REPUBLIC OF TURKEY
ERCIYES UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
DEPARTMENT OF COMPUTER ENGINEERING

CLASSIFICATION OF ARABIC TEXT USING CONVOLUTIONAL NEURAL NETWORKS

Prepared By
Bilal Shakir Faraj ALKHASAWE

Supervisor
Assist. Prof. Dr. Fehim KÖYLÜ

M. Sc. Thesis

December 2018
KAYSERI

REPUBLIC OF TURKEY
ERCIYES UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
DEPARTMENT OF COMPUTER ENGINEERING

CLASSIFICATION OF ARABIC TEXT USING CONVOLUTIONAL NEURAL NETWORKS

(M. Sc. Thesis)

Prepared By
Bilal Shakir Faraj ALKHASAWE

Supervisor
Assist. Prof. Dr. Fehim KÖYLÜ

December 2018
KAYSERI

SCIENTIFIC ETHICS SUITABILITY

I proclaim that all information's in this work were acquired in accordance with academic and ethical rules. All results and material that not been at the essence of this work are also transferred and expressed by giving reference as required by these rules and behavior, I have completely referred to and referenced all material and results that are original in this work.

Bilal Shakir Faraj ALKHASAWE

SUITABILITY FOR GUIDE

The M.Sc. thesis entitled "Classification of Arabic Text Using Convolutional Neural Networks" has been prepared in accordance with Erciyes University School of Natural and Applied Sciences Thesis Preparation and Writing Guide.

Student
Bilal Shakir ALKHASAWE

Supervisor
Assist.Prof. Dr. Fehim KÖYLÜ

Chairman of the Department of Computer Engineering
Prof. Dr. Veysel ASLANTAŞ

ACCEPTANCE AND APPROVAL PAGE

This study entitled "**CLASSIFICATION OF ARABIC TEXT USING CONVOLUTIONAL NEURAL NETWORKS"** prepared by Bilal Shakir ALKHASAWE under the supervision of Assist.Prof. Dr. Fehim KÖYLÜ was accepted by the jury as M.Sc. Thesis in computer engineering.

12/12/2018

JURY:

Supervisor : Assist.Prof. Dr. Fehim KÖYLÜ

Juror : Assist. Prof. Dr. Ö. Ufuk NALBANTOĞLU

Juror : Assist. Prof. Dr. Hasan BADEM

APPROVAL

That the acceptance of this thesis has been approved by the decision of the Institute's Board of Directors with the 02/01/2019 date and 2019/01-06 numbered decision.

02/01/2019

Mehmet Akkurt

Prof. Dr. Mehmet AKKURT
Director of the Institute

ACKNOWLEDGEMENTS

First and foremost, thanks to Allah for giving me the power and help to accomplish this research. Without the grace of Allah, I was not able to accomplish my thesis study.

Many thanks and sincere gratefulness go to my supervisor Assist. Prof. Dr. Fehim Köylu, for his help, guidance, and continuous follow-up in this research.

I would also like to express appreciation to my beloved family who spent sleepless nights with me and was always my support in the moments when there was no one to answer my queries.

At the end I would like to thank all my friends who supported me in writing and incented me to strive towards my goal.

Bilal Shakir Faraj ALKHASAWE

KAYSERI, December 2018

CLASSIFICATION OF ARABIC TEXT USING CONVOLUTIONAL NEURAL NETWORKS

Bilal Shakir ALKHASAWE

Erciyes University, Graduate School of Natural and Applied Sciences
M.Sc. Thesis, December 2018
Thesis Supervisor: Assit. Prof. Dr. Fehim KÖYLÜ

ABSTRACT

The increasing amount of Internet users and the consequent increase of online user reviews, expressing their opinions, has resulted in huge and raw data. This data includes important but unprocessed feedback about products and services. This leads to a growing interest on dealing with the analysis of this type of text data. This area of research is typically called sentiment analysis or opinion mining. In the literature different classifiers are proposed for text sentiment analysis. First raw text data is preprocessed and after that a learning algorithm is used. These algorithms are applied for English language very well. Unfortunately, Arabic is more morphological and wide used language which has different accents used in the world. Support Vector Machines, Decision Trees, and Naïve Bayes algorithms are used for classifying Arabic text in different studies. In this paper, convolutional neural networks (CNN) models are used to classify the Arabic text by using deep learning methods. LABR book review dataset is used for benchmark tests. The results are compared with Support Vector Machines and Naïve Bayes algorithms. In experiments, it is shown that some results of CNN model are better than SVM and NB classification results.

Keywords: Text classification, Arabic text mining, sentiment analysis, convolutional neural networks, deep learning.

KONVOLUTIONAL NEURAL AĞLARI KULLANILAN ARABIC METİNİN SINIFLANDIRMASI

Bilal Shakir ALKHASAWE

Erciyes Üniversitesi, Fen Bilimleri Enstitüsü
Yüksek Lisans Tezi, Aralık 2018
Tez Danışmanı: Dr. Öğr. Üyesi Fehim KÖYLÜ

ÖZET

İnternet kullanıcılarının sayısı ve çevrimiçi kullanıcı yorumları ve görüşlerinin artması, büyük ve ham verilerin oluşmasıyla sonuçlandı. Bu veriler ürün ve hizmetler hakkında önemli ama işlenmemiş geri bildirimleri içerir. Bu tür metin verilerinin analizi ilgi çekmeye yol açmaktadır. Bu araştırma alanı genellikle duygu analizi veya görüş madenciliği olarak adlandırılmaktadır. Literatürde, metin duyarlılığı analizi için farklı sınıflandırıcılar önerilmiştir. İlk başta ham metin verileri önceden işlenir ve bundan sonra bir öğrenme algoritması kullanılır. Bu algoritmalar İngilizce dili için çok iyi uygulanır. Ne yazık ki, Arapça dünyada kullanılan farklı aksanları olan ve daha morfolojik ve yaygın olarak kullanılan bir dildir. Farklı çalışmalarda Arapça metinleri sınıflandırmak için Destek Vektör Makineleri, Karar Ağaçları ve Naïve Bayes algoritmaları kullanılır. Bu çalışmada, derin öğrenme yöntemleri kullanılarak Arapça metni sınıflandırmak için evrişimsel sinir ağları (CNN) modelleri kullanılmıştır. Karşılaştırılmalı testler için LABR kitap inceleme veri seti kullanılmıştır. Sonuçlar Destek Vektör Makineleri ve Naïve Bayes algoritmaları ile karşılaştırılmıştır. Yapılan deneylerde, CNN modelinin bazı sonuçlarının SVM ve NB sınıflandırma sonuçlarından daha iyi olduğu ispatlanmıştır.

Anahtar kelimeler: Metin sınıflandırma, Arapça metin madenciliği, duyarlılık analizi, evrişimsel sinir ağları, derin öğrenme.

TABLE OF CONTENTS

CLASSIFICATION OF ARABIC TEXT USING CONVOLUTIONAL NEURAL NETWORKS

CHAPTER: FOUR

EXPERIMENTAL RESULTS

CHAPTER: FIVE

CONCLUSION AND FUTURE WORK

LIST OF ABBREVIATIONS

ANLP	**:** Arabic Natural Language Processing
C4.5	**:** Decision Tree Algorithm
CNN	**:** Convolutional Neural Network
CSV	**:** Comma-Separated Values
DL	**:** Deep Learning
KNN	**:** K-Nearest Neighbours Algorithm
LABR	**:** Large Arabic Book Review Corpus
LDC	**:** Linguistic Data Consortium
LSTM	**:** Long Short-Term Memory
ML	**:** Machine Learning
NB	**:** Naive Bayesian Algorithm
NG	**:** Negative Reviews
NLP	**:** Natural Language Processing
NLTK	**:** Natural Language Toolkit
NN	**:** Neural Network
POS	**:** Positive Reviews
SVD	**:** Singular-Value Decomposition
SVM	**:** Support Vector Machine Algorithm
T.F	**:** TensorFlow
TF-IDF	**:** Term Frequency and Inverse Document Frequency

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE

GENERAL INFORMATION

1.1. Introduction

Text classification is commonly used in data mining to obtain valuable data from a large data source. There has been as a dynamic growth in information in recent years across various fields on the Internet, which has made classifying text among the most crucial areas of study related to data mining[1]. Text classification can be defined as the labeling documents within thematic categories. or the process of classifying or structuring documents [2]. into a predefined set of categories according to a group structure that is known in advance. However, to make applications for text classification, we must go through different study areas such as AI, deep learning, and machine learning. Text classification gives us an easy way to figure out meaning from the raw Arabic text.

There are various methodologies that oversee text classification. Some of the common methods include: Bayesian, models using a decision tree, *k*-nearest neighbors, and algorithms based on statistics and distance [3]. Many applications utilize text classification to some degree, including: filtering emails based on topic or spam, routing messages, monitoring and classifying items on news sites, indexing scientific papers, and sifting through old archives of paper documents.

The vast majority of text classification methods were designed to work on English documents. As such, they are inapplicable to texts written in Arabic. Because of the complexity of the Arabic language, designing similar systems to classify Arabic texts remains a challenge. The language is made up of 28 letters and is written right-to-left. Its complex nature is due to its unique morphology, with words produced from a root. Most words are based in a three-letter root. However, some (although much fewer) come from roots of four, five, or 6 letters.

Text classification uses either a rule-based approach or machine learning. In the former, a knowledge engineer must build a system that will automatically classify the text. The latter is built from an inductive computer process that involves feeding the software with training documents that are already classified into the relevant categories [4].

In this thesis, we train different models to classify the Arabic text. We train the CNN algorithm with different structure models until finding the optimal one, which is then used in our framework. We train our model under TensorFlow packing that allows to us to use many various libraries. Using TensorFlow packing is rare with Arabic text. This is especially true when considering the CNN model with T.F. As such, its precedent is set in this study (up to our knowledge). It then becomes possible for additional studies to be carried out in this study arena using our research as a base.

When using the CNN model packed with TensorFlow, it was found as a flexible area for development, and more accurate results were if both were combined. This research can greatly assist the future of the Arabic classification text arena. When deep learning methods, especially the CNN model, was compared with the other traditional methods (Decision tree, Naive Bayesian, etc.) to perform tasks such as those found in this research, it was discovered that deep learning methods are suitable for application on Arabic text.

