and decisions, but truth is, every decision we make is highly influenced by others' opinions, consciously or unconsciously. With such pivotal importance, opinions become the nervous system of our personal decision support organism, something that caught the attention of many researchers, businesses and entities whose survival and profit are influenced by the public's opinion, that's why sentiment analysis and opinion mining are being applied in various business, political, and social domains. Companies for example, are organizations that survive on their customers and the people they work with, and with the explosion of social media, it became possible to find the thoughts and ideas of these individuals about brands, products, and services, often publically. So the field of sentiment analysis opened new possibilities for these institutions because they started seeing themselves through the eyes of their users and customers.

As mentioned above, due to its relatedness and importance to every area of our lives, Sentiment Analysis is being applied in many domains. After going through different research papers and seeing previous works in the field, we realized that most of sentiment analysis applications fall under the following 2 categories: detection and prediction.

1.2.1.1 Detection

The first category of sentiment analysis is detection. What I mean by that, is that it is being used to detect a problem, a disease, or a need, depending on the industry it's being applied in. These applications are either to study certain phenomenons or to understand something that happened in the world and find the reasons behind it, that's why they mainly focus on past and present events. The following are some examples about applications that fall under this category:

- In [12], reviews were used to rank products and merchants.

- In [13], the relationships between the NFL betting line and public opinions in blogs and Twitter were studied.
- In [14], the authors studied political standpoints.
- In [15], sentiment flow in social networks was investigated.
- In [16], Twitter sentiment was linked with public opinion polls.
- In [17], sentiments in mails were used to find how genders differed on emotional axes.
- In [18], emotions in novels and fairy tales were tracked.
- In [19] and [20], expert investors in microblogs were identified and sentiment analysis of stocks was performed.
- In [21], social influences in online book reviews were studied.
- In [22], sentiment analysis was used to characterize social relations.

Of course, there are many other applications that fall under this category but these were the main ones that caught our attention and that we wanted to highlight in order to showcase the most important areas of applications. We tried to select the most relevant ones in each field to avoid repetition.

1.2.1.2 Prediction

Although detection plays a very important part of prediction applications, this category is distinguished by its emphasis on using detected opinions to predict future results and obviously that's something that is highly needed in the business and political world, and even in health. Here are few examples that belong to this category:

- In [23], Twitter sentiment was also applied to predict election results.
- In [24], a method was reported for predicting comment volumes of political blogs.
- In [25], a sentiment model was proposed to predict sales performance.
- In [26], [27] and [28], Twitter data, movie reviews, and blogs were used to predict box-office revenues for movies.

- In [29], Twitter moods were used to predict the stock market.
- In [30], blog and news sentiment was used to study trading strategies.

Although it started mainly on written language, sentiment analysis is being used nowadays on visual and auditory data as well. With that being said, we noticed that it has different tasks that fall under it, such as subjectivity analysis, affect analysis, review mining… etc, but in general, sentiment analysis itself is a sub-field of Natural Language Processing (NLP) and even Data Mining and it emerged lately to become one of the most active research areas in those fields because it represents a large problem space.

1.2.2 Background

Throughout this chapter we will be mentioning a lot of fields that are related to sentiment analysis in a way or another, but the two most important ones needed to get a better understanding of the rest of the chapter are NLP and Text Mining. Therefore this section aims to give a brief introduction to these fields to facilitate the comprehension of the matters shared by the end of the chapter.

1.2.2.1 Natural Language Processing

With the increasing demand of machine-to-human understanding, comprehending humans' natural language became a necessity in order to facilitate human-to-machine and machine-to-human interactions. From that, a new field of study called Natural Language Processing emerged and became the center of attention of researchers. Being the intersection between linguistics and computer science, its main purpose is to allow machines *to analyze, understand, and derive meaning from human language in a smart and useful way* [31]. Using NLP, tasks such as translation, speech recognition and topic extraction became a reality for developers.

Although it has a huge share of the research community, it is still categorized as one of computer science's hardest problems due to the difficulty of interpretation of human language by machines. Language characteristics such as figures of speech, metaphors, and sarcasm, are hard information to grasp by machines that were designed to understand binary code in the first place. The main challenge is to understand the concepts and contexts of the language and not only the words themselves.

With that being said, sentiment analysis can be considered as a sub-field of NLP because it deals with language processing in order to discover the polarity of opinions. Even though it started being applied lately to other data formats, sentiment analysis was and still is mainly applied to text, and fortunately, as soon as this field emerged it solved some of NLP's famous challenges, yet there are still more that we will cover by the end of this chapter.

1.2.2.2 Text Mining

Text mining is defined as the discovery of new information by automatically extracting information from a large amount of various unstructured textual resources [32]. Valuable insights can be discovered in text-based content using text mining. Documents, emails, social media posts, became all a potential source of hidden information [33]. The way it works, it transforms words and phrases into numerical values that are structured in a database, then applies data mining techniques on it.

1.3 Levels of Sentiment Analysis

1.3.1 Document Level

Assuming that a document expresses sentiments on one entity only, document-level sentiment analysis focuses on classifying the document as a whole if it states a positive or negative opinion about the subject matter [8][10].

1.3.2 Sentence Level

Unlike the previous level, sentence-level sentiment analysis emphasis on analyzing sentences instead of seeing the document as one that's expressing opinion about some entity, which gives us the ability to detect different sentiments on the same document, about the same or different entities. The main task at this level is to detect if a positive, negative, or neutral opinion is being expressed in each sentence. Neutral usually means no opinion. With that being said, note that it is related to subjectivity and objectivity classification which makes it closely related to Wiebe's subjectivity classification work [34], however, it's not exactly the same thing because subjectivity and sentiment are two different concepts and objectivity itself can express opinion.

1.3.3 Entity and Aspect Level

As you can notice, in the previous levels we are not able to detect exactly what people liked and did not like but we only check if their opinion is positive or not, that's why the aspect-level, previously called feature-level [35], emphasis more on the opinion itself instead of focusing at

language constructs (documents, paragraphs, sentences… etc). Its core idea is that an opinion can be decomposed into a sentiment (positive or negative) and a target (of opinion). The fact of having an opinion and detecting its polarity doesn't provide us with valuable information unless we know its exact target.

1.4 Sentiment Analysis Methods

With the rise of sentiment analysis to popularity comes different studies, projects, and new onctributions constantly. Many sentiment analysis methods and techniques have been developed since its beginning, therefore, in this section, we will be sharing some of the most popular frameworks and methods used in the field [36].

1.4.1 Emoticons

Due to their popularity of use amongst almost all categories of ages, emoticons became one of the most important data through which we can detect sentiment. It is also considered as one of the easiest ways to detect polarity because just like sentiment words, emoticons express usually either a positive feeling or a negative one. Polarity is usually extracted by using a set of common emoticons [37]. These sets include the popular variations that express the primary polarities. With that being said, emoticons can also present a challenge when used inside sentences or in other contexts such as sarcasm, therefore, it is mainly used in combination with other techniques.

1.4.2 Linguistic Inquiry and Word Count

Linguistic Inquiry and Word Count (LIWC) is a text analysis tool [38]. Given a text, the latter uses a dictionary containing words and their classification categories to evaluate emotional, cognitive, and structural components. What distinguishes LIWC from other techniques is its ability to classify the given text in various sentiment categories, a single word can belong to different categories simultaneously, which gives more precision to sentiment detection.

The LIWC software gives a certain autonomy to its users by giving them the ability to include customized dictionaries instead of the standard ones. In order to measure polarity, it examines the relative rate of positive and negative effects in the feeling categories.

1.4.3 SentiStrength

SentiStrength is a technique that implements a combination of machine-learning-based methods [39]. The core classification of it relies on the set of words in the LIWC dictionary [38] with some

added features dedicated for Online Social Networks (OSN) context. These features basically include a list of negative and positive words with a list of boosters (such as "very" to strengthen and "somewhat" to weaken). The use of emoticons and repeated punctuations was used for the same purpose as well.

1.4.4 SentiWordNet

Famously used in opinion mining, SentiWordNet [40] is based on the English lexical dictionary WordNet [41]. The latter groups adjectives, nouns, verbs, and other grammatical classes into synonym sets called *synsets*. These synsets are used with WordNet to associate three scores that indicate the sentiment of the text: positive, negative, and objective (neutral). These scores are obtained using a semi-supervised machine learning method.

1.4.5 SenticNet

SentiNet [42] explores artificial intelligence and semantic Web techniques to infer the polarity of common sense concepts from natural language text at a semantic level, rather than a syntactic one. It is able to create a polarity for nearly 14,000 concepts by simply using Natural Language Processing (NLP) techniques.

1.4.6 Happiness Index

Happiness Index [43] is a sentiment scale that uses the Affective Norms for English Words (ANEW) [44], which is *a collection of 1,034 words commonly used associated with their affective dimensions of valence, arousal, and dominance*. Constructed based on the latter, Happiness Index gives scores for a given text between 1 and 9, specifying the amount of happiness expressed in that text.

1.4.7 PANAS-t

PANAS-t is a psychometric scale that aims to detect mood fluctuations of Twitter users. It consists of an adapted version of the famous psychology method known as Positive Affect Negative Affect Scale (PANAS) [45]. *PANAS-t is based on a large set of words associated with eleven moods: joviality, assurance, serenity, surprise, fear, sadness, guilt, hostility, shyness, fatigue, and attentiveness*. One of its key advantages is its ability to track the increase or decrease in sentiments over time.

1.5 Sentiment Analysis Issues and Challenges

1.5.1 Sentiment Lexicon and its Issues

In order to detect sentiment, there should be an indicator that helps us figure it out, for that we have what we call *sentiment words* or *opinion words*. Words such as: good, awesome, disgusting, awful, are words that are often used to express positive or negative sentiments. In addition to individual words, common expressions and idioms play a very important role in expressing opinions. Examples can be: Under the weather, pull yourself together, once in a blue moon… etc.

Due to the importance of the latter words and phrases, researchers designed a wide variety of algorithms to compile and generate such list of inputs reuniting them in what is called a sentiment lexicon (or opinion lexicon) to facilitate the work of applications in the field, yet it is still insufficient because the the following challenges [3]:

- For some words, such as certain homophones, the same word can express positive or negative sentiment depending on which context it is used. For instance, "killed" usually indicates negative sentiment, e.g., "I wish they killed me when I was born." but it can also imply positive sentiment, e.g., "I won that challenge! I'm the best! I killed it!"

- Sentences that contain sentiment words do not necessarily express a sentiment. This usually applies in interrogative or conditional sentences. With that being said, note that not all sentences that fall under those categories are empty of sentiment (do not express sentiment). Examples can be: "Which chocolate tastes good?" and "If I find a good one, I will buy it." Both these sentences contain the sentiment word "good," but neither expresses a positive or negative opinion on the subject of the sentence. However, not all conditional sentences or interrogative sentences express no sentiments, e.g., "Can you help me sell this aweful carpet?" and "If you want an awesome experience, go on a student exchange program."

- Sarcasm still remains as one of the main challenges of sentiment analysis because sarcastic sentences, whether they contain sentiment words or not, requires human understanding and are hard to deal with. For example: "What an amazing phone! It got broken from the first fall." Sarcasms are not so common in consumer reviews about products and services, but are very common in political discussions, which make political opinions hard to deal with.

- Last but not least, objective sentences present another challenge to the field because they don't usually contain sentiment words yet they express sentiment. Usually sentences that focus on the features of a product or facts in general fall under this category.This sentence: "This app consumes a lot of memory." implies a negative sentiment about the app since it uses a lot of memory, yet it is objective, as it states a fact, and it has no sentiment words.

1.5.2 Natural Language Processing Issues

As mentioned above, sentiment analysis is definitely a sub-field of NLP, therefore, all NLP problems, which most of them are not solved yet, appliy to it and bring more difficulty to the field [3]. Some of these problems are, coreference resolution, negation handling, and word sense disambiguation. However, it is also useful to realize that sentiment analysis is a highly restricted NLP problem because the system does not need to fully understand the semantics of each sentence or document but only needs to understand some aspects of it, i.e., positive or negative sentiments and their target entities or topics. In this sense, sentiment analysis offers a great platform for NLP researchers to make tangible progresses on all fronts of NLP with the potential of making a huge practical impact.

Researchers now also have a much better understanding of the whole spectrum of the problem, its structure, and core issues. Numerous new (formal) models and methods have been proposed. The research has not only deepened but also broadened significantly. Earlier research in the field mainly focused on classifying the sentiment or subjectivity expressed in documents or sentences, which is insufficient for most real-life applications.

1.6 Conclusion

In this chapter, we went through the basics of sentiment analysis and opinion mining. We presented different works in the field and we explored each of its type and some of its popular techniques. By the end we discussed the major obstacles that are faced in the field.

In the next chapter we will introduce deep learning and go through its basics and most relevant works to our study.

Chapter 2 Deep Learning

2.1 Introduction

Ever since the invention of programmable computers, humans have been dreaming about robots and machines that are able to think and do day-to-day activities to facilitate their lives. Today, with Artificial Intelligence (AI), such dreams became a reality and the field became one the most popular research fields that is thriving day after day, while challenging all previous barriers and setting a new bar for the future of mankind.

From being able to solve some of the most challenging intellectual problems to beating chess masters, early AI focused more on solving problems that can be described by a list of formal or mathematical rules, hence, the true challenge was to reach that human-like intelligence that allows machines to perform more abstract concepts that come naturally for humans, such as speech or facial recognition, something that is a reality nowadays.

In this chapter, we will talk about Deep Learning, one of AI's most popular research fields and we will go through its different classes and techniques with real-life application examples.

2.2 Definitions

Before we move any further, we start first by defining the most important terms that we will be using during the rest of the chapter in order facilitate the understanding of the next sections.

2.2.1 Artificial Intelligence

First, although the term Artificial Intelligence might seem self-explanatory, plenty of definitions exist and different debates are still being held about it. We will be using Poole, Mackworth & Goebel's definition in our work, as it refers to AI as Computational intelligence and defines it as follows: *Computational intelligence is the study of the design of intelligent agents* [46].

The term agent here can refer to anything that acts in a specific environment, anything that does something, it can be humans, animal, machines… etc. An intelligent agent on the other hand is a system that acts intelligently. As these researchers explain it: *What it does is appropriate for its circumstances and its goal, it is flexible to changing environments and changing goals, it learns from*

experience, and it makes appropriate choices given perceptual limitations and finite computation [46].

Scientifically speaking, AI's central goal is to understand the principles that make intelligent behavior possible regardless of the type of environment, and from the engineering perspective, its main focus is how to specify methods for the design of intelligent artifacts.

2.2.2 Machine Learning

Machine learning is a subfield of artificial intelligence. Its goal is to enable computers to learn on their own. A machine's learning algorithm enables it to identify patterns in observed data, build models that explain the world, and predict things without having explicit pre-programmed rules and models [47].

2.2.3 Artificial Neural Networks

Artificial neural networks are one of the main tools used in machine learning. As the "neural" part of their name suggests, they are brain-inspired systems which are intended to replicate the way that we humans learn. Neural networks consist of input and output layers, as well as (in most cases) a hidden layer consisting of units that transform the input into something that the output layer can use [48].

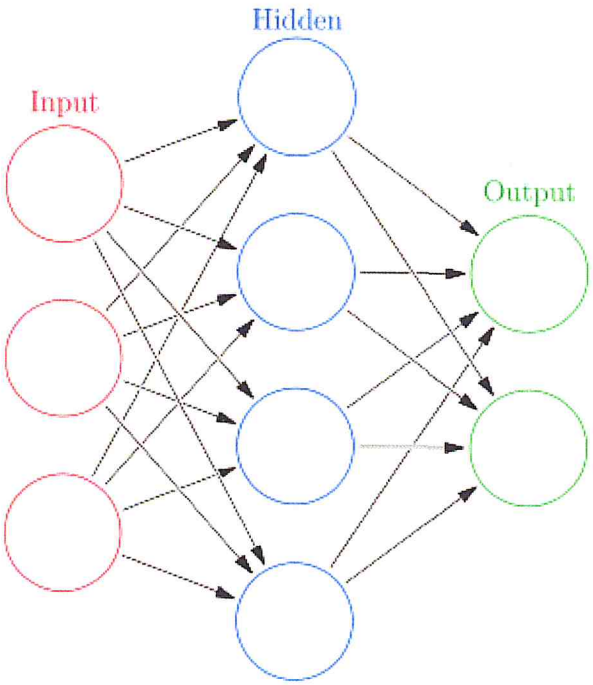


Figure 2-1: Artificial neural network [155].

There are multiple types of neural network, each of which come with its own specific use cases and levels of complexity. The most basic type of neural net is called a feedforward neural network, in which information travels in only one direction from input to output. A more widely used type of network is the recurrent neural network, in which data can flow in multiple directions. These neural networks possess greater learning abilities and are widely employed for more complex tasks such as learning handwriting or language recognition. There are also convolutional neural networks, Boltzmann machine networks, Hopfield networks, and a variety of others. Choosing the right network for your task depends on the data you have to train it with, and the specific application you have in mind. In some cases, it may be desirable to use multiple approaches, such as would be the case with a challenging task like voice recognition [48].

2.2.4 Deep Learning

Adopted from representation learning, deep learning is a set of methods that allows a machine to take raw data fed to it and automatically extract the representations needed for detection or classification, and it does that through multiple levels of representation. Representation is transformed from a level to a higher, more abstract level, through composed, simple, but non linear modules. With the composition of enough such transformations, very complex functions can be learned [49].

We take an image as an example, an image comes in the form of an array of pixel values and the learned features in the first layer of representation typically represent the presence or absence of edges at particular orientations and locations in the image. The second layer typically detects motifs by spotting particular arrangements of edges, regardless of small variations in the edge positions. The third layer may assemble motifs into larger combinations that correspond to parts of familiar objects, and subsequent layers would detect objects as combinations of these parts. The key aspect of deep learning is that these layers of features are not designed by human engineers: they are learned from data using a general-purpose learning procedure [49].

Since its beginning, deep learning has been and still is making record-breaking advances in solving problems that couldn't be solved for many years by the artificial intelligence community. It has turned out to be very good at discovering intricate structures in high-dimensional data and is therefore applicable to many domains of science, business and government. In addition to beating records in image recognition [50][51][52][53] and speech recognition [54][55][56], it has beaten other machine-learning techniques at predicting the activity of potential drug molecules [57], analysing particle accelerator data [58][59], reconstructing brain circuits [60], and predicting the effects of mutations in non-coding DNA on gene expression and disease [61][62]. Perhaps more surprisingly, deep learning has produced extremely promising results for various tasks in natural language understanding [63], particularly topic classification, sentiment analysis, question answering [64] and language translation [65][66].