Therefore, the techniques that are used for this thesis involve deep learning methods. Deep learning is a unique branch of machine learning developed with the hope of improving artificial intelligence systems. It makes sense of information found in text, images, sound, and video through various levels of representation and abstraction [5]. Deep learning has also shown itself very adept at text classification, with high levels of results through common academic benchmark issues. Other names for deep learning include hierarchical and structured learning. It is among other machine learning tools that are meant for observing and learning data representations, opposite of the algorithms that focus on tasks alone. It falls under supervised, unsupervised, and semi-supervised learning methods.

Usually, deep learning algorithms are composed by multiple computational layers, where each layer process inputs data using a simple, non-linear function, transforming the input into a more abstract representation of the data. This is what allows deep learning algorithms to provide high quality results even when raw data is used as input. They were commonly used for processing images but have recently shown success for data mining and text classification [6].

Text classification is also one of the important and typical tasks in supervised machine learning (ML). This includes assigning categories to documents, which can be a web page, library book, media articles, image gallery, and more. The term text analytics portrays an arrangement of linguistic, statistical, and machine learning techniques, that model and structure the data substance of literary hotspots for business knowledge, exploratory data analysis, research, or investigation. Moreover, the text classification model can be divided into unsupervised, semi-supervised, or supervised. Supervised deep learning has earned great success in recent years for most text classification models [7].

The analyzed text is a book review used from (LABR: A Large-Scale Arabic Book Reviews Dataset). It is a set of over 63,000 book reviews, each with a rating of 1 to 5 stars [8]. Therefore, to classify these book reviews text mining algorithms were used, with five trained CNN models, to find the most accurate one for the Arabic text.

The complexity of Arabic morphology and the lack of complete vowelization of standard Arabic texts making working on it a challenge. There is also an abundance of Arabic dialects and an extremely complex inflectional dialect, which makes the standard investigation an exceptionally mind-boggling undertaking.

The Arabic language is difficult to analyze through any means. This is because of the morphology of the language, as mentioned above, that carries a great deal of grammatical and lexical information. The Arabic syntax related to the structure of sentences is also quite different from other languages, and vastly different from English. Words in Arabic convey great deals of information, a unique trait that is not found in other languages [9].

Neural networks have recently attracted much more research focus, owed to the advancements in methodology and easier access to powerful computer tools [10]. The prevalence of what are known as deep neural networks comes from their capacity to heartily distinguish pictures. Advances in the most recent decade have been exceptionally great in their methods which can be applied to text classification.

Through NLP, computers are able to gather meaning from, understand, and analyze human languages to great effectiveness. Through NLP, many tasks are possible from the language gathering, including: translation, the development of relationships, analyzing text sentiment, organizing text into useful categories, summarizing the texts, and more.

NLP is a difficult problem in the computer science field. This is because human languages are often spoken in a broken manner, without proper grammar or plain speech. Understanding these languages requires more than gathering the meanings of the words. It requires knowing the concepts being spoken about and how the words are linked in ways that produce meanings. Although languages are among the easiest skills for humans to acquire, they remain difficult for computers to grasp because of the ambiguity is common spoken and written language. This is especially true for the Arabic language, which was used as a platform for thesis study.

TensorFlow is a computational framework that allows for the building of machine learning models and provides a great deal of toolkits for the building of models at various abstraction levels. It is possible to use lower-level APIs to build models by defining a series of mathematical operations. Alternatively, higher-level APIs (like T.F estimator) can be used to specify predefined architectures, such as linear regressors or neural networks [11]. TensorFlow a machine learning library that was created by Google’s Brain Team and was made open-source in November 2015. It was originally produced for machine learning and deep neural network research [12].

The TensorFlow Keras library, which includes the CNN models, was used in this research. These models are compared with the classical algorithms such as (SVM, NV), and their result was found to be on the same dataset which we used in (LABR A Large Scale Arabic Book Reviews Dataset) [8]. Also, matrices like (Bi- gram, N-gram) were used for the preprocessing steps. Our aim from this study is to find the best method or algorithm for doing text classification on Arabic text (Book reviews).

Keras is a High-level neural network. The API and Modular-Building models are simply stacking layers and connecting computational graphs. It runs on top of either

TensorFlow, Theano, or CNTK. We used Keras because it useful for fast prototyping, ignoring the details of implementing backprop or the writing optimization procedure. It also supports Convolution, the Recurrent layer, and the combination of both. It can also be run seamlessly on both CPU and GPU. Moreover, almost any architecture can be designed using this framework. Lastly, of course, it is an open source code with large community support [13].

1.2. Problem Definition (Aim of thesis research)

In this thesis study, we discuss how to evaluate deep learning methods on Arabic text book reviews in our case, by using CNN models' convolutional neural networks. The idea consists in finding best and the most accurate method to classify Arabic text. In more detail, we try to use information about the emotion dimensions to help text classification models to improve their prediction performance on Arabic text. In order to evaluate the tasks performance, datasets from different contexts and different topics were used.

Moreover, the challenges of using Arabic content as a platform began from the first step which is preprocessing and how-to prepper the Arabic corpus to classify it.

The problem in our thesis study includes many issues. firstly, is how to cover the huge amount of Arabic language with a weak reference in this field. Secondly, it's how to convert the codes to apply it on the Arabic text.

At the end classifying these reviews depending on their sentiment positive or negative, that give us a clear idea for them opinions.

1.3. Thesis Organization

The thesis is organized as follows:

• **Chapter 2: Related work:** gives an overview of the text classification on Arabic text and the research concluded in this field and a literature review related to our proposed framework.

• **Chapter 3: Proposed Technique:** Presents the proposed technique and describe our proposed models and how to evaluate text classification on Arabic text (Book review).

• **Chapter 4: Experimental Results:** Presents their experimental results we achieved through comparing our developed proposed approach to the other models we train it also in our experimental.

• **Chapter 5: Conclusion and Future Work:** It concludes the thesis briefly and mentions the possible direction for future work.

CHAPTER TWO

BACKGROUND AND RELATED WORK

This chapter explains the basic information relating to our work. The first section shows. the definition of text classification as well as the different techniques that have been used for English text classification as well as other languages. The main concepts about the Arabic Language are illustrated in the second part of this chapter. The related works that have been achieved in the previous works on Arabic text classification are mentioned in the last section.

2.1. Background for Classification Arabic Text

2.1.1. The Layers used within Convolutional Neural Networks

This section presents all the layers composing the model's architectures, each of them with its own section, and where is made a description of the respective usefulness purpose and functioning.

2.1.1.1. Embedding Layers

An embedding layer aims to convert word indices into dense representations. In this case, three different options are available: the first is to learn the embedding weights from scratch, i.e. The word embeddings are initialized with random values and will be improved over the training process. The second is to use pre-trained embeddings, keeping the embedding weights static. The third and last option, is to use pre-trained embeddings, but instead of keeping static the weights they will be improved during the training process[14]. We tried many dim sizes and inputs for our embedding layer, before deciding to use input with 40 words and 100 of dimensional size.

2.1.1.2. Dropout Layers

The dropout layer has the responsibility of helping models to prevent overfitting. As described throughout the document, deep neural networks are composed by multiple non-linear hidden layers, making these models very expressive. The existence of these types of layers allow the models to learn very complicated relationships between inputs and outputs. Considering the limited training data, many of the relationships will be the result of sampling noisemaker. some relationships will exist in the training set but not in the test set. When this occurs, the model does not generalize well, causing bad results in terms of the prediction performance. To avoid the overfitting problem, many methods have been developed, one of them being the dropout method. Essentially, the dropout method forces the neural network to learn multiple independent representations of the same data (dropping out neurons during the training phase), with the aim to reduce the dependence on the training set. A dropout layer will thus set to zero a given portion of its inputs [14].

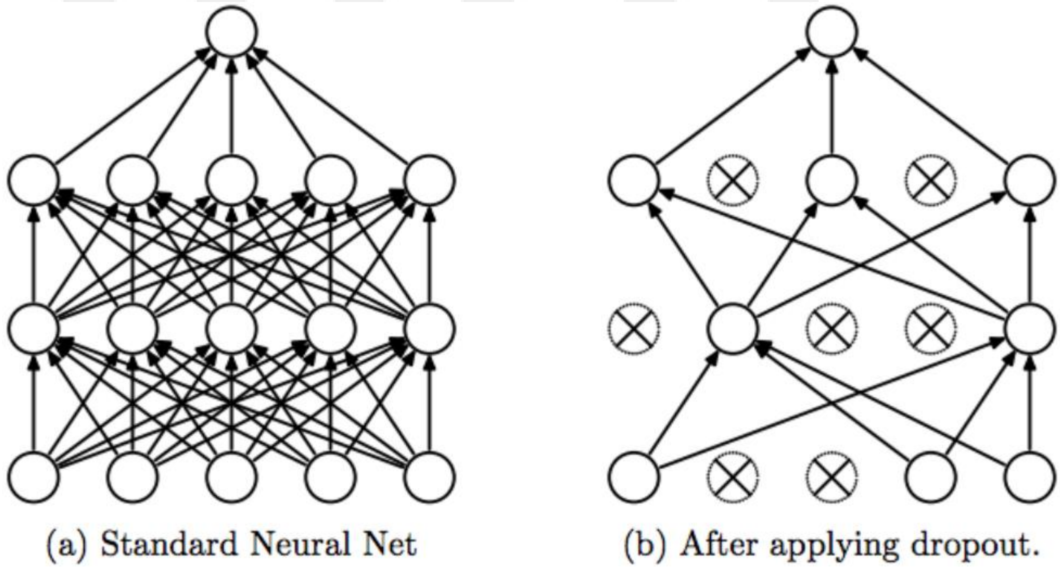

Figure 2.1. Dropout Layers [15].

2.1.1.3. Dense Layers

A dense layer represents a regular fully connected layer. The idea of this layer is to connect every neuron in the network to every neuron in adjacent layers.

2.1.1.4. Convolutional Layers

A convolutional layer is the most relevant in the case of CNN models. These layers are composed by a set of learnable filters with a small receptive field. During the forward process, each filter is convolved across the full depth of the input volume. This process is accomplished computing the dot product between the entries of the filter and the input, to produce the activation map of each filter. Finally, the key point is that the network learns filters which are activated when they see some specific type of feature at some special position in the input. The output of a convolutional layer is represented by a stack of activation maps produced by each filter. Each stack position be an output of a neuron that looks at a small region in the input[16].

2.1.1.5. Pooling Layers and Flatten Layers

A pooling layer is commonly used between successive convolutional layers in a CNN model. Their goal consists in progressively reducing the spatial size of the representation, which will cause also a reduction of network parameters and computations. Due to this fact, these layers are also an important aid to control the overfitting problem. The most traditional function to execute this operation is max pooling. The idea is to split input representation into a set of non-overlapping regions and for each of these regions select the max value. Thus, at the end, each element of the new representation is the max value of a region in the original input representation. The intuition behind this is that once a feature has been detected, its exact location is not very important comparing to the importance of its location relative to other features as showing below in figure 3.7. A flattening layer, as the name indicates, flattens the input. More properly, these layers flatten all the dimensions of their inputs into one dimension, as showing below in figure 3.8 [17].

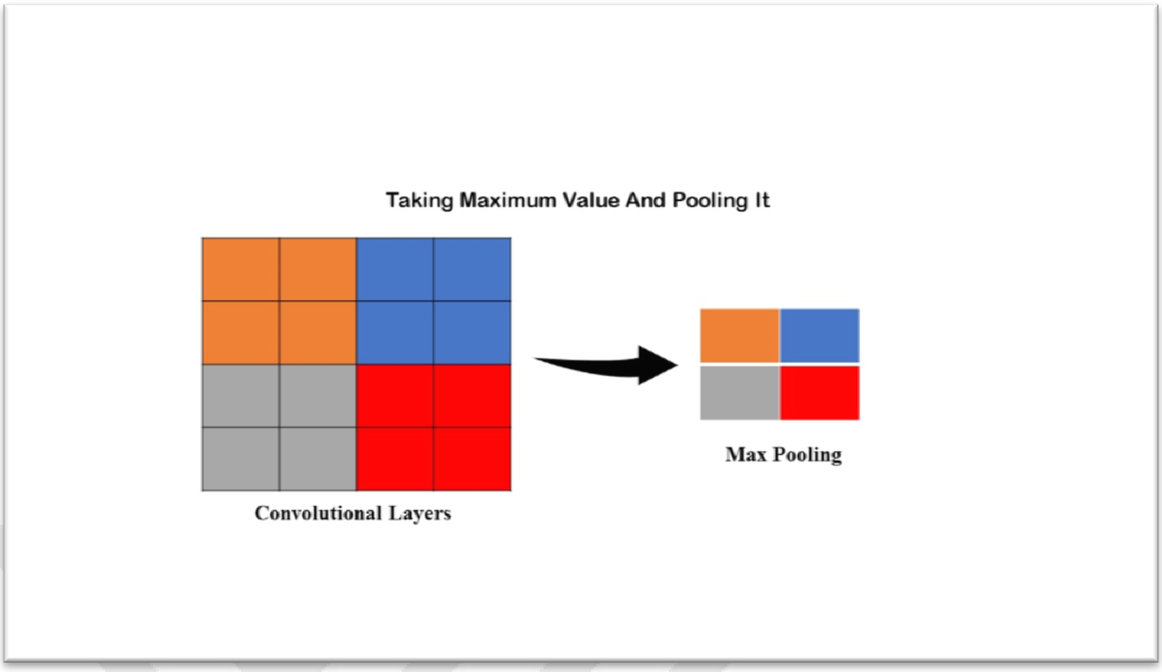

Figure 2.2. Max Pooling

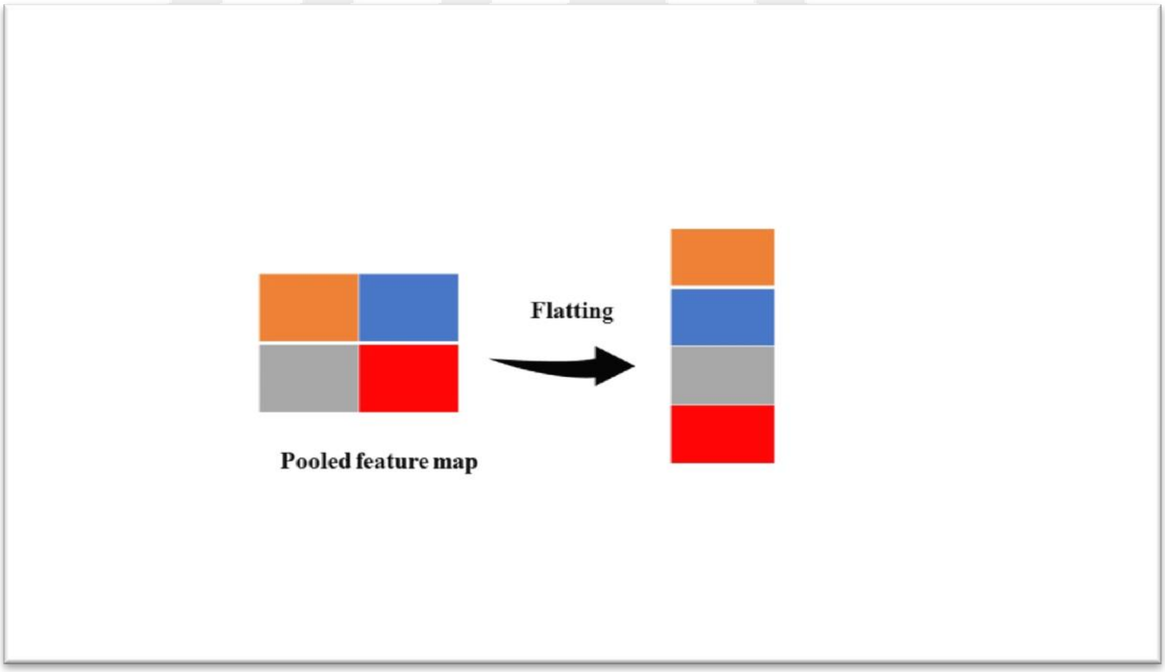

Figure 2.3. Flatting

2.1.1.6. Activation Function

An activation layer just applies an activation function to an output. An activation functions in neural networks, besides restricting outputs to a certain range, breaks the neural network linearly, allowing it to learn more complex functions than linear regression. A neuron without an activation function is equivalent to a neuron with a linear activation function, like f(x) = x. When these functions do not add any non-linearity, the entire network is equivalent to a single linear neuron. Therefore, it makes no sense building a multi-layer network with linear activation functions. Moreover, given that a single linear neuron is not capable of dealing with non-separable data, no matter how deep a multi-layer network is it can never solve any nonlinear problem. Taking this into account, the role of activation functions in a neural network is to produce a non-linear decision boundary, via a non-linear combination of the weight inputs[18].

2.2. NLP with Arabic Language

Natural Language Processing is a branch of research related to the exploration of the possibilities of computers for the comprehension and manipulation of language in either text or speech and how they can carry out related duties. NLP researchers gather information regarding human use and understanding of languages to produce useful tools and methods that assist computers in their language comprehension and the carrying out of required tasks [34].

There are many challenges for computers learning the Arabic language. The most prominent issue is the understanding the characters and the diacritical components of the language. As such, in order to accurately understand the rules of the Arabic language, it is necessary to have a deep knowledge on all of the language’s details [35].

Many difficulties are related to Arabic Natural Language Processing (ANLP), including diglossia and regional dialects. Because of this, analyzing the sentiment of Arabic text carries these ANLP difficulties, mainly because of the way syntax is combined, the morphology, and lexical features of Modern Standard Arabic and classical Arabic [36]. Also, there is no standardization of Arabic other than from media sources and in the academic world, which is a serious challenge for ANLP research, especially as it relates to analyzing the language and dialects used on social media [37], [38]. Sentiment analysis systems are usually exclusive to languages with a single form, such as English. Meanwhile, observing sentiment on an international scale necessitates analyzing text across various language forms related to the dialects. In addition, the challenges of dialectical diversity and the language’s richness hamper the job of text analysis [39]. In the literature, studies acknowledged nine principal reasons for Arabic language being particularly problematic for sentiment analyses such as complicated morphology and use of Arabic dialects that feature improper or different variants in spelling. Those apprehensions can be summarized as either a lack of language resources or problems associated with regional dialects [40].

Arabic Sentiment Analysis researchers have built the Sentiment Analysis corpus from diverse electronic online data. As far as Arabic Sentiment Analysis research is concerned, in 2013 a team led by Abdulla [41] wrote about the publicly available Arabic Twitter datasets being limited in size and coverage of Arabic dialects, with a focus on the Jordanian Twitter datasets [41] . Research on SA of Arabic emerged in 2008 with the publication of research from Abbasi [20], which presented a supervised approach to sentiment analysis of both English and Arabic in web forums. A survey on SA of Arabic was first presented in [42], a recent survey on SA of Arabic can be found in Al-Twairesh's 2014 research [43] that reviewed the corpora and datasets publicly available for Arabic SA and highlighted the construction methods of each. Also, for a comprehensive review of the resources available for Arabic SA including corpora, the research points to alOwisheq's 2016 study [44]. One of the earliest corpora for Arabic SA is the Opinion Corpus for Arabic (OCA) by Oakleaf in 2011[45] which has served as a benchmark for several Arabic SA studies. It was constructed manually through the extraction of 500 Arabic movie reviews from the Web. It contains 250 positive reviews and 250 negative reviews and was used by numerous research papers. Another corpus in the domain of reviews is the Large Arabic Book Review Corpus (LABR) by Aly. in 2013 [8]. It contains a dataset of 63,257 book reviews from the book readers' social network www.goodreads.com. The reviews were already rated from 1-5. They considered as positive reviews those with ratings 4 or 5, and negative reviews those with ratings 1 or 2. Reviews with rating 3 are considered neutral and not included in the polarity classification. AWATIF is a multi-genre corpus for MSA SA done by Abdul-Mageed in 2012 [27]. It consists of sentences from the Penn Arabic Tree Bank (PATB), Wikipedia Talk pages and Web forums. However, it hasn't been released to the public. In the expanse of Twitter datasets, in 2014, research by Refaee [38] constructed and released a corpus of Arabic tweets annotated for subjectivity and sentiment analysis available on the LREC repository of shared resources. It consists of 6,894 tweets: 833