Researchers and practitioners predict a bright future for deep learning with many more successes and that's due to its ability to take advantage of increases in the amount of available computation and data. New learning algorithms and architectures that are constantly being developed for deep neural networks in order to accelerate this progress.

2.2.5 Classes of Deep Learning

In this section, we will present the three classes of deep learning and share some of the most used neural network types for each with a brief explanation and some application examples.

2.2.5.1 Supervised Learning:

Deep networks for supervised learning, also known as discriminative deep networks, are intended to provide discriminative power for pattern classification purposes, mainly by characterizing the posterior distributions of classes conditioned on the visible data. *Target label data are always*

available in direct or indirect forms for such supervised learning. A lot of discriminative techniques for supervised learning in information and signal processing are shallow architectures, such as Hidden Markov Models (HMMs) [67][68][69][70][71][72][73][74] and Conditional Random Fields (CRFs) [75][76][77][78][79][80].

A CRF is intrinsically a shallow discriminative architecture, characterized by the linear relationship between the input features and the transitions features. Its shallow nature is made by the equivalence established between the CRF itself and the discriminatively trained Gaussian models and HMMs [81]. Deep-structured CRFs have been developed lately by stacking the output in each lower layer of the CRF, together with the original input date, onto its higher layer [82]. It has been applied successfully to phone recognition [83], spoken language identification [82], and natural language processing [82]. With that being said, comparing to a hybrid approach involving DBN, the performance of deep-structured CRFs on the phone recognition task couldn't match the results of the former.

2.2.5.2 Unsupervised Learning:

Deep networks for unsupervised learning, also known as generative learning, are intended to capture high-order correlation of the observed data, when no information about the target class labels is available, for pattern analysis purposes. When used in the generative mode, these networks have the ability to characterize joint statistical distributions of the visible data and their associated classes, when available, and being treated as part of it.

Formally speaking, unsupervised learning is defined as not using any task specific supervision information, such as target class labels, in the learning process. Many deep networks in this category, such as RBMs, DBNs, DBMs, and generalized denoising autoencoders [84], can be used to meaningfully generate samples by sampling from the networks. On the other hand, some networks in this same category, such as sparse coding networks and the original form of deep autoencoders, cannot be easily sampled, and thus are not generative in nature.

Among the different subclasses of generative or unsupervised deep networks, the energy-based deep models, such as the original form of the deep autoencoder [85][86][87], are the most common [85][88][89][90]. Most other forms of deep autoencoders are unsupervised in nature as well, but with quite different properties and implementations.

Deep Boltzmann Machine (DBM) is another prominent type of deep unsupervised models with generative ability [91][92][93][94]. It contains many layers of hidden variables that has no connection between them within the same layer. Each layer captures complicated, higher-order correlations between the activities of hidden features in the layer below. DBMs can potentially learn internal representations that become increasingly complex, something that makes them favorable for solving object and speech recognition problems.

Another representative deep generative network that is used for unsupervised learning is the Sum-Product network (SPN) [95][96]. SPNs are directed acyclic graphs with the observed variables as leaves, and the sum and product operations as internal nodes. The "sum" nodes give mixture models and the "product" ones build up the feature hierarchy.

Recurrent Neural Networks (RNNs) are another class of deep networks that can be considered for unsupervised learning. Using the previous data samples, they are used to predict the data sequence in the future without using any additional class information for learning.

2.2.5.3 Hybrid Learning

Hybrid deep networks, where the goal is discrimination which is assisted, often in a significant way, with the outcomes of generative or unsupervised deep networks. This can be accomplished by better optimization or/and regularization of the deep networks in the second category (unsupervised learning). The goal can also be accomplished when discriminative criteria for supervised learning are used to estimate the parameters in any of the deep generative or unsupervised deep networks in the first category (supervised learning).

The term "hybrid" for this third category refers to the deep architecture that either comprises or makes use of both generative and discriminative model components. In the existing hybrid architectures published in the literature, the generative component is mostly exploited to help with discrimination, which is the final goal of the hybrid architecture.

How and why generative modeling can help with discrimination can be examined from two viewpoints [97]:

• The optimization viewpoint where generative models trained in an unsupervised fashion can provide excellent initialization points in highly nonlinear parameter estimation problems (The commonly used term of "pre-training" in deep learning has been introduced for this reason); and/or

• The regularization perspective where the unsupervised-learning models can effectively provide a prior on the set of functions representable by the model.

The study reported in [97] provided an insightful analysis and experimental evidence supporting both of the viewpoints above. The DBN, a generative, deep network for unsupervised learning, can be converted to and used as the initial model of a DNN for supervised learning with the same network structure, which is further discriminatively trained or fine-tuned using the target labels provided. When the DBN is used in this way we consider this DBN–DNN model as a hybrid deep model, where the model trained using unsupervised data helps to make the discriminative model effective for supervised learning.

Another example of the hybrid deep network is developed in [98], where the DNN weights are also initialized from a generative DBN but are further fine-tuned with a sequence-level discriminative criterion, which is the conditional probability of the label sequence given the input feature sequence, instead of the frame-level criterion of cross entropy commonly used. This can be viewed as a combination of the static DNN with the shallow discriminative architecture of CRF. It can be shown that such a DNN–CRF is equivalent to a hybrid deep architecture of DNN and HMM whose parameters are learned jointly using the full-sequence maximum mutual information (MMI) criterion between the entire label sequence and the input feature sequence.

A further example of hybrid deep networks is the use of generative models of DBNs to pre-train deep convolutional neural networks (deep CNNs) [99][100][101]. Like the fully connected DNN discussed earlier, pre-training also helps to improve the performance of deep CNNs over random initialization. Pre-training DNNs or CNNs using a set of regularized deep autoencoders [102], including denoising autoencoders, contractive autoencoders, and sparse autoencoders, is also a similar example of the category of hybrid deep networks.

2.3 Sentiment Analysis Using Deep Learning Techniques

This section is a review of deep learning applications in sentiment analysis [119]. We tried to cover as many works as possible and they are categorized by deep learning techniques, that way it's easy to distinguish between them.

2.3.1 Convolutional Neural Networks (CNN)

CNNs (convolutional neural networks) [102] are one of the most popular types of neural networks, they include pooling layers and sophistication as it gives a standard architecture to map the

sentences of variable length into sentences of fixed size scattered vectors. This study [103] has proposed a novel convolutional neural network (CNN) framework for visual sentiment analysis to predict sentiments of visual content. Transfer learning approach and hyper-parameter has been used in biases and weights are utilized from pre-trained GoogLeNet. For experimental work, a dataset of twitter containing 1269 images is selected and back propagation is applied. Amazon Mechanical Turk (MTurk) and popular crowd intelligence is used to label the images. Five workers were involved to generate sentiment label in favor of every image. The proposed model was evaluated on this dataset and acquired better performance than existing systems. Results show that the proposed system achieve high performance without fine-tuning on Flickr dataset. However AlexNet was used in previous works and GoogleNet provided almost 9% performance progress than AlexNet. By converting GoogLeNet into a visual sentiment analysis framework, a better feature extraction was achieved. Stable and reliable states were achieved by using hyper parameters.

The authors of [104] have proposed a system of deep learning for sentiment analysis in Twitter. The main focus of this work was to initialize the weight of parameters of a convolutional neural network, and it is critical to train the model accurately while avoiding the requirement of adding new features. A neural language is used to initialize the word embedding and is trained by a big unsupervised group of tweets. For further refining, the embedding on bulky supervised corpus, a conventional neural network is used. To initialize the network, previously embedded words and parameters were used, having the same architecture and training on the supervised corpus as of Semeval-2015. The components used in the proposed work are activations, sentence matrix pooling, softmax and convolutional layers. To train the network, stochastic gradient descent (SGD) and non-convex function optimization algorithms were used and to calculate the gradients back propagation algorithm was used. Dropout techniques were used to enhance the neural networks regularization. The deep learning model is applied on two tasks: message-level task and phrase-level task from Semeval-2015 to predict polarity and achieve high outcomes. By applying a six test-set, the proposed model lies at first rank in terms of accuracy.

A detailed research by [104] has presented an overview of sentiment analysis related to Micro-blogs. The purpose of this effort was to get the opinions and attitudes of users about hot events by using Convolutional Neural Network (CNN). The use of CNN overcomes the problem of explicit feature extraction and learns implicitly through training data. To collect the data from target, the input URL and focused crawler have been used, 1000 micro-blog comments were collected as a corpus and divided into three labels, i.e., 274 neutral emotions, 300 negative emotions and 426 positive emotions.

The proposed model has been compared with the previous studies as those studies had used CRF, SVM and additional traditional algorithms to perform sentiment analysis with a high price. However, the performance proves that the proposed model is reasonable and sufficient to enhance the accuracy in terms of emotion analysis.

Research by [85] was motivated through the need of controlling comprehensive social multimedia content and employing both textual and visual SA techniques for combined textual-visual sentiment analysis. A convolutional neural network (CNN) and a paragraph vector model were used for both the image and textual SA accordingly. The proposed model was termed as a rule-based sentiment classifier VADER. After conducting a wide range of experiments on manually labeled and weakly labeled visual tweets, it was concluded that mutual textual-visual features outperformed the sentiment analysis algorithms which were only depend on visual contents. Getty Images had been selected to crawl data and Caffe was used to tune the CNN model. Tweets were gathered through Twitter API. To make the sentiment labels for the chosen visual tweets, Mechanical Turk (AMT) and crowd intelligence had been employed. The results recommend that the joint textual-visual model has performed better than the both single visual and textual sentiment analysis models.

In study by [105], the researcher has represented a seven-layer framework to analyze the sentiments of sentences. This framework depends on CNN (Convolutional neural network) and Word2vec for SA and to calculate vector representation, respectively. The Dropout technology, Normalization and Parametric Rectified Linear Unit (PReLU), have been used to progress the correctness and generalizability of the proposed model. The framework was verified on a dataset from rottentomatoes.com which contains movie review excerpts' corpus, the dataset consists of five labels, positive, somewhat positive, neutral, negative and somewhat negative. By comparing the proposed model with previous models such as Matrix-Vector recursive neural network (MV-RNN) and recursive neural network (RNN), the proposed model outperformed the previous models with a 45.5 % accuracy.

2.3.2 Recursive Neural Network (ReNN)

Recursive neural networks (ReNNs) [102] usually belong in supervised learning networks. It contains a tree structure which is settled before training and the nodes can have different matrices. There is no need of reconstruction of input in ReNN. The proposed work [106] builds a Treebank for Chinese sentiments of social data to overcome the deficiency of labeled and large corpus in existing models. To predict the labels at sentence level i.e positive or negative, the Recursive Neural Deep Model (RNDM) was proposed and achieved high performance than SVM, Naive Bayes and

Maximum Entropy. 2270 movie reviews were collected from the website and Chinese word segmentation tool ICTCLAS was used to segment these reviews. Five classes were settled for each sentence and the Stanford parser applied for sentence parsing. The proposed model improved the prediction of sentiment labels of sentences by concluding 13550 Chinese sentences and 14964 words. ME and NB perform higher with contrastive conjunction structure than baselines with great margin.

In this study [107], a model comprising RNTN (Recursive Neural Tensor Network) and Sentiment Treebank has been proposed to correctly clarify the compositional effects of phrases at different levels, i.e., positive and negative phrases.The proposed model was compared with all the existing models. In existing models, the meaning of long phrases cannot be expressed effectively by semantic word spaces, so for sentiment detection, more rich and supervised evaluation and training resources are needed as it requires more influential composition models. The RNTN achieved 80.7% accuracy in sentiment prediction by performing fine-grained labeling over all the phrases and outperformed previous models.

This study [108] has contributed a generalized and scaled framework to recognize top carding/malware sellers. The model is based on deep learning for sentiment analysis and used in thread classification and snowball sampling to assess the quality of the sellers' service/product by analyzing the customer feedback. The evaluation of the proposed model has been conducted on a Russian carding forum and a web crawler was used to gather the conversation from the forum.

A sentiment treebank has been used and it was trained by using recursive neural tensor network on an online review corpus. For evaluating the validity and effectiveness, two experiments were conducted in which the proposed model was compared with Naive Bayes, KNN and SVM based models. This study has searched out the sellers who are highly rated for malicious services/products and the effectiveness of deep learning for recognizing these sellers. Results have indicated that deep learning techniques accomplish superior outcomes than the shallow classifiers and it was established that the carding sellers have fewer ratings than malware sellers.

2.3.3 Deep Neural Networks (DNN)

In this study [109], author has proposed a model for sentiment analysis considering both visual and textual contents of social networks. This new scheme used deep neural network model such as denoising auto encoders and skip-gram. The base of the scheme was CBOW (Continuous Bag-Of-Words) model. The proposed model consisted of two parts CBOW-LR (logistic regression) for textual contents and was expanded as the CBOW-DA-LR. The classification was done according to the

polarity of visual and textual information. Four datasets were evaluated, i.e., Sanders Corpus, Sentiment140, SemEval- 2013 and SentiBank Twitter dataset. The proposed model outperformed the CBOWS+SVM and FSLM (fully supervised probabilistic language model). Perhaps the ESLAM (extended fully supervised probabilistic language model) in term of small training data had outperformed the current model. The feature learning and skip grams both required large datasets for best performance.

In this study [110], deep neural network architecture has been proposed to evaluate the similarity of documents. The architecture was trained by using several market news to produce vectors for articles. The T&C news have been used as dataset. The cosine similarity was calculated among labeled articles and the polarity of documents was considered but content was not considered. The proposed method accomplished superior performance in terms of similarity estimation of articles according to polarity.

2.3.4 Recurrent Neural Networks (RNN)

The Recurrent neural network (RNN) [102] is an influential model in language modeling because it does not represent the context of fixed-length that contaminate all history words. In this study [111], the HBRNN (hierarchical bidirectional recurrent neural network) has been developed to extract the reviews of customers about different hotels in a complete and concise manner. To model the sequential long term information, HBRNN has used the terminology of RNN and the prediction process was done at review level by HBRNN. The experimental data was taken from DBS text mining Challenge 2015. HBRNN performance was improved through networks parameters along with the fine tuning and the model was compared with LSTM (long short-term memory) and BLSTM (Bidirectional LSTM). After performing the experiments, the evaluation recall, F1 scores and precision was made on highly biased data. The development, test set and train splits were used for comparing outcomes with benchmark systems, tenfold cross validation used to present the performance of HBRNN.

The main challenges that was resolved is lack of online reviews with high quality and lack of high skewness in the reviewed data. Experimental Results on the dataset proved that HBRNN performed better than other methods. This model can be applied to other opinion mining activities which consists of huge data volume. This contribution [112] has been done to overcome the issue of dataset of Bangla as it is standard and large for SA (Sentiment Analysis) tasks. The issue has been resolved by providing a significant dataset for sentiment analysis of 10,000 BRBT (Bangla and Romanized Bangla Text). The Deep Recurrent model especially LSTM (Long Short Term Memory)

was used to test the dataset by using two loss functions, i.e., binary and categorical cross-entropy. Data were gathered from different sites like YouTube, Facebook, Twitter and others. The experiments were conducted to prepare dataset of one mark for another (and the other way around) to investigate the fact whether it contributes towards the better outcomes.

This author [113] proposed a sequence model to focus on the embedding of reviews having temporal nature toward products as these reviews had less focus in existing studies. The combination of gated recurrent units with recurrent neural network is used to learn dispersed representations of products and users. For sentiment classification these representations fed into machine learning classifier. The approach was evaluated on three datasets collected from Yelp and IMDB. Each review labeled according to rating score. To train the network the back-propagation algorithm with Adam stochastic optimization method has been used. Results show sequence modeling of dispersed product and user representation learning improves the performance sentiment classification of document-level and the proposed approach achieves high-tech results on the benchmark datasets. The result of proposed model compared with many baselines including recursive neural networks, user product neural network, word2vec, paragraph vector and algorithm JMARS. We have also made an analysis of some approaches based on Recurrent Neural Network and this analysis is presented in tabular form in Table IV.

2.3.5 Deep Belief Networks (DBN)

Deep belief networks (DBNs) [114] includes several hidden layers, composed by RBM (restricted Boltzmann machines). DBN has been proved efficient for feature representation. It utilizes the unlabeled data and fulfills the deficiencies of labeled analysis issues. In this paper [115], a new deep neural network structure has been presented termed as WSDNNs (Weakly Shared Deep Neural Networks). The purpose of WSDNNs is to facilitate two languages to share sentiment labels. The features of language specific and interlanguage have been presented through building multiple weakly shared layers of features. The datasets from Prettenhofer and Stein have been used containing four languages French, German, English and Japanese. In comparison with existing studies the proposed work address the challenge of shortening overlap among feature spaces of both source and target language data through cross lingual information transfer process using back propagation. DNNs used for transformation of information from source to target language. The experiments have been conducted for sentiment classification tasks of cross multilingual product reviews of Amazon. results concluded that the proposed approach is more effective and powerful in terms of cross lingual sentiment classification than the previous studies.

Another study by [116] has used deep belief network with word vector for the political detection in Korean articles. The proposed model has used SVM for bias calculation, five stage pipeline for detection of political bias, python web crawler to gather news articles, KKMA for morpheme analysis, word2Vec and scikit-learn package. The dataset contained 50,000 political articles from 01 Jan, 2014 to 28 Feb, 2015. Results showed 81.8% accuracy by correctly predicting labels and the results contained mean square error of 0.120. This research [114] has proposed a deep belief network with feature selection (DBNFS) to overcome the vocabulary problems, the network has used input corpus along with numerous hidden layers. Chi-Squared technique of feature selection was used to improve the Deep Belief Network (DBN) for the purpose of decreasing complexity of vocabulary input and for eliminating irrelevant features. By applying Chi-Squared technique, the learning phase of DBN was enhanced to DBNFS. In this work, two new tasks features, selection and reduction were used along with many other tasks of existing classification approaches, such as data partitioning, feature extraction, model training and model testing. Performance of DBNFS was demonstrated and training time and accuracy of proposed DBNFS was also compared with other algorithms. Five Dataset of sentiment classification were used for estimation, datasets are books (BOO), electronics (ELE), DVDs (DVD), kitchen appliances (KIT) and movies reviews (MOV). For fair comparison, the parameters of learning were same as of existing works. Accuracy was evaluated by comparing the amount of features before and after the feature selection and reduction. The accuracy results were compared with the previous works and were proved better DBNFS than DBN. The training time was also lower in DBNFS than DBN. Training time was improved due to simple deep structure and proposed feature selection method. The only drawback of DBN was that it is costly plus time consuming.