positive, 1,848 negative, 3,685 neutral and 528 mixed. It was annotated for morphological features, simple syntactic features, stylistic features and semantic features. The authors in Nabil's 2015 study [46] presented the Arabic Sentiment Tweets Dataset (ASTD) which is a dataset of 10,000 Egyptian tweets. It consists of 799 positive, 1,684 negative, 832 mixed and 6,691 neutral tweets. The authors also conducted a set of benchmark experiments of four-way sentiment classification and two stage classifications. MIKA is a SA corpus of MSA and the Egyptian dialect by Ibrahim in 2015 [47]. The corpus is extracted from tweets, comments on hotel reservations, TV programs, and product reviews annotated at the sentence level. It consists of 2,154 positive, 1,648 negative and 198 neutral texts. The text was also annotated with polarity strength (-10 to 10) following a set of rules that handle negation, contextual intensifiers and mixed polarity cases. In [48], the authors constructed a dataset of 33,116 reviews about hotels, restaurants, movies, and products. The sentiment of the reviews was determined through the ratings, so no manual annotation was required. A small lexicon was extracted that contained around 2,000 words classified as per review type. Several experiments were conducted to determine the best features and classifiers to be used with the dataset. This work serves as a good benchmark for the reviews genre. Most of the work mentioned above does not give any details on the annotation process and what annotation guidelines were used. In the field of Arabic SA, Abdul-Mageed in 2011 [49] outlined the annotation guidelines that were used in annotating Arabic text from the newswire genre. The annotation guidelines were linguistically motivated and provided original insights that we adopted in our work when preparing the guidelines. Moreover, in 2016, Mohammad [50] presented a practical guide to annotation of sentiment while outlining the challenges of annotating sentiment; which also helped us in shaping the guidelines in our work. LSTM neural networks are considered as state-of-the-art approaches offering very high accuracy results in several Natural Language Processing tasks such as Bi-directional LSTM-CRF [51]. For Part of Speech Tagging and Tree-LSTMs for sentiment analysis, the research of Tai in 2015 [52] is profound. Also, simpler versions of LSTMs, referred to as Gated Recurrent Units (GRUs), were carried out by Chung's team in 2014 [53]. This was used in a 2015 study by a team led by Kumar [54] as key parts of larger systems like state-of-the-art Dynamic Memory Networks, developing more complicated tasks such as questions-answering systems.

2.2.1. Arabic Language

This section describes the various methodologies that were applied by researchers in the hopes of extracting sentiments and opinion from Arabic text. Their work shows why the task of Arabic text classification is so different from deducing the sentiment of English text. The popularity of sentiment analysis has largely revolved around English language works. For example, the 2010 work of Montoyo [14] covered sentiment analysis from social media posts, film and product reviews, blog and news articles, and Web forum posts. The 2011 study of Abdul-Mageed [19] represented a breakthrough in analyzing the sentiment of Arabic Internet content. The work of this thesis also followed in the footsteps of the 2008 study by Abassi [20] in our taxonomy. The study outlined the three tasks related to Arabic sentiment analysis, and there are identical to the tasks performed on English text: class labeling, granular annotation, and the identifying of source and target. When analyzing sentiment, works such as the 2012 study of Shoukry [21] used only two classes: positive and negative. Others used only one step for the classification, like Elhawary in 2010 [22], or performed subjectivity classification alone, like Abdul-Mageed in 2010 [22]. Still, most works employed a two-step approach to classifying text, subjectivity and the classifying of polarity, which can be found in the subsequent work of Abdul-Mageed in 2011 [19] and 2012 [23]. Categorization done at the sentence level has grown into popularity, sparked by Shoukry and his research from 2012 [21]. This growth is probably due to two factors. The first is the emergence of social media text, which is usually shorter than normal speech for either convenience or by requirement (such as a Tweet). The second reason is how analyzing the sentiment of sentences is seen as a subtask of performing the sentiment analysis to an entire document. As such, improving this subtask will only improve the main task. Also, the task related to identifying the source and target of a text also relates to finding the source of the sentiment. An example would be the phrase, “I hate iPhones.” Here, the speaker is the source of the sentiment. By contrast, the phrase, “My brother hates them,” has a different source who is being quoted. In both phrases, the “iPhone” is the target. As per the findings related to this work, there was no study related to target identification under sentiment analysis for Arabic. It was, however, carried out in English in the work of Jiang in 2011 [24]. In order to reproduce such works that were carried out in English, there have been an assortment of feature types used to classify Arabic sentiment. In research by Shoukry in 2012, [21] Amira and Rafea used syntactic features, where unigrams and bigrams occurred more than 5 times in the dataset. Also, Abdul-Mageed's 2010 research [22] used tokens with POS tags as syntactic features. The studies also featured the inclusion of semantic features, including detecting adjectives of positive or negative sentiment. Bautin's research in 2008 [23] used 33 only semantic features, including the frequency of such adjectives. In 2011, Abbasi [19] employed syntactic and stylistic features in the analysis. The syntactic features included root words, punctuation marks, and n-grams. The stylistic features related to digit and function word frequency as well as whether or not URLs were present. On a general scale, the three tools used to classify sentiment in Arabic are the same as those used in English. They include approaches using: machine learning, score, and link analysis. Much the same as how English is classified, machine learning methods are the most commonly used tools for Arabic, including the widespread use of both SVM and NB. Amira in 2012 [21] and Mageed in 2010 [22] both used SVM and NB, while the latter used Instance-Based Learning (IBL) as well. Also, Abdul-Mageed in 2012 [25] and 2011 [19] used SVM alone. The method of link analysis has yet to be used for detecting Arabic sentiment. Still, this technique is quite popular on social media platforms such as Twitter due to the available data related to the source and network. In this method, such as the work of Wang in 2011 [26], researchers begin by constructing a relational graph and labelling a small number of nodes in the graph. Then, they propagate through the graph to label the remaining nodes, using algorithms such as relaxation labeling (RL). In Arabic sentiment analysis, various input domains have been used, including the social media website Twitter by Amira in 2012 [21], Abdul-Mageed in 2012 [25]. The Penn Arabic Treebank was used by Abdul Mageed 2011 [19], 2010[22], and 2012 [27]. Blogs and news were analyzed by Bautin in 2008 [23]. Product reviews were investigated by Elhawary in 2010 [28]. Forum posts were studied by Abbasi in 2008 [20]. Interestingly, Facebook, the world's leading social media platform, has received no attention from the Arabic sentiment analysis community.

2.2.2. Grammar Structural Speaking for Arabic language

Arabic language is one of the most difficult languages in the world because it relies on a set of rules and basic grammar. When Arabic is spoken, the sentence must be based on a structure such as the same language (names, verbs, adjectives, etc.) to be correct and clear when it is articulated. The structures can be:

i. Noun + Verb + Character

ii. Verb + Actor + Object

iii. Verb + Actor + Time-Loc

iv. Type of Plural

- Masculine
- Feminine
- Cracking

i. Noun + verb + character:

The first rule shows that speech in the Arabic language is built around the noun and verb and character. As seen in Table 2.1, the sentence components are linked to each other if. If any of these elements are removed, the sentence become unclear and not understandable.

Table 2.1 Structural Analysis of ANLP

Ismail	Plays	With a cat
Noun (N)	Verb (V)	Character (C)
N	V	C
N + V + C		
Ismail plays with a cat		

Table 2.1 portrays the rules and the form of noun, verb, and character to display the adoption of the sentence and the consistency between them, where if one of them were deleted the sentence is incomprehensible.

ii. Verb + Actor + Object:

The onset of the process of analysis, = c =, is separated into two elements: the actual compound of the verb and actor which is shown as (MF), and the nominal compound (object), shown by the B (M S). As such, dissect (MF) to do (P) and actor (m), and then, (P) to do verb (effect) and time (g). Lastly, nominal compound (PG) is dissected to (a) Definition Tool (define). So, the result of the analysis is the status of bilateral sports, synchronized to form the sentence of: the student opened the door.

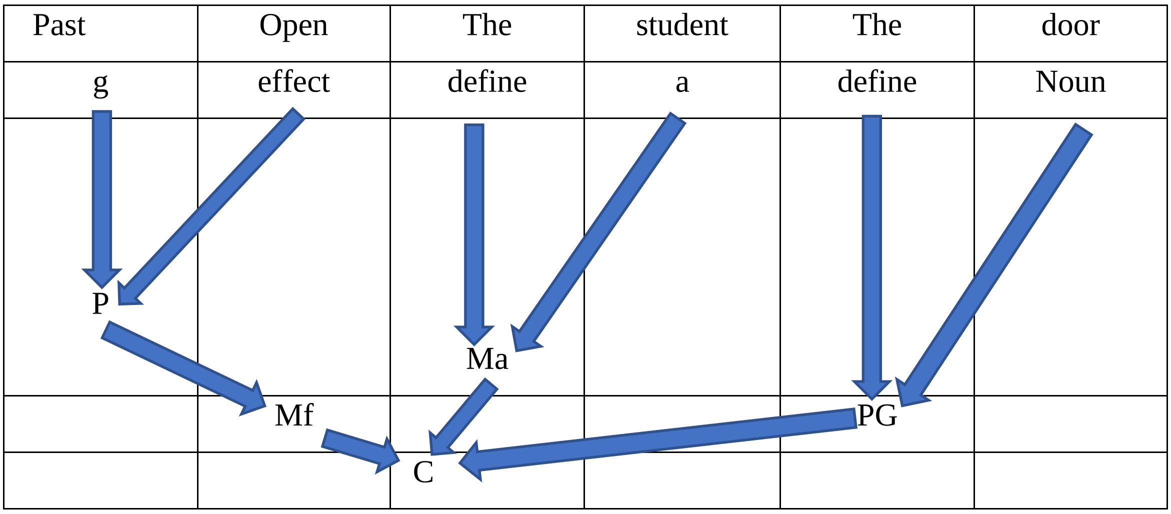

Figure 2.1: Structural Analysis of ANLP (Chomsky, 2005)

Figure 2.1 portrays the reconstruction rules: a set of rules that branches the sentence "as an initial" symbol. Chomsky tried to explain the analytical levels of inters beginning with the analytical level, which divides (c) to the following equation:

iii. Verb + Actor + Time-Loc

Example: my son went to school in the morning

This is one element of Arabic grammar that when speaking depends on time and place, as in the example above where see the presence of a time (morning) and place (School). When you delete one of these words be sentence is not understandable.

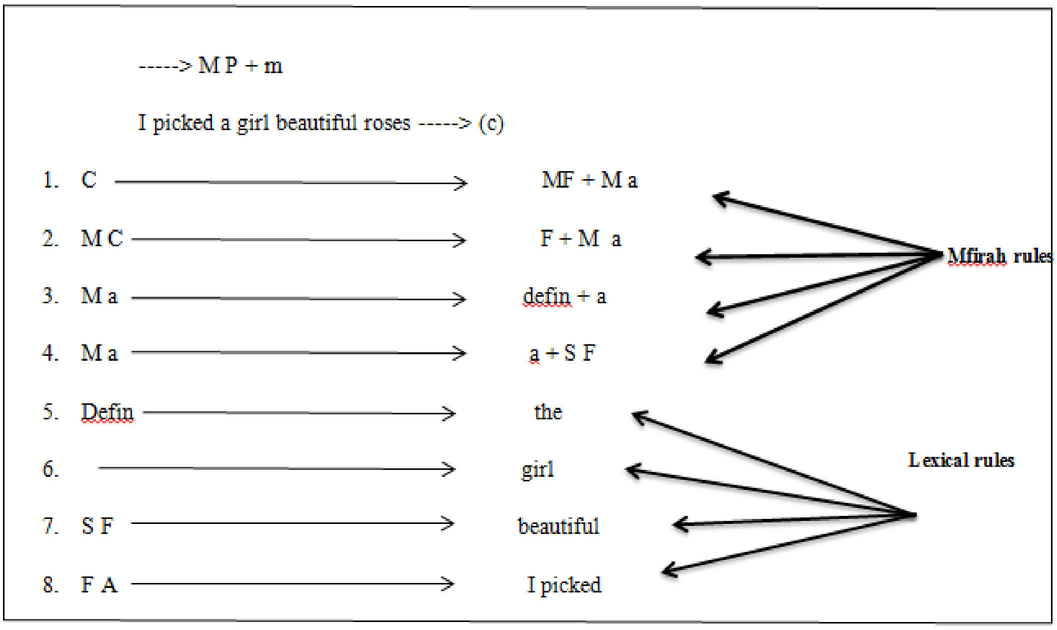

Figure 2.4 The Equation of Chomsky (Chomsky, 2005)

Figure 2.2 is the Chomsky's equation which provides two rules. "Mfirah" are the branching levels of linguistics. However, the lexical rules provide the levels of language vocabulary, and after they end, an analysis through application of the Mfirah rules, as laid out through applying the lexical rules, will generate the language chains. Its aim is the presentation of the sentence's generative evolution based on the above rules. For instance, rule No. 3, the natural result of a series of language, is altered to Definition + Name (defin + a). Chomsky also discovered discharges organized in a tree that can make the procedure simpler. This is called a Dingle composition, which produces the sentence's hidden structure, as shown in Figure 2.3.

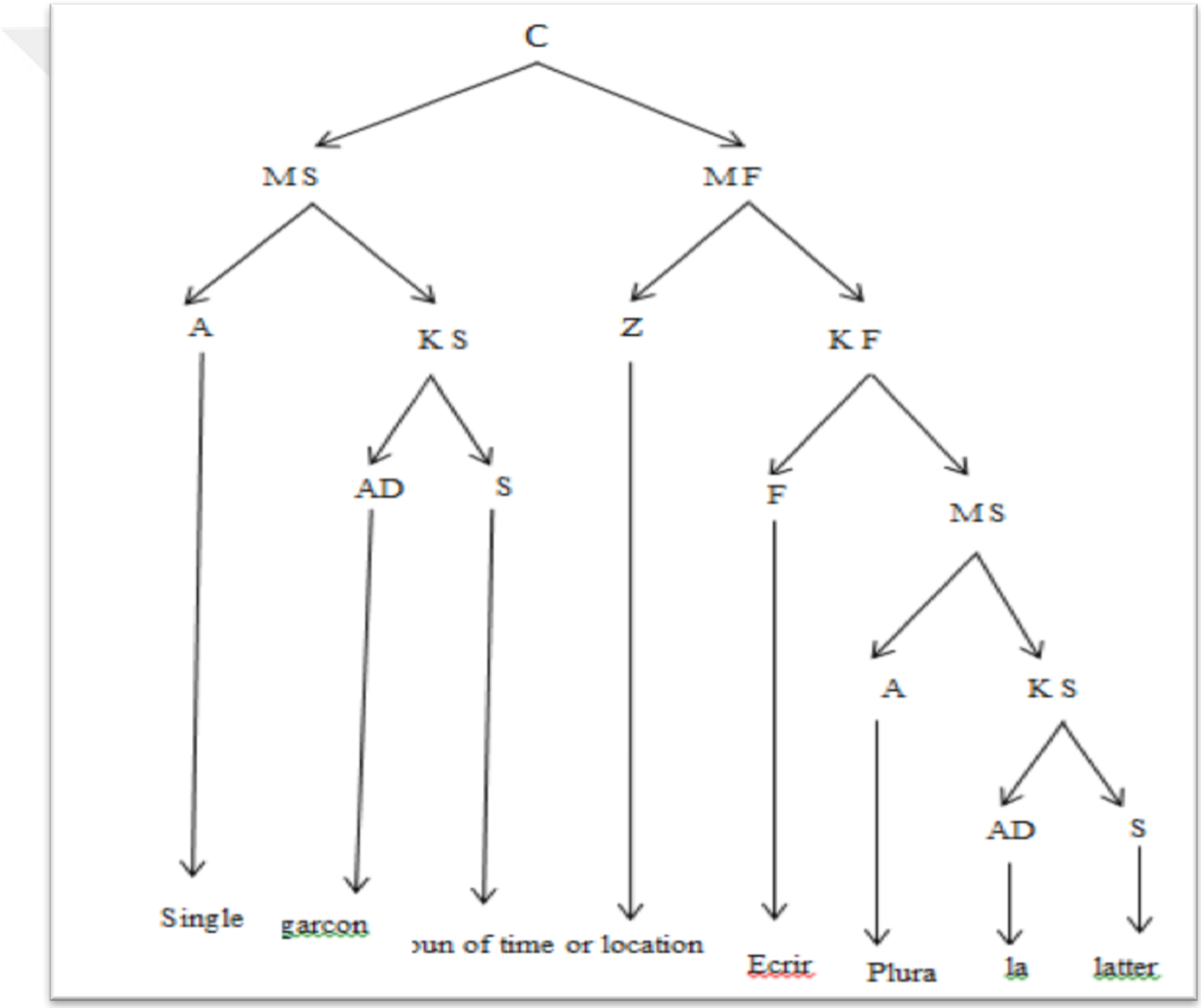

Figure 2.5. Chomsky's Tree Simplification Process

Figure 2.5 is a simplification of how the structure transforms. The first represents the ability of any deep structure while the second is the apparent pronunciation that passes through all transformational guidelines.

2.2.3. Preprocessing Methods and Techniques

Preprocessing is essential for every Natural Language Processing task. The general steps needed for preprocessing of a text in depend on the targeted requirement or application. Preprocessing steps are among the most important steps we focused on it this section to understand the other works which related with Arabic preprocessing methods and how they applied it in the previous works.

In research by led Osama in 2012 [29], an evaluation of the stemming methods related to clustering Arabic documents was carried out to find the best method for the complicated language. This work focused on three types of stemming: no stemming, stemming using roots, and light stemming. Their dataset came from BBC Arabic and their 4,763 documents were segregated into the following categories: World News, Middle East News, Business and Economy, International, Science and Technology, and Art and Culture. The results indicate that the light stemming gets the best measurement values when compared to no stemming and root-based stemming in Arabic document clustering. The results of precision recall and F-measure for three stemming cases are as follows. The first type, without stemming, has values: 0.6, 0.6 and 0.61 for precision, recall, and F-measure respectively. The second type is light stemming and has values: 0.75, 0.7 and 0.72 for precision, recall, and F-measure respectively. The last type is root-based stemming and has values: 0.54, 0.53 and 0.54 for precision, recall, and F-measure respectively. In observing the results of their experiments, it is clear that light stemming has the best values due to Arabic's complicated morphology. Due to Arabic's high inflection, stemming using roots has a great deal of backfire.

A team led by Duwairi in 2007 [30] compared three dimensionality reduction techniques: stemming, light stemming, and word clustering. Duwairi used KNN to perform the comparison. The performance metrics used were: time, accuracy, and the size of vector. She showed that light stemming is the best in term classification accuracy. Duwairi collected 1,500 documents belonging to one of three categories (sport, economics, and education). Each category has 5,000 documents. She split the corpus into 9,000 documents for training and 6,000 documents for testing. In 2008, a team led by Thabtah [31] investigated different variations of VSM and term-weighting approaches using the KNN algorithm. Her experimental results showed that the Dice distance function with tf-idf achieved the highest average score. The authors used the corpus collected by Al-Shalabi in 2006 [32]. Said and a team in 2009 [33] provided an evaluation study of several morphological tools for Arabic Text Categorization using SVMs. Their study included using the raw text, the stemmed text, and the root text. The stemmed and root text were obtained using two different preprocessing tools. The results revealed that using light stemming combined with a good performing feature selection method, such as mutual information or information gain, enhances the performance of Arabic Text Classification for small-sized datasets and small threshold values for large datasets. Additionally, using the raw text leads to the worst performance in small datasets, while its performance was among the best tools in large datasets.

2.3. Arabic Text Classification

Different studies address the problem of text classification using various techniques to classify text documents and different metrics to evaluate the accuracies of these techniques. We present in this section a number of studies and experiments of text classification conducted on Arabic language using different datasets, different features, and different classification algorithms.