2.3.6 Hybrid Neural Networks

This study [117] has proposed two deep learning techniques for the sentiment classification of Thai Twitter data, i.e., Convolutional Neural Network (DCNN) and Short Term Memory (LSTM). Data processing was conducted properly. Data was collected from the users and their followers of Thai Twitter. After filtering the data, only the users with Thai tweets and tweets with Thai characters were selected. Five experiments were conducted to achieve finest parameters for deep learning, to compare the deep learning with classical techniques and to achieve the words sequence importance.

Three-fold cross validation was used to verify the process. The results concluded that the accuracy is high in DNN than LSTM and both techniques of deep learning are higher in accuracy than SVM and Naive Bayes but lesser than Maximum Entropy. Higher accuracies were found in

original sentences than shuffled sentences so the words sequence is important. In this research study [118] a hybrid model has proposed which consists of Probabilistic Neural Network (PNN) and a two layered Restricted Boltzmann (RBM). The purpose of proposing this hybrid deep learning model is to attain better accuracy of sentiment classification. The polarity, i.e., negative and positive reviews vary according to different context in order to solve this type of problem this model performs well, neutral reviews are not considered. Experiments have done with datasets of Pang and Lee and Blitzer, et al, binary classification implemented on every dataset. The accuracy has been enhanced for five datasets by comparing with the existing state-of-the-art [120]. There are no outer resources in proposed approach such as POS tagger and sentiment dictionary etc therefore it is faster too than competitor. To attain a reduced number of features the dimensionality reduction has been implemented as previous study used a complex strategy for feature selection.

2.3.7 Other Neural Networks

In this study [121] to overcome the complexity in word level models the character-level model have been proposed. The motivation of proposed model CDBLSTM is an existing model that is DBLSTM neural networks [122]. The focus of this work is only on textual content and on the polarity analysis of tweets in which a tweet is classified into two classes, i.e., positive and negative. There can be more options than positive and negative such as natural and finer but here the model is restricted only to positive and negative classes to compare with existing published results. The tweets are encoded from character level and trained by the use of CCE (categorical cross-entropy). Experiments were conducted on two datasets, first one is latest benchmark dataset for SemEval 2016 and the second one is provided by GO dataset. Adam algorithm was implemented to train all the models and the learning rate settled to 0.1. Final predictions were obtained by the model of logistic regression. Results demonstrate that proposed approach is competent for the polarity problems. By applying different experiments results shows that CDBLSTM performs better than DBLSTM and by comparing the results with Deep Convolutional Neural Network (DCNN) [123] which performs well on twitter SPC (sentiment polarity classification). The 85.86% accuracy was achieved on STS (Stanford Twitter Sentiment) corpus and 84.82% on SemEval-2016. In this study [122], author has proposed TF-IDF, GR, and RBFNN for sentiment classification on Hinglish text. Many studies have worked on sentiment analysis of various languages such as English, Turkish, Flemmish, Spanish, Arabic and Chinese but no work has been done on Indic language. To fill this gap, the sentiments were classified in Hinglish language as it contains Hindi words along with the English. Dataset has creep from Facebook comments and viz. news, five methods of feature selection information gain,

chi-square, t-statistics, association and gain ratio have been implemented on DTM (Document-Term Matrix) and TF-IDF (Term Frequency-Inverse Document Frequency). Many classifiers have used such as SVMs (Support Vector Machine), RBFNet (Random Forest, Radial Basis Function Neural Network), Naive Bayes, J48 (Decision Tree), CART, JRip, Logistic Regression (LR) and Multi-layer Perception (MLP) methods for the classification of data. Total 840 experiments were performed on datasets and best results were achieved. The proposed triumvirate approach was proved efficient for sentiment classification of Hinglish text.

This contribution [123] overcomes the problem that occurs in effectively analyzing the emotions of customers toward companies in blog sphere. A neural network (NN) based technique is proposed which subordinate the advantages of Semantic orientation index and machine learning methods for the classification of sentiments effectively and quickly. The input of neural network is semantic orientation indexes. While considering the fault tolerance ability the BPN (backpropagation neural network) is selected as basic learner of proposed method. Data was collected from the blogs of real world such as from "LiveJournal" and "Review Center". Segmentation of words, SO indexes calculations, neural network trainings and performance evaluations have been conducted. Training and test sets were settled of datasets. The overall accuracy (OA), performance evaluation matrices and F1 were used. Results concluded that proposed method has enhanced the performance of classification and saved training time as compared to traditional ML and IR.

This contribution [124] proposed a data driven supervised approach for the purpose of feature reduction and development of lexicon specific to twitter sentiment analysis about brand. Statistical analysis and n-gram were used for twitter specific lexicon and feature reduction accordingly. The existing models SVM and Naive Bayes were used for twitter specific lexicon to compete with the existing studies. The classification was done in proposed model by using artificial neural networks for twitter specific lexicon and this difference outperformed the existing models. The input matrix was parse matrix. Datasets were divided into training and test datasets to achieve better accuracy. The feature engineering phase was done through preprocessing activities for transforming the documents in simple form and to produce the vector presentation. The data was collected from Twitter API v1.0. The proposed model was applied on Justin Bieber Twitter corpus and it was established that the emotions contain high explanatory power as compared to existing studies. The proposed model has reduced the features of Justin Bieber corpus and enhanced the classification accuracy and high amount of coverage. Only six expressions were found related to Justin Bieber brand out of 181 and

others were found twitter-specific. The proposed model has facilitated the Justin Bieber brand to identify the issues and views about the brand.

2.4 Conclusion

As it was shown in this chapter, even though it's still considered as a new technology, deep learning has been and is being used in many areas, including sentiment analysis, and its results are phenomenal.

With that being said, with all these studies, we noticed that no one has ever addressed our problematic specifically, therefore, we will explore in the next chapter some of the works done related to that plus some interesting deep learning studies that helped us in our research.

Chapter 3 Related Works

3.1 Introduction

As in any other research project, studying previous works done in the same field and to solve the same problematic plays a very important role in using the right approach and finding the most appropriate solution, therefore, in this chapter we will share the highlights of our literature review.

We start the chapter by sharing some statistics about mental health and depression in order to highlight our problematic and emphasize on why is it important to address such matters. We dive deeper into our problematic by sharing the main obstacles found in depression detection and the traditional techniques used commonly. In the second section of the chapter, we present the different works done previously to solve the problems addressed in the first section and share the main contributions and results of each. After that, in the discussion section, we share the most important remarks that were noticed during our literature review and what decisions did we make based on those observations.

3.2 Major Depressive Disorder

Mental health is considered as one of today's world's most prominent plagues. Estimations show that one fourth of American citizens suffer from a diagnosable mental disorder [125], this, combined with the 2015 US Census for Residents 18 and older tell us that approximately 80 million United States citizens suffer from a mental disorder. One in three of these citizens may be suffering from clinical depression, formally known as Major Depressive Disorder (MDD), that's why this matter is becoming more and more popular research topic lately [126].

Nearly 300 million people suffer from clinical depression globally. What's even scarier is that 3 – 5% of males and 8 – 10% of female from the total world's population are likely to experience a major depressive episode within a period of one year [127]. MDD often co-occurs with other illnesses and mental conditions. According to [128], it is experienced by one fourth of cancer patients, one third of heart attack survivors, and up to 75% of individuals diagnosed with an eating disorder. Even more absurd conditions and factors have been associated with depression, and combined together, they usually weigh heavily on patients making the quality of their life and their peers' more burdensome [129].

Research has shown that individuals suffering from one or more of mental illnesses will likely experience a snowball effect towards other, something that increase the probability of suicide exponentially [130]. According to [131], more people are committing suicide than committing a homicide, more specifically, for every two of the latter we have three of the former. With that being said, [132] explains that depression was the cause of over two-thirds of 30,000 suicides reported in the past year. Furthermore, other studies have shown that untreated depression is the number one cause of youth suicides, knowing that the latter itself is the third leading cause of death among children [133]. To conclude, these statistics showcase some of the realities and potentials damages that depression cause, leading us to agree about the urgency of paying attention and solving this high priority problem that threatens our society.

Nonetheless, current identification, support, and treatment methods of clinical depression are considered as inefficient, and while 87% of the governments in the world provide some form of primary care to combat mental illnesses, 30% of them provide no institution at all for mental outreach [134]. What makes mental health complicated is that the diagnosis is extracted from the patient's self-reported experiences, behavior questionnaires, and surveys, no laboratory test has been created specifically for the matter. For instance, if we take depression, it comes in different degrees and the examinations are usually done through one of the popular questionnaires used by psychologists, such as the *Center of Epidemiologic Studies Depression Scale (CES-D)* [135], *Beck's Depression Scale (BDI)* [136], *and Zung's Self-Rating Depression Scale (SDS)* [137], but as we mentioned above these examinations lack empirical data as they use the patient's observations or a third-party's ones, which puts the results under the risk of flawed subjective human testing that can be manipulated easily, often with the purpose of gaining antidepressants or just to hide one's own depression from peers [129].

Before we tackle the flaws of the current treatment methods, we need to address the biggest challenge of depressed individuals, which is seeking out treatment. As the World Health Organization reports [138], the vast majority, and most particularly youth, of depressed people never seek out for help due the probable consequences that can be faced, such as blame and stifled self-esteem. Thus, depression often goes unrecognized, even during primary health care examinations [139].

On the bright side, once identified and treated, clinical depression has proven to have far-reaching impacts upon society in a short period, as [133] study shows, up to 80% of treated patients showed improvements within four to six weeks. Another study by the National Institute of Mental Health [140] showed that remission rates reach over 65% after six months of treatment. With that

being said, our challenge seems to lie in the identification phase rather than the treatment one, letting us think of new improved methods and techniques that can help us identify MDD.

Nowadays, people are using social media to express their feelings and share their innermost thoughts and desires, most importantly, all of that is done in a naturalistic way, giving us an opportunity to overcome the manipulation issue addressed in self-reported depression questionnaires. Thus, it allows us to capture these thoughts in their rawest form and use them to identify the publisher's present state of mind, which can be used, using sentiment analysis techniques, to detect clinical depression. The question that remains is how can we do that? For that we are going to go through the different works that have been done in this fiend and move on later in the next chapter to our proposed solutions.

Major Depressive Disorder identification has been the subject of research of many fields, psychiatry, psychology, medicine and even sociolinguistics fields. We mentioned previously the questionnaires used mainly in medicine and psychology, CES-D [135], BDI [136], and SDS[137], but other studies, such as (Redei et al.,) showed that there are biological markers as well, which could increase specify in the diagnosis for clinical depression. Their analysis of 26 candidate's blood transcriptomic markers in a sample of 15 – 19 year-old subjects resulted in a correct diagnosis for 11 out of 14 candidates who suffered from depression; another panel was able to distinguish between MDD or comorbid anxiety for 18 individuals [141]. Redei et al. is notable for being the first significant approach towards identifying depression from a medical perspective, although numerous exist from a data science perspective [141].

Approaches that utilize objective information, such as log data about an individual's activities to predict depression have been studied recently. Resnik et al. has formulated a method for identifying depression in individuals through analyzing textual data written by these individuals. They obtained topics from the essays written by college students by applying latent Dirichlet allocation (LDA), a popular topic-extraction model within Machine Learning [142]. Through using these discovered topics from a statistical model, they were able to estimate depression and neuroticism in college students with an r value of .45, thus discovering a slight correlation between neuroticism, depression, and academic works by college attendees. Resnik et al. becomes relevant for their novel use of topic modeling; otherwise these academic works are often a poor dataset to derive diagnoses from [142].

3.3 Detecting Depression in Social Media

In [129], a dataset created by Coppersmith et al. for the Computational Linguistics and Clinical Psychology (CLPsych) 2015 Shared Task was used to study the potential of using Twitter as a tool for measuring and predicting Major Depressive Disorder. Several statistical classifiers were developed and compared: Decision Trees, Linear Support Vector Classifier, Naive Bayes 1-gram, Naive Bayes 2-gram, Logistic Regression, and Ridge Classifier. When it comes to accuracy, the unigram-based Naive Bayes approach excelled with 86% accuracy, but it fell short when it comes to precision, recall, and F-score. On the other hand, the Logistic Regression model scored the highest in precision with a 86% precision, outperforming all the other classifiers. When it comes to recall, the Linear SVM attained the highest score (83%) out of all, yet it fell behind in precision and accuracy. With that being said, the study shows that if prioritization has to be made, recall is more important than precision, because identifying a few false positives is better than strictly identifying the most depressed individuals and missing potentially affected ones. Accuracy was prioritized as well over F1-score because a model which identifies depression well is more important than one which becomes unreliable through a myriad of false positives. Finally, taking all of this in consideration, and adding the fact that computational resources and time are to be considered, a multinomial approach towards a Naïve Bayes classifier was chosen to be explored for further research due to the results it showed in the experiments.

Another study [143] reviews recent studies that aimed to predict mental illness, including but not limited to depression only, using social media. Mentally ill users have been identified using screening surveys, their public sharing of a diagnosis on Twitter, or by their membership in an online forum, and they were distinguishable from control users by patterns in their language and online activity. The study emphasizes on the potential of using social media for mental illness detection and how it can fill the gaps existing in other sources of data used in the experiments, however, it highlights the major obstacle in such platforms, which is the difficulty to detect the illness in people who are unaware of their mental health status. It addresses as well one of the most important points that pops up in such studies, which is how ethical and legal is it to use public data for such purposes.

In another study [144] a user-level and tweet-level classifiers were developed and compared to discover which is best to detect at-risk individuals. The data used was collected from the #BellLetsTalk Twitter campaign, which is a wide-reaching initiative that aims to break the silence around mental illness and support mental health across Canada. Both classifiers were tested through

different experiments and the results showed that the user-level models perform much better even with a small number of features.

[145] aims to make timely depression detection via harvesting social media data. In this study, benchmark datasets were constructed specifically for online depression detection. A well-labeled depression dataset, a non-depression one, and a large-scale depression-candidate one as well. After that, six depression-related feature groups were extracted, covering not only the clinical depression criteria, but online behaviors as well. A multimodal depressive dictionary learning (MDL) model was proposed and validated through a series of experiments, which showcased an outstanding performance comparing to other related works. Finally, an analysis have been conducted on feature contributions and online behaviors of depression in order to reveal the ones not covered in the depression criteria and that are important for such researches.

In [146], a dataset set was built via automatically derived samples from a large amount of Twitter data. The study examines four mental health disorders in particular: depression, bipolar disorder, post traumatic stress disorder, and seasonal affective disorder. A statistical classifier was used while building the dataset to differentiate between the users with these disorders. An LIWC analysis of each disorder was conducted after that in order to measure the deviations in each illness from a control group. Different experiments were done during the process and taken together, the results indicate that there are diverse sets of quantifiable signals relevant to mental health observable in Twitter.

Another study [147] uses machine learning to automatically categorize anxiety patients' internal sentiment and emotions using classifiers based on n-grams syntactic patterns, sentiment lexicon features, and distributed word embeddings. The dataset used was annotated by psychology experts, specifically targeted towards sentiment analysis for mental health. In this work, the traditional "neutral" polarity class was divided into both a dual polarity sentiment (both positive and negative) and a neither positive nor negative sentiment. A four-class and binary-class polarity prediction experiments and the results showed the the latter brought better results, 90% accuracy, while the former fell short with only 78%.

This study [148] uses deep learning to investigate the relationship between computational models and psychological states. A CNN, an LSTM, and a GRU were used for experiments and the results showed that even though CNN outperformed the other two models in terms of accuracy, its output is unreasonable comparing to human's sentiments, thus, it was concluded that the accuracy of

the model can't reflect the psychological state of a person. GRU on the other hand, showed more reasonable results.

[149] is the only work that we found that uses deep learning to solve the same problematics as ours. The dataset used was generated by scraping tweets off various Twitter pages and labelling them with the aid of a polarity score generated by Textblob's python package. Different deep learning models were experimented with in this study (CNN, RNN, and GRU). Other criteria was taken in consideration to compare the performance, they examined character-based against word-based models and pretrained embeddings against learned embeddings. The results showed that word-based GRU outperformed all the other models with 98% accuracy. A word-based CNN was considered as one of the most effective models, as it resulted in a 97% accuracy.

Last but not least, many other studies were consulted, such as [150][151][152][153][154]. Most of them covered the same topics and discoverings shared in other studies above. We will discuss in the next section how these works contributed in our research and influenced our proposed model.

3.4 Discussion

The first thing we noticed during our literature review is that many works addressed the same problematic as ours, but almost all of them used traditional statistical models as classifiers. The different studies shared in chapter two showed the popularity of deep learning usage in sentiment analysis but as we started narrowing our focus to only mental health, only two works were found the use deep learning for mental health purposes, and both of them are very recent, one of them was published during the course of research.Thus, we conclude that deep learning techniques weren't taken advantage of to solve mental health problems.

Next, through the works shared, we noticed that the richest source of data in terms of quantity, diversity, and rich content (sentiment-wise) is Twitter. It has be proven that such data can be used to study clinical matters, especially when it comes to mental illnesses.

Moreover, all of the studies showed how important and influential the data used is on the final results. Using the same techniques and methods on different datasets always outcomes different results, making the dataset one of, if not the most, important factor in a success or failure of a study.

Although all the studies shared above contributed to our study and conclusions, here is a table to summarize all the works that woked specifically for a similar purpose as ours:

Study	Year	Goal	Model	Dataset	Performance
[129]	2016	Identifying Depression on Twitter	Decision Trees	CLPsych 2015 Shared Task	Precision: 67% Recall: 68% F1-score: 75% Accuracy: 67%
			Linear Support Vector Classifier		Precision: 83% Recall: 83% F1-score: 83% Accuracy: 82%
			Naive Bayes 1-gram		Precision: 81% Recall: 82% F1-score: 81% Accuracy: 86%
			Naive Bayes 2-gram		Precision: 82% Recall: 82% F1-score: 82% Accuracy: 82%
			Logistic Regression		Precision: 86% Recall: 82% F1-score: 84% Accuracy: 82%
			Ridge Classifier		Precision: 81% Recall: 79% F1-score: 78% Accuracy: 79%
[144]	2017	Detecting at-risks users	Tweet-level classifier exp1-Undersample	#BellLetsTalk campaign	Precision: 12% Recall: 80% F1-score: 21% Accuracy: 61%
			User-level classifier exp5-Original	#BellLetsTalk campaign	Precision: 70% Recall: 85% F1-score: 77% Accuracy: 78%
			User-level classifier exp5-Original	CLPsych 2015 Shared Task	Precision: 58% Recall: 77% F1-score: 66% Accuracy: 60%

[145]	2017	Depression detection	MDL	Depression Dataset	Precision: 85% Recall: 85% F1-score: 85% Accuracy: 85%
[149]	2018	Depression detection	RNN	Twitter Depression Dataset	Precision: 92% Recall: 89% F1-score: 90% Accuracy: 90%
			GRU		Precision: 99% Recall: 97% F1-score: 98% Accuracy: 98%
			GRU with CHAR	Twitter Depression Dataset	Precision: 96% Recall: 93% F1-score: 94% Accuracy: 94%
			CNN		Precision: 99% Recall: 98% F1-score: 99% Accuracy: 98%

Table 3-1 : Comparison table of depression detection works.

With that being said, as a result to our literature review, the final decisions were made about what our study should use and how are we going to continue the rest of the process:

- We will use Twitter as a data source. More specifically the benchmark datasets developed in [145]. More details will be provided about it in chapter five.
- We will focus on binary-class polarity but instead of the common positive-negative model, ours will be depressed, not depressed.
- We will use a tweet-level model instead of a user-level one because according to our research that's the area that need more enhancements the most.
- We will use a deep learning model. More details about the techniques chosen will be shared in the next chapter.
- Finally, we use accuracy as the most important metric for our results because of the reasons given in [129].

These were the most important points that we decided after going through our literature review. More details will be shared in the next chapters about the choice of models and everything that was used for this project.

3.5 Conclusion

In this chapter, we presented the most popular works that were conducted the same problematic as ours. We covered the most relevant ones to our study and most influential ones in terms of building our proposed model and choosing the right data and criteria.

The next part of this thesis will be dedicated to share all the details about our proposed models and the process we went through to design it, build it and test it.

Part II: Proposed Model

Chapter 4 Model Design

4.1 Introduction

After going through the literature review and seeing some of the works that have been done in this field for similar purposes, we will introduce in this chapter the architecture of our proposed model. We start first by presenting the model in general, then we dive deeper into each one of its components by explaining its origins and how has it been adapted for our goal and problematic.

4.2 Architecture

After choosing the right dataset for our study, we proposed the following architecture for our deep learning model. It consists of five layers: Preprocessing layer, Word Embeddings layer, CNN layer, Bi-LSTM with Attention layer, and finally a Softmax layer.

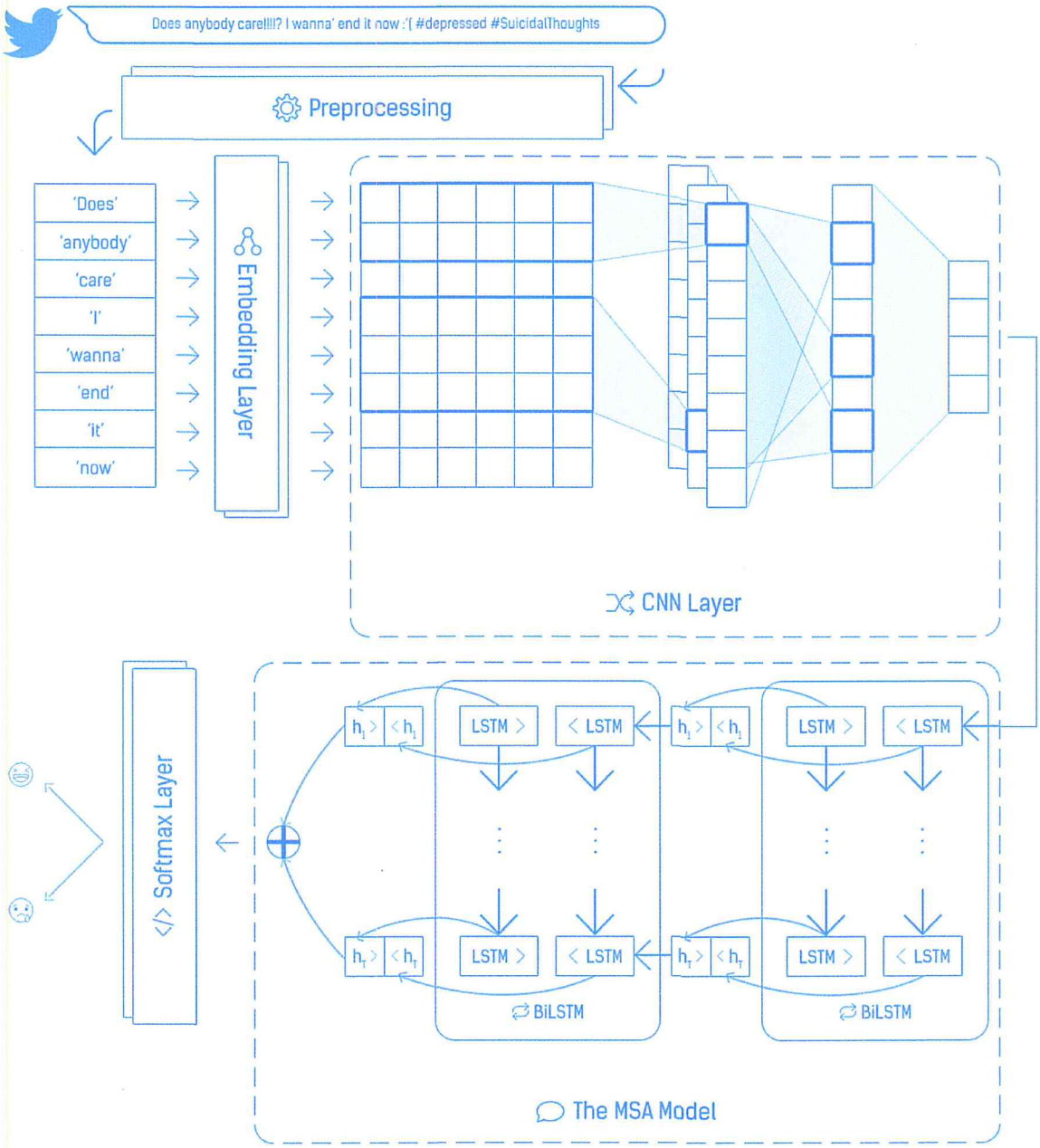


Figure 4-1 : Our depression detection model architecture.

Different works have used CNN or LSTM separately for similar research studies but according to our research, none have used these two deep learning methods simultaneously for the same purpose as ours.

4.2.1 Preprocessing

As in any other social media platform, users tend to express themselves in everyday's slang language, which makes it rare to find well-formed sentences that respect grammatical and linguistic rules, furthermore, abbreviations, emojis and smileys are widely used, especially when expressing feelings, opinions, or any form of self-expression, in other words, sentiment, which is the subject of our study. With that being said, these factors have been known to pose a major challenge in NLP and as much as this field is reaching its most advanced levels, they still are considered as the most important bottlenecks when dealing with raw text. In addition to that, dealing with tweets gives us other factors to take in consideration, such as URLs, hashtags, mentions, and reserved words (RT, FAV).

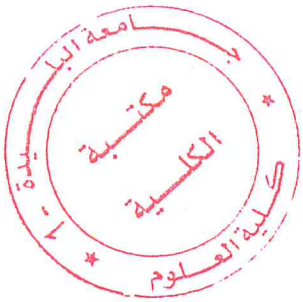
Even though some of this data can be useful to the sentiment expressed in the tweet, keeping them requires a very complexe model that is able to handle every possibility, changing our focus from depression detection to solving natural language processing issues. Thus, the use of a text preprocessor becomes a necessity rather than an optional step.

Our preprocessing phase is divided in two major steps, cleaning and tokenization.

4.2.1.1 Cleaning

As a first step, we treat the tweets as one big text corpus, we parse through it and delete all of the following elements:

- Punctuation
- URLs
- Hashtags
- Mentions
- Reserved words (RT, FAV)
- Emojis
- Smileys



During this phase, we treat all tweets equally, we don't take in consideration the language of the tweet or any syntactic features of the sentences and words. The tweets are stored then in one text file which will be used as the corpus fed to our neural network.

4.2.1.2 Tokenization

Given a character sequence and a defined document unit, tokenization is the task of chopping it up into pieces, called tokens, perhaps at the same time throwing away certain characters, such as punctuation. These tokens are often loosely referred to as terms or words, but it is sometimes important to make a type/token distinction. A token is an instance of a sequence of characters in some particular document that are grouped together as a useful semantic unit for processing. A type is the class of all tokens containing the same character sequence.

4.2.2 Embedding Layer

In traditional NLP systems and techniques, words are treated as atomic units, no notion of similarity is present because these words are represented as indices in a vocabulary. [156] explains that using distributed representations of words in a vector space help learning algorithms to achieve better performance in natural language processing tasks, such as sentiment analysis, by grouping similar words.

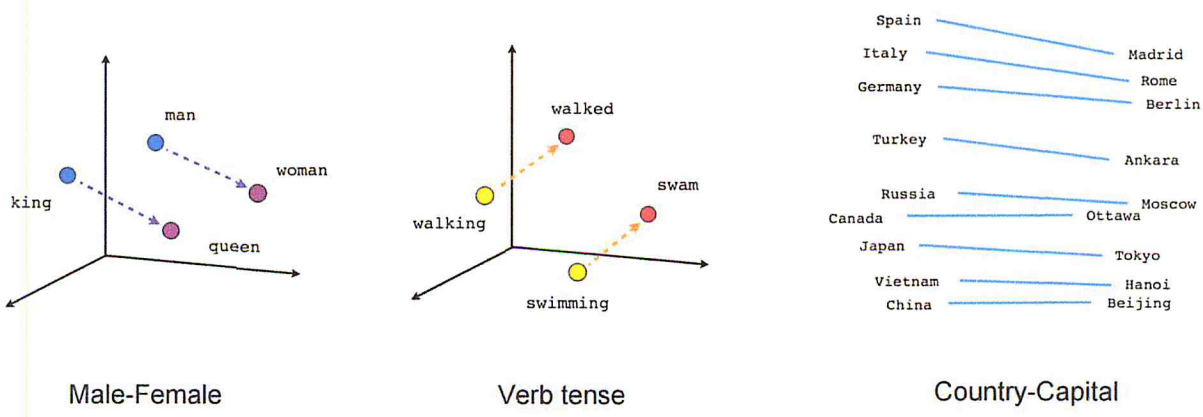


Figure 4-2: Vector representations of words using Word2Vec model.

Somewhat surprisingly, it was found that similarity of word representations goes beyond simple syntactic regularities [157]. Using a word offset technique where simple algebraic operations are performed on the word vectors, it was shown for example that $vector("King") - vector("Man") + vector("Woman")$ results in a vector that is closest to the vector representation of the word $Queen$ [158].

The word representations computed using neural networks are very interesting because the learned vectors explicitly encode many linguistic regularities and patterns. Somewhat surprisingly, many of these patterns can be represented as linear translations. For example, the result of a vector calculation $vec(\text{“}Madrid\text{”}) - vec(\text{“}Spain\text{”}) + vec(\text{“}France\text{”})$ is closer to $vec(\text{“}Paris\text{”})$ than to any other word vector [157][158].

As any area of research, there are many techniques on how to use this technique but before we move to that let's define first word vectors to understand better what comes next.

4.2.2.1 Skip-gram model

The Skip-gram model, introduced by [156], is an efficient method for learning high quality vector representations of words from large amounts of unstructured text data. Unlike former neural network architectures used for learning word vectors, training the Skip-gram model doesn't involve dense matrix multiplications, which makes it outrageously efficient by giving the possibility of training the model on more than 100 billions words in one day, on an optimized single-machine implementation [156].

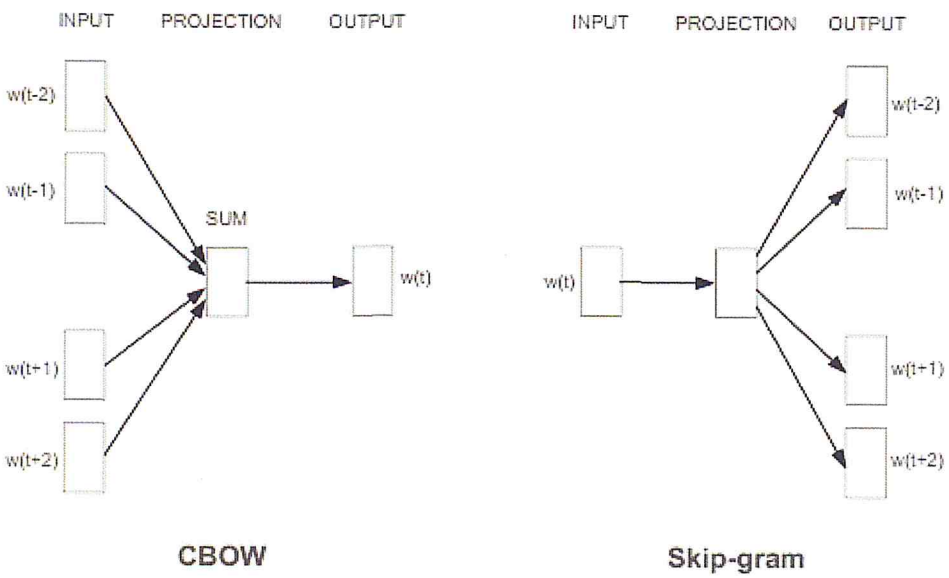


Figure 4-3: The Skip-gram model architecture [156].

The Skip-gram model's main objective lies in finding word representations that are useful for predicting the surrounding words in a sentence or a document, or a tweet in our case, which is its

main differentiator from other word embedding architectures. To make it clearer, for a given sequence of training words $w_1, w_2, w_3, \ldots, w_T$, the model's objective is to maximize the average log probability

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0} log\, p(w_{t+j}\,|w_t)$$

Where c is the size of the training context, and the more we increase it the more training examples we will have and thus a possibility of having a higher accuracy, at the expense of the training time.

4.2.3 CNN Layer

Famously know in the Computer Vision field, Convolutional Neural Networks (CNNs) provoked major breakthroughs in Image Classification and are the core of most Computer Vision systems today. CNNs are a special type of feedforward neural networks, originally inspired from the human visual cortex, they consist of multiple convolutional layers, each of which performs the function latter's cells [159].

In recent years, CNNs started being used even in Natural Language Processing tasks and the results were surprisingly impressive. By using word vectors to build the input matrix of the model, text was treated in the same way as images, both for feature-extraction and classification, and ever since it became one the most used neural networks in NLP [160]. Before we dive deeper in the details of CNNs' architecture, we start by defining Convolution to understand how features get extracted from images and text.

4.2.3.1 Convolution

As Figure 4-4 shows, convolution can be thought of as a sliding window function that we apply to a matrix in order to extract features. This window is known as a kernel, filter, or feature detector.

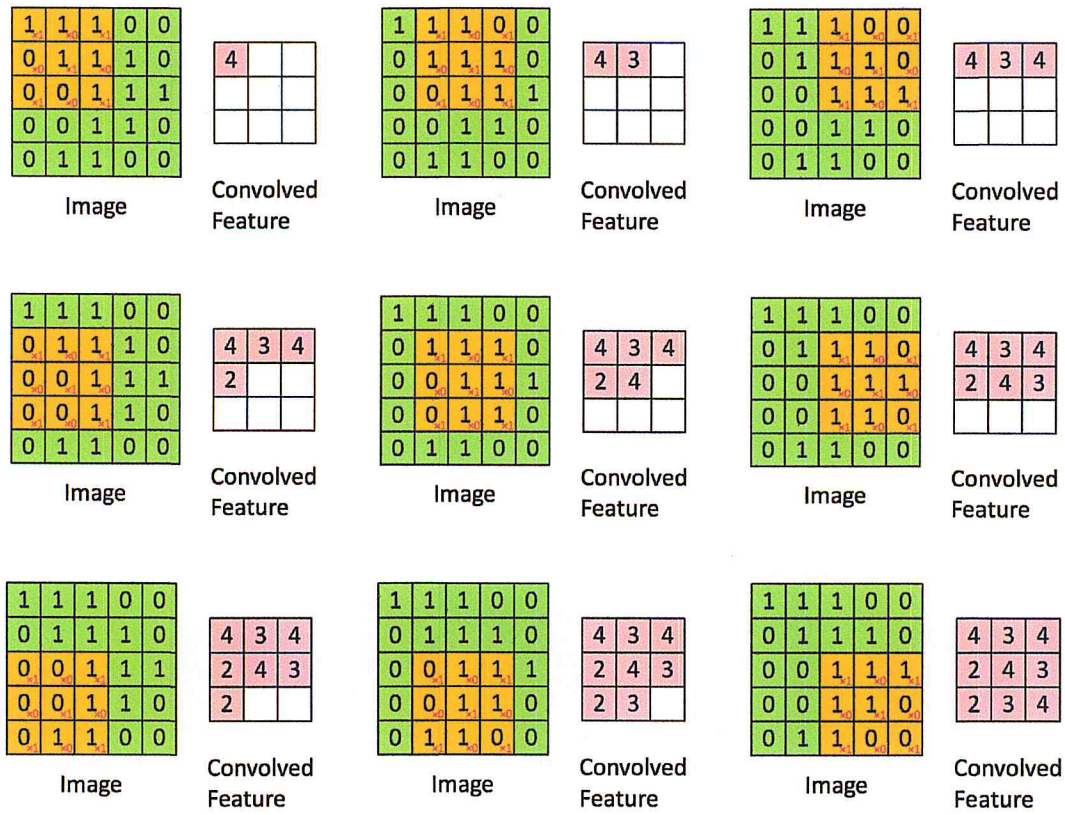


Figure 4-4: The feature extraction process in a CNN.

Taking this example, the matrix on the left represents an image, and to make it simple we say that it's a black and white image. Each entry in the matrix represents one pixel, 0 for black and 1 for white. As the Figure 4-4 shows, we're using a 3x3 filter that multiplies its values element-wise with the original matrix then sums up. We get the full convolution by sliding the filter over the whole matrix and doing the same procedure for each element.

4.2.3.2 Convolutional Neural Networks Model

Inspired from [161], [162] created the model shown in Figure 4-5 for sentence classification purposes and it's from this model that we're going to build the convolutional part of our model.

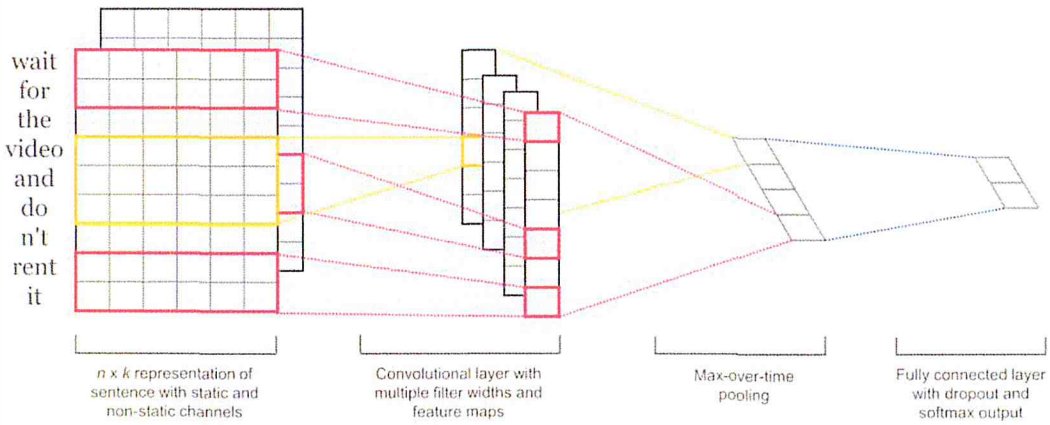


Figure 4-5: CNN model architecture.