For instance, research by Sawaf and his team in 2001 [13] used a statistical approach based on the maximum entropy technique to classify the Arabic NEWSWIRE corpus of the Linguistic Data Consortium (LDC) of the University of Pennsylvania which covers four classes: politics, economics, culture and sports. The main objective was to simplify Arabic classification difficulties by avoiding morphological analysis and to use sub word units (character n"grams). The best accuracy reported was 62.7%. Also, in the same research field, a team led by El Kourdi in 2004 [14] used the Naïve Bayes algorithm to classify 300 web documents into five classes: health, business, culture, science and sport. The average accuracy achieved was 68.78% and the best accuracy reported was about 92.8%. On the other hand, research from the team of Syiam in 2007 [15] showed through experimentation that the suggested hybrid method of statistics and light stemmers is the most suitable stemming algorithm for the Arabic language and gives a general accuracy of about 98%. In 2008, a team led by Mesleh [16] applied a support vector machine (SVM) to classify Arabic articles with Chi “Square feature selection. The corpus contained 1445 documents that vary in length and were collected from online Arabic newspaper archives. They were divided into nine categories: Computers, Economics, Education, Engineering, Law, Medicine, Politics, Religion, and Sport. They focused on Feature Subset Selection (FSS) methods for TC tasks with a particular interest on the usage of an ant colony optimization algorithm to optimize the process of FSS. They compared Ant Colony Optimization Based “Feature Subset Selection Method (ACO Based"FSS algorithm) with six advanced feature subset selection methods: Chi"square, Information Gain, Mutual Information, NGL coefficient, CSS score and Odds Ration in the classification of Arabic articles. Compared to the six classical FFS methods, the ACO Based"FSS algorithm achieved better text classification and was adapted to handle the large number of features in theme classification tasks. In another attempt in 2009, Harrag and El"Qawasmah [17] applied neural networks (NN) on Arabic text. Their experimental results show that using NN with Singular Value Decomposition (SVD) as a feature selection technique gives better result (88.3%) than the basic NN (without SVD) (85.7%). They also experienced scalability problems with high dimensional text dataset using NN. Harrag collected his corpus from Hadith encyclopedia (◌̊QRSTSU ˡKLMNO from the nine books. It contains 435 documents) ˡVLWXO belonging to 14 categories. He applied light stemming and stop words removal on his corpus. In 2010, Hammouda and the team [18] proposed a system of indexing and classification of Arabic texts enabling the text classification using a heuristic approach. The Arabic dataset is collected from Internet, newspapers and magazines. They collected 25 documents to manually produce a corpus composed of around 12,500 words, with 5 documents for each of the following 5 categories: sports, medicine, politics, economics and agriculture. The system performance was evaluated on the different chosen domains, achieving 90% of precision and 85% of recall. In this investigation, we propose two theme classification methods: a Semiautomatic method (which we called: SACM) and an Automatic method (which we called: ACM) that are tested on an Arabic corpus containing five different themes.

2.4. Text Classification Methods for Other Language

The following information relates to methods of feature selection found in the literature. In 2005, Neumann [55] proposed a SVM-based feature selection and classification method based on four innovative methods that worked to improve the classifier. The study also employed linear and nonlinear SVM classifiers to regularize the data and feature selection that was embedded. Novovicova's 2005 study [56] introduced algorithms based on information-theoretic feature selection with forward selection techniques using enhanced mutual-information criteria functions. In 2005, Liu and Schumann analyzed data mining feature selection for credit scoring models. They provided an empirical study of four machine learning feature selection methods: ReliefF, Correlation-based, Consistency-based, and Wrapper algorithms. They also analyzed the improvements to the performance of scoring models.

In 2005, Sanchez-Marono and her team [57] introduced a technique to select feature subsets using a wrapper. They used ANOVA decomposition as the basis for the development of a new wrapper feature subset selection method, in which functional networks are used as the induction algorithm. Peng's team in 2005[58] derived an equivalent form, called minimal-redundancy-maximal-relevance criterion (MRMR), for first-order incremental feature selection. They presented a two-stage feature selection algorithm by combining MRMR and other more sophisticated feature selectors (wrappers, for example) which was used in the selection of a compact set of superior features at minimal costs. Gasca and the team in a 2006 research [59] proposed to eliminate redundancy and irrelevance using a new MLP-based feature selection method. Morariu's team in 2006 evaluated different feature selection methods, including: random selection, Information Gain, Support Vector Machine (SVM_FS) and genetic algorithm based SVM (GA_FS) feature selection methods. They obtained the best result from SVM_FS and GA_FS. Singh and Liu [60] proposed feature subset selection bias for classification learning in a 2006 study. They analyzed its statistical properties and factors that affect the feature selection bias, also they found that the bias may not cause negative impact classification. A 2006 study by Liu and Zheng [61] presented a novel feature selection method named filtered and supported sequential forward search (FS_SFS) in the context of support vector machines (SVM). They found that FS_SFS has two important properties to reduce the time of computation. First, it dynamically maintains a subset of samples for the training of SVM. Second, both the discriminant ability of individual features and the correlation between them are effectively filtered out nonessential features. Research in 2007 by Zhao and Liu [62] introduced searching for interacting features. Their proposed algorithm efficiently handled feature interaction for efficient feature selection. Gomez and her team in a 2007 study [63] proposed information-theoretic feature selection for the classification of hysteresis curves. Their algorithm extracts a large number of features with the highest content of information and then selects the appropriate ones through a mutual information criterion. Zhang's research in 2008[64] presented a new filter method for feature selection with pair wise constraints. The team proposed to use pair wise constraints, which specify whether a pair of data samples belongs to the same class (must-link constraints) or different classes (cannot-link constraints). They compared the performance of their proposed algorithm with the well-known Fisher Score and Laplacian Score algorithm. In 2009, Woo Cho [65] proposed a new feature selection method using orthogonal filtering and nonlinear representation of data for an enhanced discrimination performance. He implemented orthogonal filtering to remove unwanted variation of data. The proposed method adopts kernel principal component analysis, one of nonlinear kernel methods, to extract nonlinear characteristics of data and to reduce the dimensionality of data. A team led by Li in 2009[66] proposed mutual information-based feature selection for Case Based Reasoning (MICBR) in software cost estimation. This approach combines the 'wrapper' and 'filter' mechanisms into a hybrid, which is another kind of feature selector with much lower complexity than wrappers, and the features selected by filters are likely to be generalized to other conditions. Akay's 2009 study [67] proposed Support Vector Machines combined with feature selection for breast cancer diagnosis. He conducted different experiments on different training-test partitions of the Wisconsin breast cancer dataset (WBCD), which is commonly used among researchers who use machine learning methods for breast cancer diagnosis.

2.4.1. Classic Algorithms for text Classification

Supervised machine learning is a field of study where algorithms are produced based on external situations in the formation of hypotheses that generalize them in order to predict future situations. Its goal is the formation of concise models that distribute labels related to class as predictive features. The classifiers produced then classifies areas where the features are preknown by the system and the label is unknown. What follows is a compilation of related literature on the topic of classifier schemes.

In 2005, Baglioni and her team [68] introduced DrC4.5 which improved J4.8 through prior knowledge. The work based on improving classifiers and used Bayesian Casual Maps as the casual dependencies in this variation of the famous J4.8 algorithm. Li's 2006 [69] study was an analysis of issues with complex classification based on preset rules with the aim of predicting absent values or digits in the dataset. This technique was formed out of a widely-used classification method using rules and identified how complex a ruleset can be within a hierarchy through k-optimal rulesets. The lessening of the size connecting to the lessening of the complexness. Through the work, it was shown that the k-optimal ruleset classifiers are more complex than some of the benchmark classifiers like CBA and C4.5. Aizawa's 2000 study [70] discussed the feature quantity that is formed from theoretical opinions of the cooccurrence of terms and documents. The work described this feature quantity and then applied it to various datasets for retrieving the desired information. In 2002, Rogati and Yang [71] introduced a great deal of filter feature selection tools that can be applied to classifying text and discovered that the technique using $\chi 2$ statistics had a consistent optimal performance compared to information gain in every Naïve Bayesian, SVM, and k-nearest classifier. In 2002, Fransconi [72] suggested a broader text categorization that allows documents to arrange a page sequence. This study employed a hybrid hidden-Markov technique that categorized documents of several pages and discovered that errors can be lessened in Naïve Bayes when using contextual data prior to classifying. Sebastini's 2002 study [1] discussed machine learning as it related to categorizing text and surveyed the major methods used before highlighting their common issues. He named three such problematic areas: representing documents, building classifiers, and evaluating their potential. Forman's 2003 study [73] extensively analyzed feature selection metrics for classifying text in the discovery that the suggested Bi-Normal Separation (BNS) using F1 was better than the other methods and had the best results when using information gain. Liu's 2003 study [74] suggested term contribution (TC) for feature selection after extensively comparing the viability of various methods such as: document frequency, term-strength, methods using entropy, and information gain with contributed terms. The results showed that the suggested TC model had better performance than the compared methods. A team led by Lewis in 2004 [75] analyzed the Reuters Corpus Volume I (RCVI) for the production of a new benchmark in categorizing text. The work discussed the coding policies and quality control methods that were used to produce RCVI and included the desired semantics of the hierarchal category taxonomies, along with the necessary fixes for removing erroneous data. Bergsma's 2004 study [76] proposed a similarity-based feature weighting for categorizing text using a database that was automatically clustered into dependency-based word-similarity to discover similar terms. The findings showed that growing the key terms' weight allows for more effective text categorization.