Let $x_i \in R^k$ be the k-dimensional word vector corresponding to the i-th word in the sentence. A sentence of length n (padded where necessary) is represented as

$$x_{1:n} = x_1 \oplus x_2 \oplus \ldots \oplus x_n$$

where $\oplus$ is the concatenation operator. In general, let $x_{i:i+j}$ refer to the concatenation of words $,xi+1,\ldots,xi+j$. A convolution operation involves a filter $w \in R^{hk}$, which is applied to a window of h words to produce a new feature. For example, a feature ci is generated from a window of words $x_{i:i+h-1}$ by

$$c_i = f(w \cdot x_{i:i+h-1} + b)$$

Here $b \in R$ is a bias term and f is a non-linear function such as the hyperbolic tangent. This filter is applied to each possible window of words in the sentence $\{x_{1:h}, x_{2:h+1}, \ldots, x_{n-h+1:n}\}$ to produce a feature map

$$c = [c_1, c_2, \ldots, c_{n-h+1}]$$

with $c \in R^{n-h+1}$.We then apply a max-overtime pooling operation [161] over the feature map and take the maximum value $\hat{c} = max\{c\}$ as the feature corresponding to this particular filter. The idea is to capture the most important feature—one with the highest value—for each feature map. This pooling scheme naturally deals with variable sentence lengths.

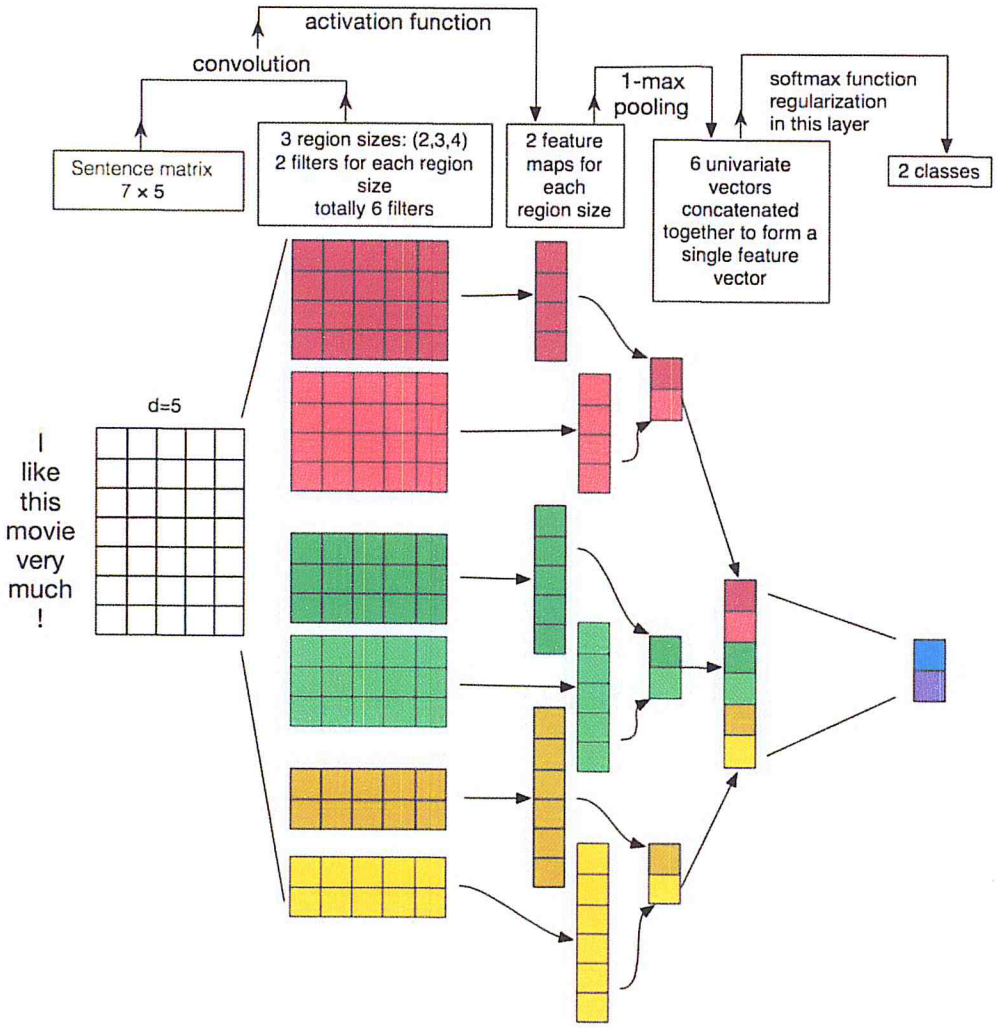


Figure 4-6: The steps of convolutions in a CNN.

Here we went through the step-by-step process through which one feature is extracted from one filter, but as Figure 4-6 shows, the CNN model uses multiple filters (with varying window sizes) to obtain multiple features. The layer formed by these features is called *penultimate layer*, and is connected to a final *softmax layer* that provides the probability distribution over labels [162].

Same as it was described in the previous section, in our model, we're going to start with a tokenized tweet which we then convert to a tweet matrix, the rows of which are word vector representations of each token. These are the outputs of the embedding layer previously defined. According to [163], we can then effectively treat the tweet matrix as an image, and perform convolution on it using linear filters.

In text applications there is inherent sequential structure to the data. Because rows represent discrete symbols (namely, words), it is reasonable to use filters with widths equal to the dimensionality of the word vectors (i.e., d). Thus we can simply vary the 'height' of the filter, i.e., the number of adjacent rows considered jointly. We will refer to the height of the filter as the region size of the filter.

The only difference between the standard CNN model and our model is that we don't have a softmax layer but we have instead a BiLSTM with Attention layer through which the features will be inputted, and we're going to cover the process of the latter in the next section.

4.2.4 BiLSTM with Attention Layer

4.2.4.1 RNNs

Recurrent Neural Networks (RNNs) are a class of neural networks whose connections between neurons form a directed cycle. Comparing it with feedforward neural networks, RNNs are distinguished by having a "memory" that is used for processing sequential information. Memory here is defined by performing *the same task for every element of a sequence with each output being dependent on all previous computations*, in order words, it's remembering information about what has been processed so far, which makes them very efficient and more human-brain-like [159].

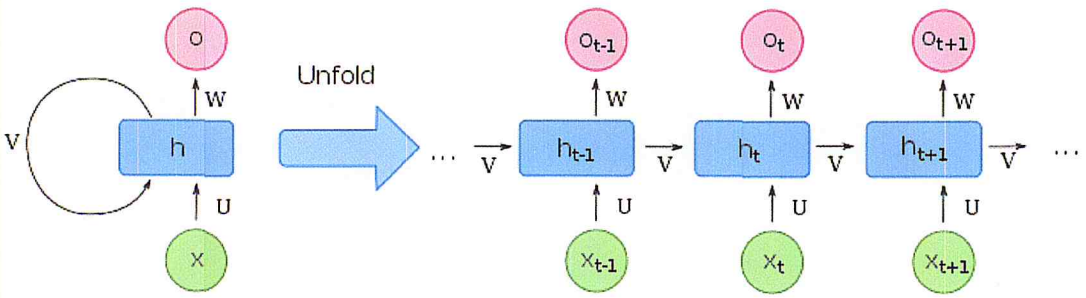


Figure 4-7: An unfolded basic recurrent neural network.

Figure 4-7: An unfolded basic recurrent neural network. shows an unfolded basic recurrent neural network. "Unfolded" simply means that we write out the network for the complete sequence, taking a tweet of 7 words as an example, the network would be unfolded into a 7-layer neural network, one layer for each word [164].

The pillar variables of an RNN, as highlighted in Figure 4-7, are defined as the following:

- x_t is the input at time step t. For example, x_7 can be the vector corresponding to the eighth word of the tweet.
- s_t is the hidden state at time step t. This is what we described previously as the "memory" of the RNN. It is calculated based on the previous hidden state and the input at the current step $s_t = f(U_{x_t} + W_{s_{\{t-1\}}})$ f is usually a non-linear function, such as tanh or ReLU.
- o_t is the output at step t. For example, if we wanted to predict the next word in a sentence it would be a vector of probabilities across our vocabulary. $o_t = softmax(V_{s_t})$.

There are a few things to note here:

- You can think of the hidden state s_t as the memory of the network. s_t captures information about what happened in all the previous time steps. The output at step o_t is calculated solely based on the memory at time t. As briefly mentioned above, it's a bit more complicated in practice because s_t typically can't capture information from too many time steps ago.
- Unlike a traditional deep neural network, which uses different parameters at each layer, a RNN shares the same parameters (U, V, W above) across all steps. This reflects the fact that we are performing the same task at each step, just with different inputs. This greatly reduces the total number of parameters we need to learn.
- The above diagram has outputs at each time step, but depending on the task this may not be necessary. For example, when predicting the sentiment of a sentence we may only care about the final output, not the sentiment after each word. Similarly, we may not need inputs at each time step. The main feature of an RNN is its hidden state, which captures some information about a sequence.

4.2.4.2 LSTMs & BiLSTMs

We can't start talking about LSTMs before addressing the limitations of RNNs that triggered the invention of the former in the first place. RNNs success and popularity is due to its ability to connect previous information to the present task [164], something we called previously as "memory".

When the present task only needs recent information, this doesn't cause any obstacles but the older the information needed is the more this "memory" forgets about it. once the task requires older information, something we call "context", RNNs fail to identify it, in other words, to "remember" it.

To make it clearer, let's use a word prediction model as an example. If we take the sentence "the fish in the" independently and we try to predict the next word after it, it's obvious that it will be "sea". This doesn't need any further context, the gap between the relevant information and where it's needed is small and due to the "memory" of the RNN we are able to use past information "fish" to predict "sea". If we take the same sentence in a paragraph that starts with "I visited AquaDom…" and ends with "... the fish in the", recent information suggests that the next word is probably "sea" but if we take in consideration the context of AquaDom, it's probably "aquarium", and this is where RNNs fail because as the gap grows between the needed information and the current task, the connection between them fades away [165].

Long Short Term Memory (LSTM) networks on the other hand are a special kind of RNN that is designed specifically to solve RNNs' long-term dependency problem. Originally introduced by [166], LSTMs are being used in a wide variety of problems due to their ability to remember information over long periods of time.

As shown previously, standard RNNs have a chain-like form composed of repeating modules of a simple structure, such as a single *tanh* layer as Figure 4-8 shows it.

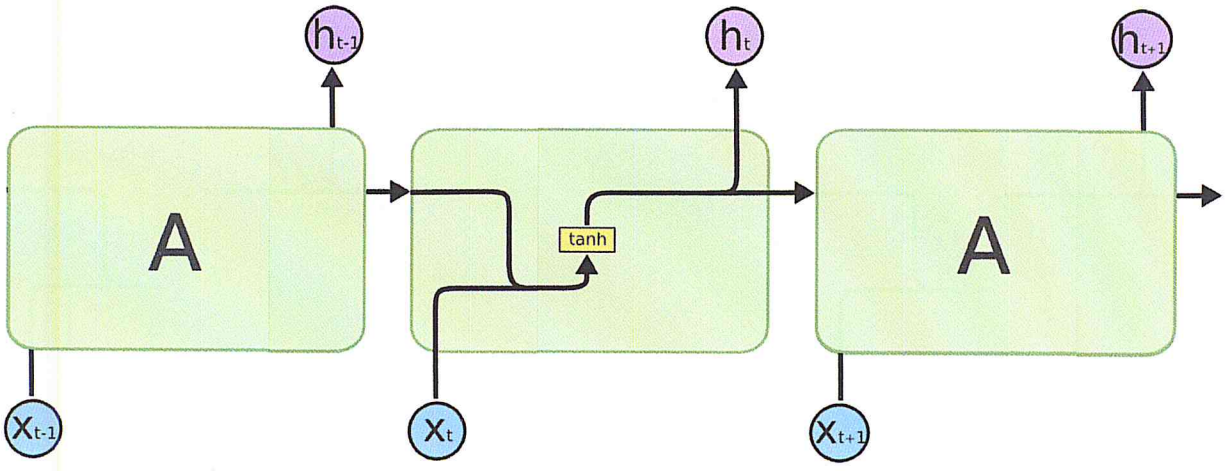


Figure 4-8 : A standard RNN [165].

Similarly to standard RNNs, LSTMs have the same structure, the difference is in the repeating modules, which contain a four-layer neural network instead of a single one. As Figure 4-9 shows it, *each line carries an entire vector, from the output of one node to the inputs of others* [165].

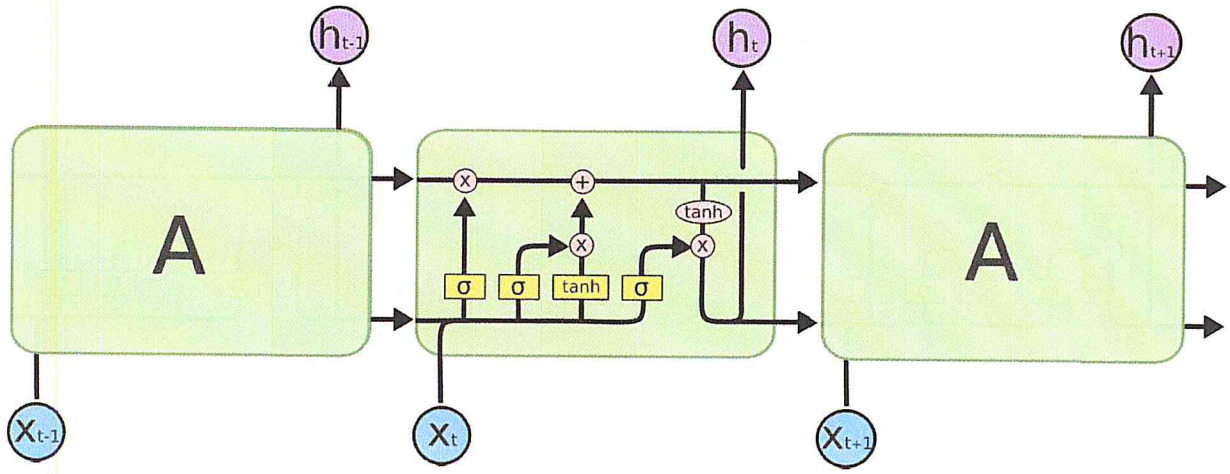


Figure 4-9 : An LSTM neural network [165].

The pink circles represent pointwise operations, like vector addition, while the yellow boxes are learned neural network layers.

The key element of LSTMs' long-term dependency solution is the cell state, highlighted in Figure 4-10. It plays the role of a conveyor belt as it runs through the whole network with minor linear interaction, allowing the information to flow smoothly without going through any changes.

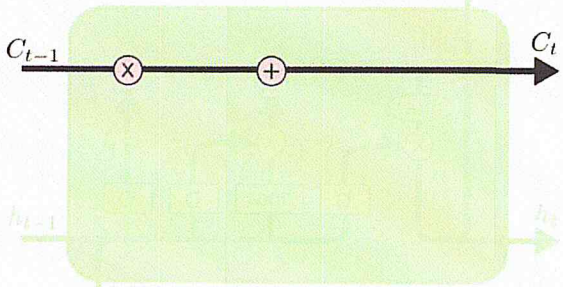


Figure 4-10 : The cell state of an LSTM network's module [165].

The information stored in the cell state can be removed or updated by the LSTM network through regulating gates that control the information passing through a sigmoid neural net layer and a pointwise multiplication operation. An LSTM protects and controls the cell state via three of these gates.

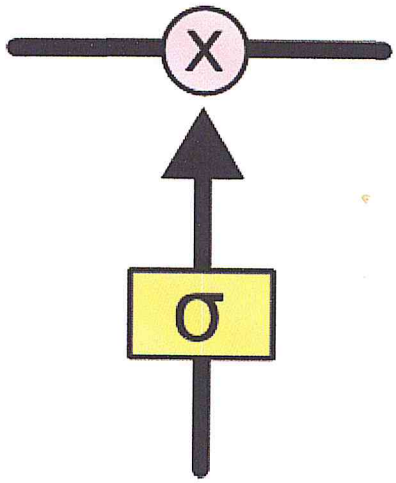
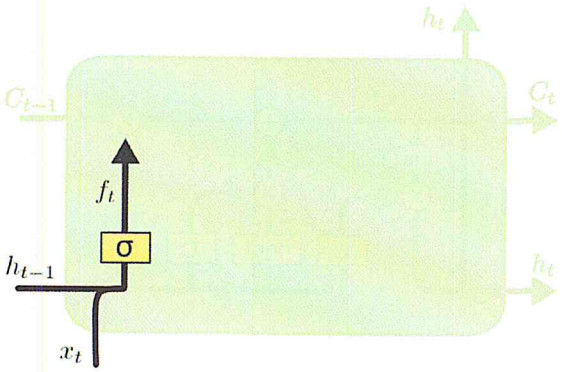


Figure 4-11 : An example of a gate [165].

Figure 4-11 is an example of a gate. The sigmoid layer gives in its output a value between zero and one, representing how much of each component should pass to the cell state. Zero means "nothing should pass" and one means "everything should pass".

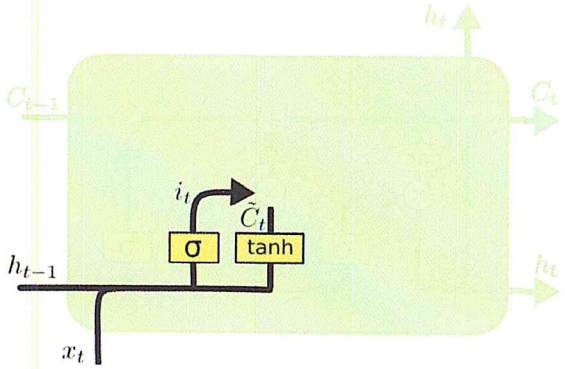
To understand better, let's go through a step-by-step explanation of how an LSTM works. We start by the first gate, which is found in the first layer of the recurrent module of the network, also known as the "*forget gate layer*". It is responsible of deciding what information should be forgotten, in other words, which information should be deleted from the cell state, in the same way described above.



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

Figure 4-12: The first step of an LSTM module [165].

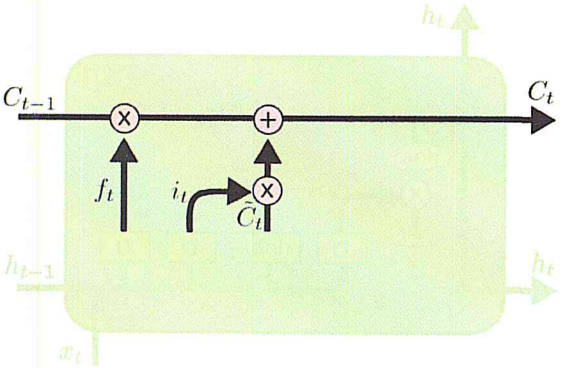
The next step consists of deciding what new information are we going to store in the cell state, and it's divided into two parts. First comes another sigmoid layer known as the "input gate layer" that decides which values we should update. Second, we have a tanh layer that creates a vector of new candidate values, $\tilde{C}_t$, that can be added to the state. This ends by combining the outputs of both layers and updating the state.



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

Figure 4-13 : The second step of an LSTM module [165].

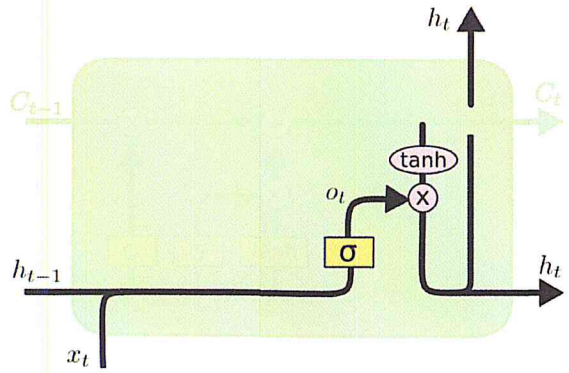
After that, we update the old cell state, C_{t-1}, into the new cell state C_t by using the decisions taken in the previous steps. The old state is multiplied by ft in order to forget what was decided in the first step. We add then $\tilde{C}_t$ to get the new candidate values according to all the decisions made.



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Figure 4-14 : The third step of an LSTM module.

Finally, a decision should be made about the output, and it will be based on a filtered version of the cell state. First, a sigmoid layer is run to decide what parts of the latter we're going to output, then, we put the cell state through tanh and multiply it by the output of the sigmoid gate to get only the parts that were decided to be output.



$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t]\ +\ b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

Figure 4-15 : The last step of an LSTM module [165].

So this basically how an LSTM works, and this kind of mechanism is what allows to it hold the most relevant information until the end of the procedure. Now that it is clear how a standard LSTM functions, we're going to explore a more complex model that we will be using for our project which is called the "MSA Model", which short for "Message-level Sentiment Analysis".

4.2.4.3 The MSA Model

As mentioned previously, our model is uses the MSA model of [167] as a part of it and its sequential layer consists of 2-layer bidirectional LSTM (BiLSTM) with an attention mechanism.

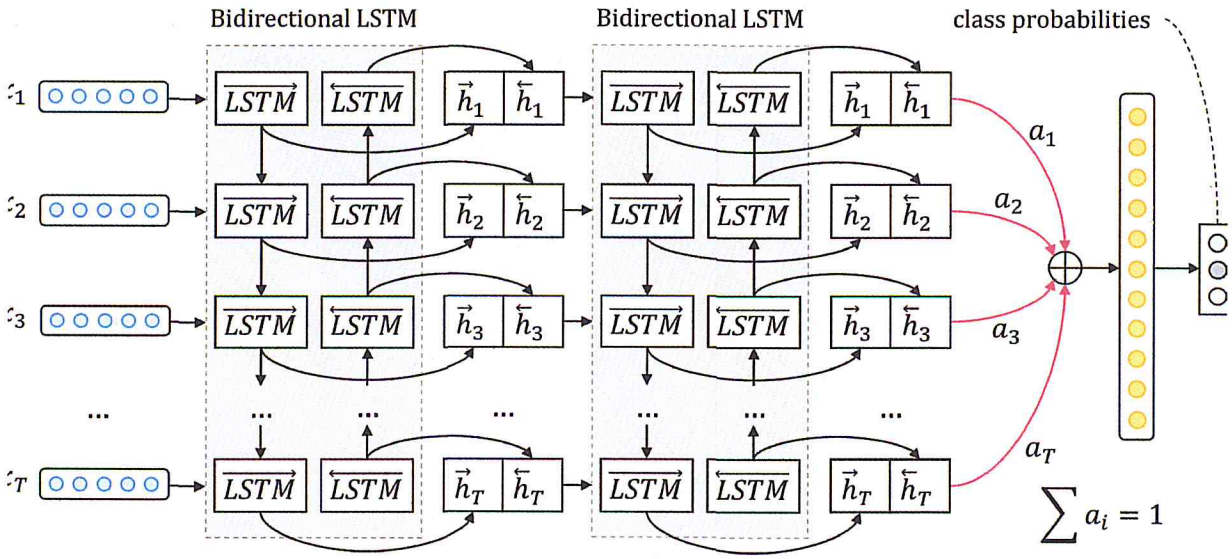


Figure 4-16 : The MSA model [167]: A 2-layer bidirectional LSTM with attention over that last layer.

The MSA model consists of the following layers [167]:

- **Embedding Layer:** In this layer, the Twitter message is fed to the network as a sequence of words that will be projected to a low-dimensional vector space RE, where E is the size of the embedding layer and T is the number of words in a tweet.

- **BiLSTM Layers**: In general, an LSTM takes the words of a tweet as an input and produces annotations $H = (h1, h2, \ldots, hT)$, *where hi is the hidden state o the LSTM at time-step i, summarizing all the information of the sentence up to* xi. For this case, the use of bidirectional LSTM (BiLSTM) brings the advantage of getting word annotations that summarize the information from both directions, first, going forward from $x1$ to xT, then backward from xT to $x1$. The final annotation of a given word is the the contactination of the annotations from both directions. Finally, the reason why two layers of BiLSTMs are used is to make the model learn more abstract features.

- **Attention Layer:** The uniqueness of this BiLSTM model is lies in this layer. Knowing that not all words contribute equally to the expressions of a sentiment in a message, the attention mechanism allows the model to find the relative importance of each word to the expression by assigning a weight ai to each word annotation then computing the fixed representation r of the whole message as the weighted sum of all the word annotations.

4.2.5 Softmax Layer

In the final layer, the representation r is fed to the final fully-connected softmax layer as a feature vector used for classification, the result will be a probability distribution over all classes.

4.3 Conclusion

To conclude, we have gone through the different phases and layers of our proposed model. Most importantly, we dug deep in each part to give a better understanding of the basics of all the models used, this way everything is justified.

In the next chapter, we're going to describe the process we went through in order to implement this model and the results that we got at the end.

Chapter 5 Experiments and Results

5.1 Introduction

Now that we have our model designed, it's time to turn theory into practice and evaluate our proposed architecture. We will start first by presenting the experimental setup that we used to develop our model and conduct our tests, then we share more details about the data used in our experiments and the process followed, and by the end, we share the results we got and we compare them with other results from a similar study.

5.2 Experimental Setup

5.2.1 Hardware

In such studies, hardware plays a very crucial role, especially when it comes to how much data can be handled for the given time and how much time does it take to train the deep neural network and evaluate it. Therefore, we feel the need to state what type of machine was used for the experiments that were done and what are its characteristics.

All the experiments described in this chapter we conducted on a machine with an Intel(R) Core(™) i5-5200U CPU @ 2.20GHz x64-based processor and a 4.00GB RAM. In addition to that, around 205 GB of free space were needed in order to store the dataset used.

5.2.2 Development Environment

To implement that models designed and run all the programs and tests, we used JetBrains PyCharm Community Edition 2018.1.2 development environment [168]. PyCharm is an intelligent Python IDE (Integrated Development Environment) with a huge collection of tools out of the box, including an integrated debugger and test runner, Python profiler, a built-in terminal, and much more.

Python version 3.5.2 was used in our work. We chose Python because of different reasons. First, its simplicity is life saving, it makes expressing complex equations and formulas, such as the ones needed in our project, much easier, which makes the whole development process less time-consuming. Second, its diverse and rich libraries, the ones dedicated for deep learning and the ones used for other data structures handling and other matters. Third, with Python, we don't need to worry about memory management since it does it for us.

5.2.3 Librairies

As it was mentioned in the previous section, one of the reasons that Python was used for this project is the amount of important libraries that exist, therefore, we feel the need to present the most important ones that were used to make our model a reality.

5.2.3.1 TensorFlow

TensorFlow™ [169] is an open source software library for high performance numerical computation. Its flexible architecture allows easy deployment of computation across a variety of platforms (CPUs, GPUs, TPUs), and from desktops to clusters of servers to mobile and edge devices. Originally developed by researchers and engineers from the Google Brain team within Google's AI organization, it comes with strong support for machine learning and deep learning and the flexible numerical computation core is used across many other scientific domains.

5.2.3.2 NumPy

NumPy [170] is the fundamental package for scientific computing with Python. Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

5.2.3.3 NLTK

NLTK [171] is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active discussion forum.

5.2.3.4 Gensim

Gensim [172] is a Python library for topic modelling, document indexing and similarity retrieval with large corpora. Target audience is the natural language processing (NLP) and information retrieval (IR) community.

5.2.3.5 Panda

Pandas [173] is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language.

5.2.3.6 SeaBorn

Seaborn [174] is a Python visualization library based on matplotlib. It provides a high-level interface for drawing attractive statistical graphics.

5.2.4 Source Code

The following are some excerpts from the source code that showcase how were some of the important parts of the model implemented.

```
num_features = 300

min_word_count = 3

num_workers = multiprocessing.cpu_count()

context_size = 7

downsampling = 1e-3

seed = 1

tweets2vec = w2v.Word2Vec(
    sg=1,
    seed=seed,
    workers=num_workers,
    size=num_features,
    min_count=min_word_count,
    window=context_size,
    sample=downsampling
)

tweets2vec.build_vocab(sentences)

print("Word2Vec vocabulary length: ", len(tweets2vec.wv.vocab))

tweets2vec.train(sentences, total_examples=tweets2vec.corpus_count, epochs=tweets2vec.iter)

if not os.path.exists('trained'):
    os.makedirs('trained')

tweets2vec.save(os.path.join('trained', 'depressed.w2v'))
```

Figure 5-1 : The source code of Word2Vec model implementation.

```
# Create a convolution + maxpool layer for each filter size
pooled_outputs = []
for i, filter_size in enumerate(filter_sizes):
    with tf.name_scope("conv-maxpool-%s" % filter_size):
        # Convolution Layer
        filter_shape = [filter_size, embedding_size, 1, num_filters]
        W = tf.Variable(tf.truncated_normal(filter_shape, stddev=0.1), name="W")
        b = tf.Variable(tf.constant(0.1, shape=[num_filters]), name="b")
        conv = tf.nn.conv2d(
            self.embedded_chars_expanded,
            W,
            strides=[1, 1, 1, 1],
            padding="VALID",
            name="conv")
        # Apply nonlinearity
        h = tf.nn.relu(tf.nn.bias_add(conv, b), name="relu")
        # Maxpooling over the outputs
        pooled = tf.nn.max_pool(
            h,
            ksize=[1, sequence_length - filter_size + 1, 1, 1],
            strides=[1, 1, 1, 1],
            padding='VALID',
            name="pool")
        pooled_outputs.append(pooled)
```

Figure 5-2 : The source code of the CNN model implementation.

```
model = Sequential()
model.add(embeddings_layer(max_length=max_length, embeddings=embeddings,
                           trainable=False, masking=True, scale=False,
                           normalize=False))

if noise > 0:
    model.add(GaussianNoise(noise))
if dropout_words > 0:
    model.add(Dropout(dropout_words))

for i in range(layers):
    rs = (layers > 1 and i < layers - 1) or attention
    model.add(get_RNN(unit, cells, bi, return_sequences=rs,
                      dropout_U=dropout_rnn_U))
    if dropout_rnn > 0:
        model.add(Dropout(dropout_rnn))

if attention == "memory":
    model.add(AttentionWithContext())
    if dropout_attention > 0:
        model.add(Dropout(dropout_attention))
elif attention == "simple":
    model.add(Attention())
    if dropout_attention > 0:
        model.add(Dropout(dropout_attention))

if final_layer:
    model.add(MaxoutDense(100, W_constraint=maxnorm(2)))
    # model.add(Highway())
    if dropout_final > 0:
        model.add(Dropout(dropout_final))

model.add(Dense(classes, activity_regularizer=l2(loss_l2)))
model.add(Activation('softmax'))
```

Figure 5-3 : The source code of the MSA model's implementation.

5.3 Evaluation

5.3.1 Data

Due to the limitations of access and to the narrow category of people that we're targeting with our work, before we do anything we needed to check the available data and build our architecture based upon it instead of proposing an architecture then collecting the data for it.

For that, we used a dataset that was used in a very similar study [145], which is constructed of three (3) parts: Depression dataset, non-Depression dataset, and Depression-candidate dataset.

5.3.1.1 Depression Dataset

The first dataset, named D1, is based on tweets between 2009 and 2016 and only contains tweet of people who used the anchor tweet "(I'm/I was/I am/I've been) diagnosed with depression" and that was inspired from [146] work, where they suggested that this tweet is enough to categorizing someone as depressed. With that being said, 1402 depressed users have been obtained making the dataset as 292564 within one month.

5.3.1.2 Non-Depression Dataset

The second dataset, named D2, is kind of the opposite of the first one, as it contains that tweets of users that never posted any tweet that contains the character string "depress". The tweets that were selected were of December 2016 and it resulted in more than 10 billion tweets of more than 300 millions users.

5.3.1.3 Depression-candidate Dataset

Last but not least, the final dataset, D3, contains people who might potentially have depression, and that is characterized by people whose anchor tweet contained the character string "depress" and they ended up with 36993 depression candidates with over 35 millions tweets.

To sum it all up, this dataset is exactly what we have been looking for and given the variety of sub-datasets and their preciseness we were able to propose an architecture that is more user-oriented and focused on depression instead of sentiment polarity in general.

5.3.2 Performance Measures

Now that we have our architecture ready, before we move on to implementing it we need to define the metrics through which we will measure the results of our work and compare it to other works.

To do that, we need to introduce something called the confusion matrix and the different variables that we will use to calculate all our performance measures [175].

		Predicted	
		Negative	Positive
Actual	Negative	True Negative	False Positive
	Positive	False Negative	True Positive

Table 5-1 : The confusion matrix.

- **True Positives (TP)** - These are the correctly predicted positive values which means that the value of actual class is yes and the value of predicted class is also yes.
- **True Negatives (TN)** - These are the correctly predicted negative values which means that the value of actual class is no and value of predicted class is also no.
- **False Positives (FP)** – When actual class is no and predicted class is yes.
- **False Negatives (FN)** – When actual class is yes but predicted class in no.

5.3.2.1 Precision

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. The question that this metric answer is: How many of the users we identified as depressed are actually depressed? High precision relates to the low false positive rate.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

5.3.2.2 Recall

Recall is the ratio of correctly predicted positive observations to the all observations in actual class - yes. The question recall answers is: Out of all of the depressed users, how many did we properly detect?

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

5.3.2.3 F1-Score

F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if you have an uneven class distribution. Accuracy works best if false positives and false negatives have similar cost. If the cost of false positives and false negatives are very different, it's better to look at both Precision and Recall.

$$F1\ Score = 2 \times \frac{Recall \times Precision}{Recall + Precision}$$

5.3.2.4 Accuracy

Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations. One may think that, if we have high accuracy then our model is best. Yes, accuracy is a great measure but only when you have symmetric datasets where values of false positive and false negatives are almost same. Therefore, you have to look at other parameters to evaluate the performance of your model.

$$Accuracy = \frac{True\ Positive + True\ Negative}{True\ Positive + False\ Positive + False\ Negative + True\ Negative}$$

5.3.3 Training

In this section, we will go through the detailed process that we went through to train our model. Since our model includes different types of neural network, we will go through each of them separately.

In order to get the best out of this project, we didn't limit ourselves to training only our model but we experimented with different approaches so that we get rich insights by the end of the study. Since we have distinct datasets that are depression oriented, using a supervised learning approach was the best option we could take. Therefore, instead of using a pre-trained word embedding, we trained our own word embeddings. We also trained a CNN and the MSA model separately using our depression-trained word embeddings to see how efficient they are and what improvements does our model bring.

5.3.3.1 Training the Word2Vec neural network

Although there is a wide variety of pre-trained sentiment analysis word embeddings, we wanted to make our deep neural network more depression oriented, therefore, we decided to train our own word embeddings using the datasets D1 and D2. As it was mentioned above, D1 contains tweets of depressed people, therefore, it allows us to detect the similarities between the words and expressions they use by generating more relevant word vectors to our study. Same goes for D2, but we didn't mix them, we generated the vectors for each dataset separately so that we used them later depending on our needs.

5.3.3.2 Training the CNN

Even though we met a study that tested a CNN-only model for depression detection, we wanted to experiment with the our own data and word embeddings, therefore, our first experiment was to train a CNN-only model with the results we got from the previous step to see how efficient CNNs are, when used individually.

5.3.3.3 Training the MSA model

Our second experiment was to train the MSA model that we mentioned in the previous chapter, individually as well, using our depression-trained word embeddings to see if the BiLSTM deep neural network is able to detect depression with high efficiency.

5.3.3.4 Training our model

Last but not least, the most important experiment, and the one all of this study was done for, we used the word embeddings we trained to feed the first part of our model, which is the CNN, and then we took the vectors output from it and fed it to the MSA model.

5.3.4 Results

The following table showcases the results we got from each experiment in addition to the results that were shared from [145], which used the same datasets as us, but with statistical models instead.

Model	Accuracy
Naive Bayesian	73%
Multiple Social Networking Learning	82%
Wasserstein Dictionary Learning	77%
Multimodal Depressive Dictionary Learning	85%
CNN	98%
MSA	98%
CNN+MSA	**99%**

Table 5-2 : The results of the different experiments conducted on the dataset we used.

As it is shown in the table, we notice that deep learning methods are way more effective than any traditional methods, no matter how advanced and tailored it is. Most importantly, we see that the deeper the neural network is the better the learning is, the better results we get.

5.4 Conclusion

To conclude, this experimentation process was the most interesting part of this whole study as we got to experiment with different methods and tweak different parameters in order to improve the performance. With that being said, it was the most time-consuming task to do, as it took several hours just to try with one tweak in the training or evaluation to see different results.