A team led by Lan in 2005 [77] experimented on the existing term-weighing methods that employ SVM before introducing a better method named tf,rf that improves the power of discriminating the terms. A team led by Tan in 2005 [78] refined text categorization through enhancing the centroid classifier and Naïve Bayes classifier algorithms, which outperformed the standard SVM. A team led by Wanas in 2006 [79] studied 14 local and global feature selection tools, before suggesting their standard deviation (STD) and maximum deviation (MD) globalization methods, which showed much better performance than the methods they studied in document frequency. Xu and Li in their 2007 study [80] suggested a tool to weigh terms from various categories, choosing the top six from the Reuter-21578 corpora. This work discovered that TM2 is able to consider the frequency of documents that contain a given term and also how much that term occurs in the various documents across the categories. In their work, they fixed 2,000 features before randomly selecting 4, 8, 12, 16, and 20 categories five times each. They discovered that TM2 has a different performance with an increasing category list. A team led by Chen in 2009 [81] introduced feature selection for classifying text with Naïve Bayes with two metrics of evaluation that were applied to datasets of various classes: the multi-class-odds ratio and class discrimination. Their work showed that their suggested metrics had a better performance than similar methods.

CHAPTER: THREE

MATERIAL AND METHODS (PROPOSED TECHNIQUE)

3.1. Dataset collection (Arabic Text)

Although Arabic is the 5th most-spoken language in terms of native speakers according to the Internet World State ranking, it is considered as a poor content language over the web, unlike English, with very few web pages specializing in Arabic review. We have searched for a source which communicates real opinions and at the same time the opinions are written in Arabic. We decided to use a large scale Arabic dataset which we founded in LABR, partly because it's free and open source [8]. The method for collecting and annotating the dataset and present its properties and statistics was determined using the following tasks:

Table 3.1. Important Dataset Statistics.(LABR.[8])

Number of reviews	63,257
Number of users	16,486
Avg. reviews per use	3.84
Median reviews per user	2
Number of books	2,131
Avg. reviews per book	29.68
Median reviews per book	6
Median tokens per review	33
Max tokens per review	3,736
Avg. tokens per review	65
Number of tokens	4,134,853
Vocabulary words	79086
Number of sentences	342,199

They downloaded over 220,000 reviews from the book readers social network www.goodreads.com during the month of March 2013. These reviews were from the top 2,143 books on the list of Best Arabic Books. They filtered out the non-Arabic reviews and performed several pre-processing steps to clean up HTML tags and other unwanted content. At the end, they ended up with 63,257 Arabic reviews. We also observed all the sentiment that was given for the Arabic text in the event that there were any mistakes to be found.

Table 3.2. Samples from our applied dataset

Text id	Book Reviews Text Translate to English	Book Reviews Text in Arabic	Text Sentiment
1	I appreciate this novel	إني أقدّر هذه الرواية كثيرا	POS
2	Having deepened it and found it less than my expectations	بعد ان تعمقت فيها وجدتها اقل من توقعاتى	NEG
3	The splendor of the author's style is very simple	من روعة إسلوب الكاتب السلس البسيط للغاية	POS
4	It is not pleasant overall, and its events are not interesting	ليست ممتعة فى المجمل واحداثها غير مشوقه	NEG
5	From what I read the coolest. The style of Joseph Zidan is wonderful	من أروع ما قرأت. أسلوب يوسف زيدان رائع	POS
6	This novel made me feel that I had benefited almost nothing	هذه الرواية جعلتني أشعر أني لم استفد شيئا تقريبا	NEG
7	Very nice novel and interesting events from first to last page	رواية جميلة للغاية وأحداث مثيرة للاهتمام من الصفحة الأولى حتى الأخيرة	POS
8	This novel is poor in terms of vocabulary	هذه الرواية رديئة من ناحية المفردات اللغوية	NEG
9	This story flows in the spirit. It is safe. Relaxing	هذه الحكاية متدفقة بالروح. إنها آمنة. مطمئنة	POS
10	This novel did not fascinate me, and I never enjoyed it	لم تبهرني هذه الرواية و لم أستمتع بها ابدا	NEG

3.2. Text Pre-Processing

It is necessary to preprocess the text data before the feature vectors can be extracted. To do so, tokenization is required, which is the identification of each individual word and the reduction of the typographical variations. After, the preprocessing technique is applied to the clean text that involves normalizing the text, stemming, and removing stop words. The tokenization process is made simple through the NLTK library. Once the tokenization is finished, the various steps involved in the preprocessing are carried out accumulatively for a final normalized and stepped text without any stop words. The steps can be seen in Figure 3.1 below.

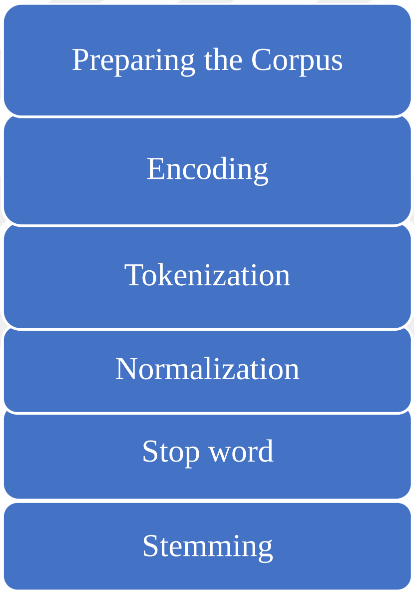

Figure 3.1. Pre-Processing Stage

3.3. Preparing the Corpus

Preparing the corpus is one of the most important stages in the approach. The corpus is a collection of documents in one domain. We use these documents in the process of sentiment analysis which include many different steps of analysis the Arabic text. We used 63,257 documents from LABR and we build the framework depending on it. All the documents used are in plain text when we load them into Gate to facilities the processing.

3.4. Encoding

This step deals with unifying the encoding to avoid character appearance problems and to standardize any dataset for future use. Also, it is used in the normalization process.

Encoding is a numbering scheme that assigns each text character in a character set to a numeric value. A character set can include alphabetical characters, numbers, and other symbols. Different languages commonly consist of different sets of characters. So many different encoding standards exist to represent the character sets that are used in different languages. Windows CP-1256 encoding ("Unicode Windows 1256", 2015) is a code page used to write Arabic, Persian, and Urdu under Microsoft Windows, it encodes every abstract single letter of the basic Arabic alphabet, not glyphs. Another popular encoding is UTF-8 ("Unicode 8.0 Character Code Charts" 2015) which is a variable-width encoding that can represent every character in the Unicode character set. We choose UTF-8 for all text content by converting any content to UTF-8[88].

3.5. Tokenization

Tokenization is the procedure of analyzing and splitting the input text into several tokens such as, number, word, space, symbol, etc. as shown in Figure (3.2). It is necessary step in our preprocessing text.

One of the initial steps of preparing any content corpora is to isolate the info content into appropriate units. These units could be characters, words, numbers, sentences or some other fitting unit. The meaning of a word here isn't the correct syntactic shape that is the reason we consider it a 'token'. A token could allude to a syntactic word, a number or, as in Arabic a whole grammatical phrase (e.g. (وسنساعدهم)"and we shall help them"). The way toward removing tokens is called tokenization[89]. Tokenization is the assignment of hacking it up into pieces, called tokens, maybe in the meantime discarding certain characters, for example, accentuation. A token is an occasion of a succession of characters in some record that are assembled together as a valuable semantic unit for handling. A sort is the class of all tokens containing a similar character succession. A term is a (maybe standardized) type that is incorporated into the IR framework's word reference [90].

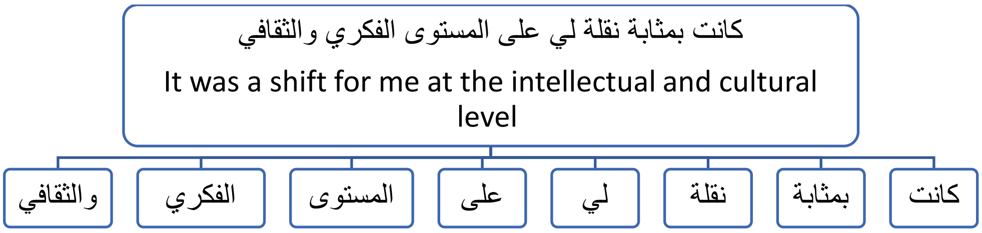

Figure 3.2. Tokenization Process

3.6. Normalization

Normalization makes the Arabic text a consistent and clean form which means that all of the different forms of a given word are changed into a common form. For example, the word “انت” can have many different forms like “إنت”"أنت", “انـــــــــــــت”, etc. Each of the different forms are counted as different words so they must be changed into a common form. Python is used to implement the package to adjust these words. It carries out various tasks, such as removing the diacritical marks from words, removing the “ء” (Hamza), changing both "ي"and "ى"to "ي"(y), and more. The normalization library is quite effective at normalizing the texts [91] with the rules explained by Table 3.3 below.

Table 3.3. Normalization Rule

Rule	Example
Tashkeel	حدثنا > حَدَّثَنا
Tatweel	الله >اللـــــــــــــــــه
Hamza	ء -> ء or ىء or ؤ
Alef	إ -> ا or أ or آ
lamalef	لإ -> لا or لأ or لآ or لا
Yeh	ى -> ي or ي
Heh	ة -> ة or ه

3.7. Morphological Analysis Techniques (Stemming and Light Stemming)

Stemming algorithms are necessary for a great deal of applications including data compression, the processing of natural language, and retrieving information. The latter is used for the Arabic language. Through the application of stemming algorithms for feature selection, there is a reduction in the number of features. This is useful in Arabic because the lexical forms of different words are formed from the basic roots. Snowball library which is a collection of stemming algorithms, is used in stemming process. As such, the features produced from a common root represent one feature (the word’s stem)

[92]. A great deal of stemmers were designed for English and European languages that are tasked with the removal of suffixes. However, special stemming algorithms are necessary for Arabic, as outlined by the following reasons [93]:

1. Because Arabic is a Semitic language, it has a vastly different structure than the Indo-European languages of English and French.
2. Arabic depends on roots and word templates for producing meaningful words.
3. The consonants of any Arabic root may change or be removed through the language's morphology.

The three stemming approaches are: root-based, light, and statistical stemmers, as shown in Figure 3.3 below.