General Conclusion

The objective of our study was to solve one of mental health's biggest issues, which is the inefficiency of the traditional identification methods of mental illnesses, more specifically, we chose to focus on major depressive disorder, also known as clinical depression, because of its major popularity and likelihood of spreading, and most importantly, to prevent the tragedies that might occur from such disorder. For that, we wanted to design and propose a new way of identifying depression using an advanced artificial intelligence technology known as deep learning.

In order to achieve that, we reviewed as many studies about the matter as we could. First, we conducted a research about depression and its traditional identification methods. We then reviewed the different projects that tackled the same problematic as ours, more specifically, works that were computer science related. Such research introduced us to different approaches and allowed us to get a clear perspective about where does the current solutions stand. It allowed us as well to notice that technologies such as deep learning weren't taken advantage of for the sake of solving these problems and tackling such life-saving matters.

Once we got a clear perspective on the current realities, we proceeded to see where technology stands for similar or related issues. For that, we started by conducting a research on sentiment analysis and reviewing a wide variety of works related, but not limited, to mental health. We then advanced to review studies that involved deep learning in this fast-growing niche, we could see that it is an area getting the attention of many research projects but almost no application in mental health was found. Matter of fact, the only study published that included the latter was published during the period of our research, which shows that only recently people are starting to pay attention to such issues. Of course, we didn't limit ourselves by staying only on this field, we tried to go through as many works as we could, not specifically related to our problematic. This includes text classification and pattern recognition models, all using deep learning. By the end of this process, we had enough understanding of the all those technologies and the current state of the art to be able to develop and propose our own model to reach our main objective.

Our next step then was to propose our own model and use the knowledge we acquired during the literature review to predict the most suitable approach to solve our problematic and achieve our goal. For that, we took advantage of an existing model that was used previously strictly for sentiment analysis, and we tried to include it in our depression detection model with our unique enhancements. Our model mainly consists of two types of deep neural networks, a convolutional neural network and a recurrent one. This combination we believed, in addition to a depression-trained word embeddings layer in entrance, could predict better results than any of the works found in the literature thus far.

Finally, we implemented our model using a humble experimental setup. We used a dataset developed specifically for depression studies, which allowed us to take a supervised learning approach. This helped us conduct different experiments and compare their results in order to discover which technique is the most efficient. We compare our experiments by the end with the results of another study that used the same data, but different methods, more specifically, traditional ones. We proved through that that deep learning is much more efficient than the older statistical (or any other) classification techniques, which makes it more qualified to tackle neglected world problems, such as the one we chose to pursue, not only for its current outstanding results, but for its promising future as well.

5.5 Contributions

Our main contributions with this work lie in the three following points:

- We trained our own word embeddings using a depression-dedicated dataset.
- We combined a CNN layer with the MSA model in order to improve the feature extraction process and enhance the model's performance.
- We analyzed through different experiments the performance of three deep learning models in order to provide more perspectives and insights for depression researches.

5.6 Perspectives

As future perspectives, we see this study as a first step towards unveiling new ways of identifying depression, and other mental illnesses, not only in Twitter or social media, but also in different type of platforms and applications that can be used in day-to-day basis.

First, we suggest that a more complex should be developed from ours, this new one should be able to take in consideration not only the tweets' text but also their timing and other metadata as well, furthermore, a user-based, instead of a tweet-based model will probably get better results, in terms of

diagnosis. Such model should be able to treat more than the tweets of a given user, but all his details as well, this includes: country, profile picture, number of followers, color of the profile, age… etc and all possible details that can be retrieved from a Twitter account.

Second, we believe that our model can be used with different mental illnesses, not only depression. The only part missing to prove that is datasets dedicated specifically for various mental disorders. If one manages to develop one, we believe that there is a possibility of getting the same, or very close, results as ours. We also believe that our model can be used on any corpus as long as it contains self-expressed statements and it will provide the same results.

Finally, our most hopeful and futuristic perspective is for the model to be enhanced to be able to detect the degree of depression an individual is suffering from. If such possibility becomes a reality, then the uses of the model will grow exponentially. One proposition we offer future researchers interested by the subject, is to create a journaling platform for depression patients in order to follow up and report their state of progress to their doctors using that last suggested model.

To conclude, this project has been a quite rewarding learning experience because it introduced me to new unfamiliar technologies and allowed me to acquire new skills and master existing ones.

References

[1] Cambridge Dictionary, "Meaning in the Cambridge English Dictionary," Cambridge Dictionary, 2018. [Online]. Available: https://dictionary.cambridge.org/dictionary/english/sentiment. [Accessed: 23-Jun-2018].

[2] Cambridge Dictionary, "Meaning in the Cambridge English Dictionary." 2018.

[3] B. Liu, Sentiment Analysis and Opinion Mining, no. May. 2012.

[4] K. Dave, K. Dave, S. Lawrence, S. Lawrence, D. M. Pennock, and D. M. Pennock, "Mining the peanut gallery: Opinion extraction and semantic classification of product reviews," Proc. 12th Int. Conf. World Wide Web, pp. 519–528, 2003.

[5] T. Nasukawa, T. Nasukawa, J. Yi, and J. Yi, "Sentiment analysis: Capturing favorability using natural language processing," Proc. 2nd Int. Conf. Knowl. capture, pp. 70–77, 2003.

[6] S. Das and M. Chen, "Yahoo! for Amazon: Extracting market sentiment from stock message boards," Proc. Asia Pacific Financ. ..., vol. 35, p. 43, 2001.

[7] S. Morinaga, K. Yamanishi, K. Tateishi, and T. Fukushima, "Mining product reputations on the Web," in Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '02, 2002, p. 341.

[8] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: sentiment classification using machine learning techniques," Empir. Methods Nat. Lang. Process., vol. 10, no. July, pp. 79–86, 2002.

[9] R. Tong, "An operational system for detecting and tracking opinions in on-line discussion," Work. Notes ACM SIGIR 2001 Work. Oper. Text Classif., pp. 1–6, 2001.

[10] P. D. Turney, "Thumbs up or thumbs down?," in Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02, 2001, no. July, p. 417.

[11] J. M. Wiebe, "Learning subjective adjectives from corpora," Proc. Natl. Conf. Artif. Intell., no. 1, pp. 735–741, 2000.

[12] M. McGlohon, N. Glance, and Z. Reiter, "Star Quality : Aggregating Reviews to Rank Products and Merchants," Wall Str. J., pp. 114–121, 2010.

[13] Y. Hong and S. Skiena, "The Wisdom of Bookies? Sentiment Analysis Versus. the NFL Point Spread.," Fourth Int. AAAI Conf. Weblogs Soc. Media, pp. 251–254, 2010.

[14] B. Chen, L. Zhu, D. Kifer, and D. Lee, "What Is an Opinion About? Exploring Political Standpoints Using Opinion Scoring Model," Twenty-Fourth AAAI Conf. Artif. Intell., pp. 1007–1012, 2010.

[15] M. Miller, C. Sathi, D. Wiesenthal, J. Leskovec, and C. Potts, "Sentiment Flow Through Hyperlink Networks.," Fifth Int. AAAI Conf. Weblogs Soc. Media, pp. 550–553, 2011.

[16] B. O 'connor, R. Balasubramanyan, B. R. Routledge, and N. A. Smith, "From Tweets to Polls: Linking Text Sentiment to Public Opinion Time Series."

[17] S. Mohammad and T. Yang, "Tracking Sentiment in Mail: How Genders Differ on Emotional Axes," Proc. ACL 2011 Work. Comput. Approaches to Subj. Sentim. Anal., pp. 70–79, 2011.

[18] S. M. Mohammad, "From once upon a time to happily ever after: Tracking emotions in mail and books," in Decision Support Systems, 2012, vol. 53, no. 4, pp. 730–741.

[19] R. Bar-Haim, E. Dinur, R. Feldman, M. Fresko, and G. Goldstein, "Identifying and following expert investors in stock microblogs," Proc. Conf. Empir. Methods Nat. Lang. Process., pp. 1310–1319, 2011.

[20] R. Feldman, B. Rosenfeld, R. Bar-Haim, and M. Fresko, "The Stock Sonar — Sentiment Analysis of Stocks Based on a Hybrid Approach," Iaai, no. c, pp. 1642–1647, 2011.

[21] P. Sakunkoo and N. Sakunkoo, "Analysis of Social Influence in Online Book Reviews," AAAI's ICWSM 2009, pp. 1968–1970, 2009.

[22] G. Groh and J. Hauffa, "Characterizing Social Relations Via NLP-based Sentiment Analysis," Proc. Fifth Int. AAAI Conf. Weblogs Soc. Media, pp. 502–505, 2011.

[23] A. Tumasjan, T. Sprenger, P. Sandner, and I. Welpe, "Predicting elections with Twitter: What 140 characters reveal about political sentiment," Proc. Fourth Int. AAAI Conf. Weblogs Soc. Media, pp. 178–185, 2010.

[24] T. Yano and N. A. Smith, "What ' s Worthy of Comment ? Content and Comment Volume in Political Blogs *," Fourth Int. AAAI Conf. Weblogs Soc. Media, vol. i, pp. 359–362, 2010.

[25] Y. Liu, J. Huang, A. An, and X. Yu, "ARSA: A sentiment-aware model for predicting sales performance using blogs," Proc. ACM Spec. Interes. Gr. Inf. Retr. (SIGIR, 2007.

[26] S. Asur and B. A. Huberman, "Predicting the future with social media," Proc. Int. Conf. Web Intell. Intell. Agent Technol., pp. 492–499, 2010.

[27] M. Joshi, D. Das, K. Gimpel, and a. N. Smith, "Movie Reviews and Revenues: An Experiment in Text Regression," Hum. Lang. Technol. 2010 Annu. Conf. North Am. Chapter Assoc. Comput. Linguist., pp. 293–296, 2010.

[28] E. Sadikov, A. Parameswaran, and P. Venetis, "Blogs as predictors of movie success," Third Int. AAAI Conf. Weblogs Soc. Media, pp. 304–307, 2009.

[29] J. Bollen, H. Mao, and X.-J. Zeng, "[1010.3003] Twitter mood predicts the stock market." [Online]. Available: http://arxiv.org/abs/1010.3003. [Accessed: 23-Jun-2018].

[30] W. Zhang and S. Skiena, "Trading Strategies to Exploit Blog and News Sentiment.," Icwsm, vol. 34, no. Chan 2003, pp. 375–378, 2010.

[31] "Introduction to Natural Language Processing (NLP) - Algorithmia Blog," Algorithmia Blog, 2016. [Online]. Available: https://blog.algorithmia.com/introduction-natural-language-processing-nlp/. [Accessed: 27-Jun-2018].

[32] C. C. Aggarwal and C. X. Zhai, "Mining text data," Min. Text Data, vol. 9781461432234, pp. 1–522, 2013.

[33] M.-E. G. Rossi, F. D. Malliaros, and M. Vazirgiannis, "Spread it good, spread it fast: Identification of influential nodes in social networks," Proc. 24th Int. Conf. World Wide Web, pp. 101–102, 2015.

[34] J. M. Wiebe, R. F. Bruce, and T. P. O'Hara, "Development and use of a gold standard data set for subjectivity classifications," in Proceedings of the Association for Computational Linguistics, 1999, pp. 246–253.

[35] M. Hu and B. Liu, "Mining and summarizing customer reviews," in Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '04, 2004, p. 168.

[36] P. Gonçalves, M. Araújo, F. Benevenuto, and M. Cha, "Comparing and Combining Sentiment Analysis Methods," 2014.

[37] Cool-Smileys, "List of Text Emoticons: The Ultimate Resource." 2010.

[38] Y. R. Tausczik and J. W. Pennebaker, "The psychological meaning of words: LIWC and computerized text analysis methods," Journal of Language and Social Psychology, vol. 29, no. 1. pp. 24–54, 2010.

[39] M. Thelwall, "Heart and soul: Sentiment strength detection in the social web with sentistrength," Proc. CyberEmotions, vol. 5, pp. 1–14, 2013.

[40] A. Esuli and F. Sebastiani, "SENTIWORDNET: A Publicly Available Lexical Resource for Opinion Mining," Proc. 5th Conf. Lang. Resour. Eval., pp. 417–422, 2006.

[41] G. A. Miller, "WordNet: a lexical database for English," Commun. ACM, vol. 38, no. 11, pp. 39–41, 1995.

[42] E. Cambria, R. Speer, C. Havasi, and A. Hussain, "SenticNet : A Publicly Available Semantic Resource for Opinion Mining," Artif. Intell., vol. 10, pp. 14–18, 2010.

[43] P. S. Dodds and C. M. Danforth, "Measuring the happiness of large-scale written expression: Songs, blogs, and presidents," J. Happiness Stud., vol. 11, no. 4, pp. 441–456, 2010.

[44] M. M. Bradley and P. P. J. Lang, "Affective Norms for English Words (ANEW): Instruction Manual and Affective Ratings," Psychology, vol. Technical, no. C-1, 1999.

[45] D. Watson, L. A. Clark, and A. Tellegen, "Development and validation of brief measures of brief measures of positive and negative affect: the PANAS scales.," J. Pers. Soc. Psychol., vol. 54, no. 6, pp. 1063–1070, 1988.

[46] D. L. Poole, A. Mackworth, and R. G. Goebel, "Computational Intelligence and Knowledge," Comput. Intell. A Log. Approach, no. Ci, pp. 1–22, 1998.

[47] V. Maini, "A Beginner's Guide to AI/ML – Machine Learning for Humans – Medium," Medium, 2017. [Online]. Available: https://medium.com/machine-learning-for-humans/why-machine-learning-matters-6164faf1df12. [Accessed: 23-Jun-2018].

[48] L. Dormehl, "What is an artificial neural network? Here's everything you need to know," Tech Radar, 2018. [Online]. Available: https://www.techradar.com/news/what-is-5g-everything-you-need-to-know. [Accessed: 23-Jun-2018].

[49] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436–444, 2015.

[50] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," Adv. Neural Inf. Process. Syst., vol. 60, no. 6, pp. 1–9, 2012.

[51] C. Farabet, C. Couprie, L. Najman, and Y. Lecun, "Learning Hierarchical Features for Scene Labeling," Pattern Anal. Mach. Intell. IEEE Trans., vol. 35, no. 8, pp. 1915–1929, 2013.

[52] J. Tompson, A. Jain, Y. LeCun, and C. Bregler, "Joint Training of a Convolutional Network and a Graphical Model for Human Pose Estimation," Adv. Neural Inf. Process. Syst., pp. 1799--1807, 2014.

[53] C. Szegedy et al., "Going deeper with convolutions," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2015, vol. 07–12–June, pp. 1–9.

[54] T. Mikolov, A. Deoras, D. Povey, L. Burget, and J. Černocký, "Strategies for training large scale neural network language models," 2011 IEEE Work. Autom. Speech Recognit. Understanding, ASRU 2011, Proc., pp. 196–201, 2011.

[55] G. Hinton et al., "Deep Neural Networks for Acoustic Modeling in Speech Recognition," IEEE Signal Process. Mag., vol. 29, no. 6, pp. 82–97, 2012.

[56] T. N. Sainath, A. Mohamed, B. Kingsbury, B. Ramabhadran, I. B. M. T. J. Watson, and Y. Heights, "Deep Convolutional Neural Networks for LVCSR," Icassp 2013, pp. 8614–8618, 2013.

[57] J. Ma, R. P. Sheridan, A. Liaw, G. E. Dahl, and V. Svetnik, "Deep neural nets as a method for quantitative structure-activity relationships," J. Chem. Inf. Model., vol. 55, no. 2, pp. 263–274, 2015.

[58] T. Ciodaro, D. Deva, J. M. De Seixas, and D. Damazio, "Online particle detection with neural networks based on topological calorimetry information," in Journal of Physics: Conference Series, 2012, vol. 368, no. 1.

[59] "Higgs Boson Machine Learning Challenge | Kaggle." [Online]. Available: https://www.kaggle.com/c/higgs-boson. [Accessed: 23-Jun-2018].

[60] M. Helmstaedter, K. L. Briggman, S. C. Turaga, V. Jain, H. S. Seung, and W. Denk, "Connectomic reconstruction of the inner plexiform layer in the mouse retina," Nature, vol. 500, no. 7461, pp. 168–174, 2013.

[61] M. K. K. Leung, H. Y. Xiong, L. J. Lee, and B. J. Frey, "Deep learning of the tissue-regulated splicing code," Bioinformatics, vol. 30, no. 12, pp. 121–129, 2014.

[62] H. Y. Xiong et al., "The human splicing code reveals new insights into the genetic determinants of disease," Science (80-.)., vol. 347, no. 6218, 2015.

[63] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural Language Processing (Almost) from Scratch," J. Mach. Learn. Res., vol. 12, pp. 2493–2537, 2011.

[64] A. Bordes, S. Chopra, and J. Weston, "Question Answering with Subgraph Embeddings," pp. 615–620, 2014.

[65] S. Jean, K. Cho, R. Memisevic, and Y. Bengio, "On Using Very Large Target Vocabulary for Neural Machine Translation," pp. 1–10, 2014.

[66] I. Sutskever, O. Vinyals, and Q. V Le, "Sequence to Sequence Learning with Neural Networks," in Proceedings of the 27th International Conference on Neural Information Processing Systems, 2014, pp. 3104–3112.

[67] R. Chengalvarayan and L. Deng, "Speech trajectory discrimination using the minimum classification error learning," IEEE Trans. Speech Audio Process., vol. 6, no. 6, pp. 505–515, 1998.

[68] M. Gibson and T. Hain, "Error Acoustic Model Estimation," vol. 18, no. 6, pp. 1269–1279, 2010.

[69] X. He, L. Deng, and W. Chou, "Discriminative learning in sequential pattern recognition: A unifying review for optimization-oriented speech recognition," IEEE Signal Processing Magazine, vol. 25, no. 5. pp. 14–36, 2008.

[70] H. Jiang and X. Li, "Parameter estimation of statistical models using convex optimization," in IEEE Signal Processing Magazine, 2010, vol. 27, no. 3, pp. 115–127.

[71] B. Juang, W. Chou, and C. Lee, "Minimum Classification Error Rate Methods For Speech Recognition - Speech and Audio Processing, IEEE Transactions on," Speech Audio Process. IEEE Trans., vol. 5, no. 3, pp. 257–265, 1997.

[72] D. Povey and P. C. Woodland, "Minimum Phone Error and I-Smoothing for Improved Discriminative Training," Proc.ICASSP, vol. 1, no. 1, pp. 105–108, 2002.

[73] L. Xiao and L. Deng, "A geometric perspective of large-margin training of gaussian models," IEEE Signal Process. Mag., vol. 27, no. 6, pp. 118–123, 2010.

[74] D. Yu, L. Deng, X. He, and A. Acero, "2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07," 2007 IEEE Int. Conf. Acoust. Speech Signal Process. - ICASSP '07, vol. 4, p. IV-1137-IV-1140, 2007.

[75] I. Heintz, E. Fosler-Lussier, and C. Brew, "Discriminative input stream combination for conditional random field phone recognition," IEEE Trans. Audio, Speech Lang. Process., vol. 17, no. 8, pp. 1533–1546, 2009.

[76] Y. Hifny and S. Renals, "Speech Recognition Using Augmented Conditional Random Fields," IEEE Trans. Audio. Speech. Lang. Processing, vol. 17, no. 2, 2009.

[77] J. Peng, L. Bo, and J. Xu, "Conditional Neural Fields," Adv. Neural Inf. Process. Syst., vol. 9, pp. 1–9, 2009.

[78] D. Yang, P. Dixon, Y.-C. Pan, T. Oonishi, M. Nakamura, and S. Furui, "Combining a Two-step Conditional Random Field Model and a Joint Source Channel Model for Machine Transliteration," in Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration (NEWS 2009), 2009, pp. 72–75.

[79] D. Yu, S. Wang, Z. Karam, and L. Deng, "Language recognition using deep-structured conditional random fields," Acoust. Speech Signal Process. (ICASSP), 2010 IEEE Int. Conf., pp. 5030–5033, 2010.

[80] G. Zweig and P. Nguyen, "A segmental CRF approach to large vocabulary continuous speech recognition," IEEE Work. Autom. Speech Recognit. Underst., pp. 152–157, 2009.

[81] G. Heigold, H. Ney, P. Lehnen, T. Gass, and R. Schlüter, "Equivalence of Generative and Log-Linear Models," IEEE Trans. Audio, Speech Lang. Process., vol. 19, no. 5, pp. 1138–1148, 2011.

[82] D. Yu, S. Wang, and L. Deng, "Sequential labeling using deep-structured conditional random fields," IEEE J. Sel. Top. Signal Process., vol. 4, no. 6, pp. 965–973, 2010.

[83] D. Yu and L. Deng, "Deep-structured hidden conditional random fields for phonetic recognition.," Interspeech, vol. @, no. September, pp. 2986–2989, 2010.

[84] Y. Bengio, N. Boulanger-Lewandowski, and R. Pascanu, "Advances in optimizing recurrent networks," in ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings, 2013, pp. 8624–8628.

[85] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in Adv. in Neural Inf. Proc. Syst. 19, 2006, pp. 153–160.

[86] L. Deng, M. Seltzer, D. Yu, A. Acero, A.-R. Mohamed, and G. Hinton, "Binary Coding of Speech Spectrograms Using a Deep Auto-encoder," Interspeech, no. September, pp. 1692–1695, 2010.

[87] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," Science (80-.)., 2006.

[88] Y. Bengio, "Learning Deep Architectures for AI," Found. Trends® Mach. Learn., vol. 2, no. 1, pp. 1–127, 2009.

[89] Y. LeCun, S. Chopra, M. A. Ranzato, and F. J. Huang, "Energy-based models in document recognition and computer vision," in Proceedings of the International Conference on Document Analysis and Recognition, ICDAR, 2007, vol. 1, pp. 337–341.

[90] J. Ngiam, Z. Chen, P. W. Koh, and A. Y. Ng, "Learning Deep Energy Models," Proc. 28th Int. Conf. Mach. Learn. (ICML 11), pp. 1105–1112, 2011.

[91] I. J. Goodfellow, M. Mirza, A. Courville, and Y. Bengio, "Multi-Prediction Deep Boltzmann Machines," in Proceedings of Neural Information Processing Systems, NIPS 2013, 2013, pp. 548–556.

[92] R. Salakhutdinov and G. Hinton, "Deep Boltzmann Machines," 2009 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work. CVPR Work. 2009, no. 2, pp. 2735–2742, 2009.

[93] G. Hinton and R. Salakhutdinov, "A better way to pretrain deep Boltzmann machines," Adv. Neural Inf. ..., no. 3, pp. 1–9, 2012.

[94] Srivastava and Salakhutdinov, "Multimodal Learning with Deep Boltzmann Machines," Adv. neural Inf. Process. Syst., vol. 15, pp. 2222–2230, 2012.

[95] R. Gens and P. Domingos, "Discriminative Learning of Sum-Product Networks.," Nips, pp. 1–9, 2012.

[96] H. Poon and P. Domingos, "Sum-Product Networks: A New Deep Architecture."

[97] D. Erhan, A. Courville, and P. Vincent, "Why Does Unsupervised Pre-training Help Deep Learning ?," J. Mach. Learn. Res., vol. 11, pp. 625–660, 2010.

[98] A.~Mohamed, D.~Yu, and L.~Deng, "Investigation of full-sequence training of deep belief networks for speech recognition," in Proc. Interspeech, 2010.

[99] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09, 2009, pp. 1–8.

[100] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Unsupervised learning of hierarchical representations with convolutional deep belief networks," Commun. ACM, vol. 54, no. 10, p. 95, 2011.

[101] H. Lee, P. T. Pham, Y. Largman, and A. Y. Ng, "Unsupervised feature learning for audio classification using convolutional deep belief networks.," Nips, vol. 9, pp. 1096–1104, 2009.

[102] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: a review and new perspectives.," IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 8, pp. 1798–1828, 2013.

[103] Y. Bengio, R. De Mori, G. Flammia, and R. Kompe, "Global Optimization of a Neural Network - Hidden Markov Model Hybrid," IEEE Trans. Neural Networks, vol. 3, no. 2, pp. 252–259, 1992.

[104] Y. Bengio et al., "A Neural Probabilistic Language Model," J. Mach. Learn. Res., vol. 3, pp. 1137–1155, 2003.

[105] E. Battenberg and D. Wessel, "Analyzing Drum Patterns using Conditional Deep Belief Networks," Int. Soc. Music Inf. Retr. Conf., pp. 37–42, 2012.

[106] Y. Bengio, P. Simard, and P. Frasconi, "Learning Long-Term Dependencies with Graident Descent is Difficult," in Saudi Med J, 2012, vol. 33, pp. 3–8.

[107] Y. Bengio, É. Thibodeau-Laufer, G. Alain, and J. Yosinski, "Deep Generative Stochastic Networks Trainable by Backprop," 2013.

[108] Y. Bengio, L. Yao, G. Alain, and P. Vincent, "Generalized Denoising Auto-Encoders as Generative Models," 2013.

[109] J. Bergstra and U. Yoshua Bengio, "Random Search for Hyper-Parameter Optimization," J. Mach. Learn. Res., vol. 13, pp. 281–305, 2012.

[110] A. Biem, S. Katagiri, E. McDermott, and B. H. Juang, "An application of discriminative feature extraction to filter-bank-based speech recognition," IEEE Trans. Speech Audio Process., vol. 9, no. 2, pp. 96–109, 2001.

[111] J. Bilmes, "Dynamic graphical models," IEEE Signal Process. Mag., vol. 27, no. 6, pp. 29–42, 2010.

[112] J. A. Bilmes and C. Bartels, "Graphical model architectures for speech recognition," IEEE Signal Process. Mag., vol. 22, no. 5, pp. 89–100, 2005.

[113] X. Glorot, A. Bordes, J. Weston, and Y. Bengio, "A Semantic Matching Energy Function for Learning with Multi-relational Data," 2013.

[114] A. Bordes and J. Weston, "Learning Structured Embeddings of Knowledge Bases," Artif. Intell., no. Bengio, pp. 301–306, 2009.

[115] L. Bottou, "From machine learning to machine reasoning: An essay," Mach. Learn., vol. 94, no. 2, pp. 133–149, 2014.

[116] Y. Bengio, "Artificial neural networks and their application to sequence recognition," no. June, 1991.

[117] E. Arısoy et al., "Deep Neural Network Language Models," NAACL-HLT 2012 Work. Will We Ever Really Replace N-gram Model. Futur. Lang. Model. HLT, pp. 20–28, 2012.

[118] L. B. Y. Le Cun and L. Bottou, "Large scale online learning," Adv. Neural Inf. Process. Syst., vol. 16, p. 217, 2004.

[119] Q. Tul et al., "Sentiment Analysis Using Deep Learning Techniques: A Review," Int. J. Adv. Comput. Sci. Appl., vol. 8, no. 6, 2017.

[120] L. Arnold, S. Rebecchi, S. Chevallier, and H. Paugam-Moisy, "An Introduction to Deep Learning," Esann, no. April, p. 12, 2011.

[121] R. Goebel and W. Wahlster, "Integrated Uncertainty in Knowledge Modelling and Decision Making," vol. 7027, 2011.

[122] A. Graves, N. Jaitly, and A.-R. Mohamed, "Hybrid Speech Recognition With Deep Bidirectional Lstm," 2013.

[123] C. Dos Santos and M. Gatti, "Deep convolutional neural networks for sentiment analysis of short texts," ... 25th Int. Conf. ..., pp. 69–78, 2014.

[124] K. Ravi and V. Ravi, "Sentiment classification of Hinglish text," 2016 3rd Int. Conf. Recent Adv. Inf. Technol. RAIT 2016, pp. 641–645, 2016.

[125] U. B. of D. Collaborators, . "The state of US health, 1990- 2010: Burden of diseases, injuries, and risk factors," Jama, vol. 310, no. 6, pp. 591–608, 2013.

[126] L. A. Pratt and D. J. Brody, "Depression in the U.S. Household Population, 2009 – 2012," NCHS Data Brief, no. 172, 2014.

[127] L. Andrade et al., "The epidemiology of major depressive episodes: Results from the International Consortium of Psychiatric Epidemiology (ICPE) Surveys," Int. J. Methods Psychiatr. Res., vol. 12, no. 1, pp. 3–21, 2003.

[128] M. Hewitt and J. H. Rowland, "Mental health service use among adult cancer survivors: Analyses of the National Health Interview Survey," J. Clin. Oncol., vol. 20, no. 23, pp. 4581–4590, 2002.

[129] M. Nadeem, M. Horn, and G. Coppersmith, "Identifying Depression on Twitter," CoRR, pp. 1–9, 2016.

[130] S. E. Chance et al., "AN EMPIRICAL STUDY OF THE PSYCHODYNAMICS OF SUICIDE: A PRELIMINARY REPORT," Depression, vol. 4, pp. 89–91, 1996.

[131] R. S. Levine, I. Goldzweig, B. Kilbourne, and P. Juarez, "Firearms, Youth Homicide, and Public Health," J. Health Care Poor Underserved, vol. 23, no. 1, pp. 7–19, 2012.

[132] R. C. W. Hall, D. E. Platt, and R. C. W. Hall, "Suicide risk assessment: A review of risk factors for suicide in 100 patients who made severe suicide attempts: Evaluation of suicide risk in a time of managed care," Psychosomatics, vol. 40, no. 1, pp. 18–27, 1999.

[133] J. J. Mann et al., "Suicide Prevention Strategies: A Systematic Review," JAMA, vol. 294, no. 16, p. 2064, 2005.

[134] R. Detels, "1.1 The scope and concerns of public health," in Oxford Textbook of Public Health, 2009.

[135] S. Radloff, "The-CES-D-Scale: A Self-report Depression Scale for Research in the General Population," Appl. Psychol. Meas., pp. 385–401, 1977.

[136] A. T. Beck, C. H. Ward, M. Mendelson, J. Mock, and J. Erbaugh, "An Inventory for Measuring Depression," Arch. Gen. Psychiatry, vol. 4, no. 6, pp. 561–571, 1961.

[137] W. W. K. Zung, C. B. Richards, and M. J. Short, "Self-Rating Depression Scale in an Outpatient Clinic: Further Validation of the SDS," Arch. Gen. Psychiatry, vol. 13, no. 6, pp. 508–515, 1965.

[138] Marina Marcus, M. T. Yasamy, M. van Ommeren, D. Chisholm, and S. Saxena, "Depression: A Global Public Health Concern," in Depression: A Global Crisis, 2012, p. 32.

[139] B. G. Tiemens et al., "Training primary-care physicians to recognize, diagnose and manage depression: Does it improve patient outcomes?," Psychol. Med., vol. 29, no. 4, pp. 833–845, 1999.

[140] M. Sinyor, A. Schaffer, and A. Levitt, "The Sequenced Treatment Alternatives to Relieve Depression (STAR*D) trial: A review," Can. J. Psychiatry, vol. 55, no. 3, pp. 126–135, 2010.

[141] K. Pajer et al., "Discovery of blood transcriptomic markers for depression in animal models and pilot validation in subjects with early-onset major depression," Transl. Psychiatry, vol. 2, 2012.

[142] P. Resnik, A. Garron, and R. Resnik, "Using Topic Modeling to Improve Prediction of Neuroticism and Depression," Proc. Conf. Empir. Methods Nat. Lang. Process., no. October, pp. 1348–1353, 2013.

[143] S. C. Guntuku, D. B. Yaden, M. L. Kern, L. H. Ungar, and J. C. Eichstaedt, "Detecting depression and mental illness on social media: an integrative review," Current Opinion in Behavioral Sciences, vol. 18. pp. 43–49, 2017.

[144] Z. Jamil, D. Inkpen, P. Buddhitha, and K. White, "Monitoring Tweets for Depression to Detect At-risk Users," in CLPsych, 2017, pp. 32–40.

[145] G. Shen et al., "Depression detection via harvesting social media: A multimodal dictionary learning solution," in IJCAI International Joint Conference on Artificial Intelligence, 2017, pp. 3838–3844.

[146] G. Coppersmith, M. Dredze, and C. Harman, "Quantifying Mental Health Signals in Twitter," Proc. Work. Comput. Linguist. Clin. Psychol. From Linguist. Signal to Clin. Real., pp. 51–60, 2014.

[147] B. Shickel, M. Heesacker, S. Benton, A. Ebadi, P. Nickerson, and P. Rashidi, "Self-Reflective Sentiment Analysis," in Proceedings of the Third Workshop on Computational Linguistics and Clinical Psychology, 2016, pp. 23–32.

[148] H. Jo, S.-M. Kim, and J. Ryu, "What we really want to find by Sentiment Analysis: The Relationship between Computational Models and Psychological State," 2017.

[149] A. Wang and D. Singh, "Detecting Depression Through Tweets," pp. 1–9, 2018.

[150] M. Zaydman, "Tweeting About Mental Health: Big Data Text Analysis of Twitter for Public Policy," no. January, 2017.

[151] R. Y. Aryya and Gangopadhyay, "Social Media Analytics for Behavioral Health," Int. J. Emerg. Ment. Heal. Hum. Resil., vol. s2, no. 3, pp. 616–617, 2015.

[152] S. S. Kale, "Tracking Mental Disorders Across Twitter Users," pp. 1–60, 2015.

[153] M. Park et al., "Depressive moods of users portrayed in twitter," Proc. ACM SIGKDDWorkshop Heal. care informatics, pp. 1–8, 2012.

[154] M. Webb, J. Burns, and P. Collin, "Providing online support for young people with mental health difficulties: Challenges and opportunities explored," Early Interv. Psychiatry, vol. 2, no. 2, pp. 108–113, 2008.

[155] "File:Colored neural network.svg - Wikipedia," 2013. [Online]. Available: https://en.wikipedia.org/wiki/File:Colored_neural_network.svg. [Accessed: 27-Jun-2018].

[156] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," 2013.

[157] T. Mikolov, G. Corrado, K. Chen, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," Proc. Int. Conf. Learn. Represent. (ICLR 2013), pp. 1–12, 2013.

[158] T. Mikolov, W. Yih, and G. Zweig, "Linguistic regularities in continuous space word representations," Proc. NAACL-HLT, no. June, pp. 746–751, 2013.

[159] L. Zhang, S. Wang, and B. Liu, "Deep Learning for Sentiment Analysis: A Survey," p. 34, 2018.

[160] D. Britz, "Understanding Convolutional Neural Networks for NLP – WildML," 7 November, 2015. [Online]. Available: http://www.wildml.com/2015/11/understanding-convolutional-neural-networks-for-nlp/. [Accessed: 27-Jun-2018].

[161] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural Language Processing (Almost) from Scratch," J. Mach. Learn. Res., vol. 12, pp. 2493–2537, 2011.

[162] Y. Kim, "Convolutional Neural Networks for Sentence Classification," 2014.

[163] R. Collobert and J. Weston, "A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning," Proc. 25th Annu. Int. Conf. Mach. Learn. (ICML 2008), 2008.

[164] wildml, "Recurrent Neural Networks Tutorial, Part 1 – Introduction to RNNs – WildML." [Online]. Available: http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/. [Accessed: 27-Jun-2018].

[165] C. Ola, "Understanding LSTM Networks -- colah's blog," 2015-08-27, 2015. [Online]. Available: http://colah.github.io/posts/2015-08-Understanding-LSTMs/. [Accessed: 27-Jun-2018].

[166] S. Hochreiter and J. Urgen Schmidhuber, "LONG SHORT-TERM MEMORY," Neural Comput., vol. 9, no. 8, pp. 1735–1780, 1997.

[167] C. Baziotis, N. Pelekis, and C. Doulkeridis, "DataStories at SemEval-2017 Task 4: Deep LSTM with Attention for Message-level and Topic-based Sentiment Analysis," in Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), 2017, pp. 747–754.

[168] "PyCharm: Python IDE for Professional Developers by JetBrains," JetBrains, 2000. [Online]. Available: https://www.jetbrains.com/pycharm/. [Accessed: 27-Jun-2018].

[169] "TensorFlow." [Online]. Available: https://www.tensorflow.org/. [Accessed: 27-Jun-2018].

[170] "NumPy — NumPy." [Online]. Available: http://www.numpy.org/. [Accessed: 27-Jun-2018].

[171] "Natural Language Toolkit — NLTK 3.3 documentation." [Online]. Available: https://www.nltk.org/. [Accessed: 27-Jun-2018].

[172] "gensim: Topic modelling for humans." [Online]. Available: https://radimrehurek.com/gensim/. [Accessed: 27-Jun-2018].

[173] "Python Data Analysis Library — pandas: Python Data Analysis Library." [Online]. Available: https://pandas.pydata.org/. [Accessed: 27-Jun-2018].

[174] "seaborn: statistical data visualization — seaborn 0.8.1 documentation." [Online]. Available: https://seaborn.pydata.org/. [Accessed: 27-Jun-2018].

[175] R. Joshi, "Accuracy, Precision, Recall & F1 Score: Interpretation of Performance Measures - Exsilio Blog," 2016-09-09, 2016. [Online]. Available: http://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/. [Accessed: 27-Jun-2018].