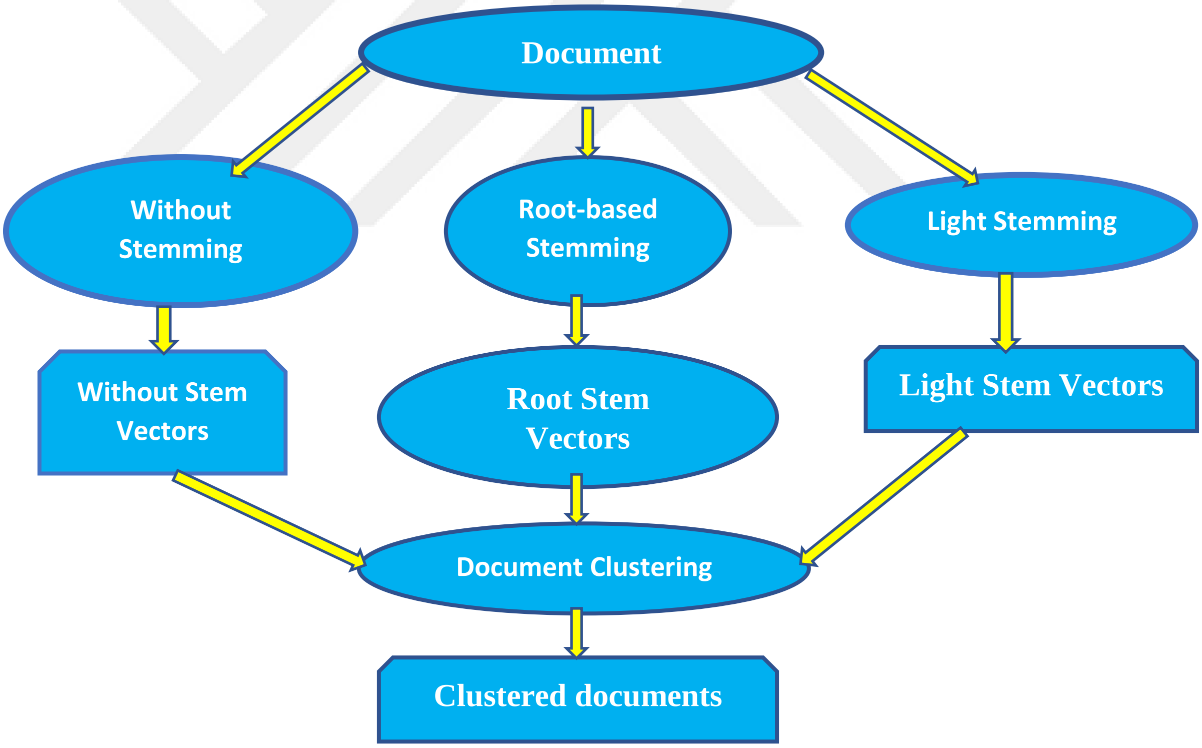

Figure 3.3. Stemming System Architecture

3.8. Root-based Stemming

This type of stemming utilizes root extraction and analyzes the morphology of Arabic words. It is shown in Figure 3.4 below [56].. It should be noted that words in Arabic all derive from a root common to the meaning. For example, the three words (الكتاب الكاتب

المكتبة) mean "the library", "the writer" and "the book" and are reduced to one stem (كتب) meaning “write” [94].

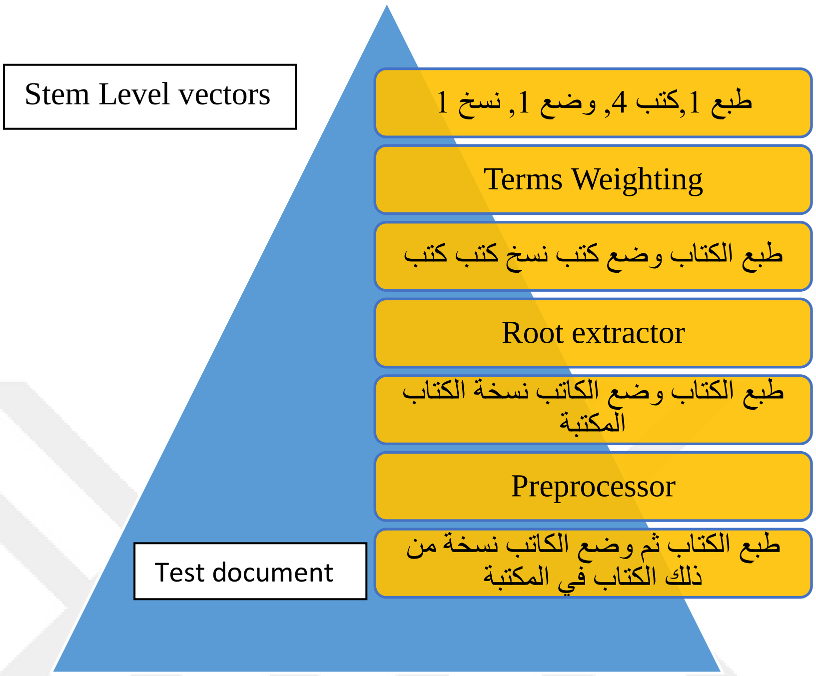

Figure 3.4. Preprocessing with root-based stemming

3.9. Light Stemming

Light stemming is based on the premise that although there are many word variations derived from a common root, they may not possess similar meanings. To this end, the root extraction algorithms can impact word meanings. Through light stemming, words are properly categorized and retain their meanings by removing prefixes and suffixes to extract the root [95]. The process is demonstrated by Figure 3.5 below. It deserves mention that through light stemming, the difference between (الكاتبونand الكتاب) is recognized, which mean "the book" and "the writers.” The light stems of these words are (كاتب كتاب), meaning “book” and “writer” [52].

Algorithm 4.2: Arabic Light Stemming Algorithm Steps [96]**.**

1. Normalize word:

- Remove diacritics.

- Replace (أ)(إ)(آ) with (ا).

- *Replace(ة) with(ه)*

- *Replace(ى) with(ي)*

Stem prefixes:

2. Remove prefixes: (و), (لل), (وال), (بال), (كال), (فال)

Stem suffixes:

3. Remove suffixes:(ي) ,(ه) ,(ية) ,(ين) ,(ون) ,(ات) ,(ان) ,(ها)

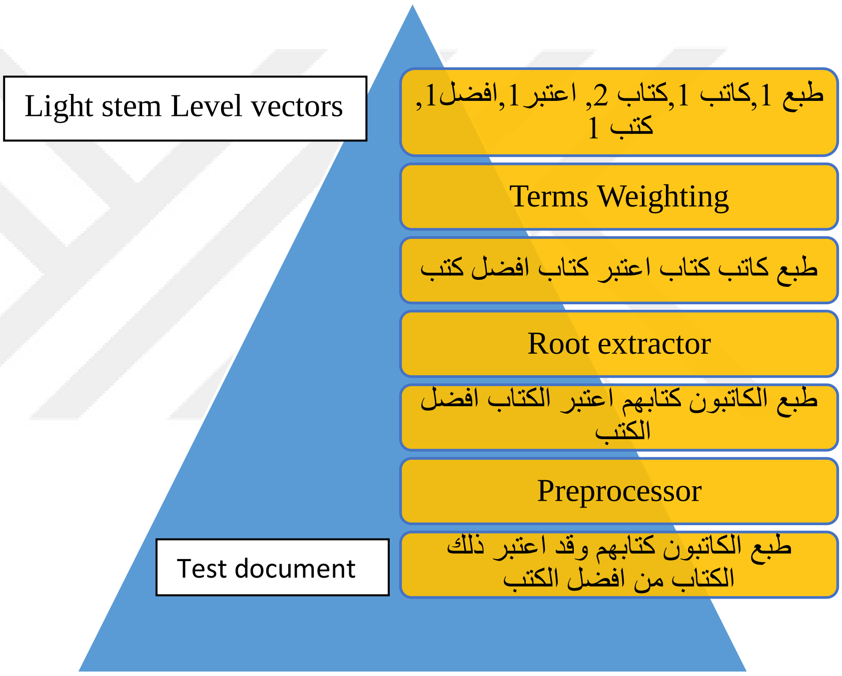

Figure 3.5. Preprocessing with light stemming

3.10. Stop-Word Removal

Stop-word removal is a procedure of eliminating language words that do not carry any significance to a text or carry little meaning. The removal of the stop-word changes the document length and subsequently affects the weighting process. Also, it can increase the efficiency of the indexing process as 30% to 50% of tokens in a large text collection can represent stop-words Categories of stop-words cover adverbs, conditional pronouns, prepositions, and pronouns, transformers such as verbs, and letters, referral names and affixes such as prefixes, infixes, and postfixes. We used Natural Language Toolkit (NLTK) library which has 750 stop words for Arabic Table (3.4) lists some of Arabic stop-words.

The list of Arabic Stop-words will be used and updated by prevent remove some stop-word in documents. We eliminate any matching between stop- word list and my dataset words[97].

Table 3.4. Stop-Word List Samples

حول	اما	الذي	تلك	أن	وكانت	عنه
إما	دون	بعد	وحتى	حين	اللاواتيً	ضد
حتى	وتلك	وعلى	والذي	كذلك	لكن	الذي
وكذلك	ان	وكان	ومنذ	وليسٌ	هذا	إن
وكان	اللذان	أما	عليها	وهو	ومن	اللتان

3.11. CNN Model Discerption

The Convolutional Neural Network model applies convolutions over the inputs to compute the outputs, in contrast with other neural networks that only connect each input neuron to each output neuron of the previous layer. In CNNs, each convolutional layer applies filters over the data, and during this process, the model learns the values of its filters based on the task that we want to perform. At the end of this process, the results of each layer are combined. In this case, the architecture used is composed by three filters with length 2, 3, 4, and 5. The CNN model was developed with TFLearn3 which a modular and transparent deep learning library is. It is built on top of TensorFlow and is designed to provide a high-level API towards TensorFlow to speed-up the development of deep learning models [13]. It supports different kinds of deep learning models such as LSTM (long short-term memory), feature extraction networks and in this case convolutional neural networks[12].

3.11.1. Evaluation Methodology

We try different CNN structure in our thesis study for example figure () are used as cascade structure but we did not get a good result from it. On the other hand, channel structure in figure 3.9 we used and we get good results from it. The evaluation methodology has components that are specific for each task and dataset. However, a common methodological aspect is present in all the neural network models, namely the use of a callback with the early stopping function. Specifically, the early stopping function allows us to stop the training process when a monitored quantity has stopped improving, in this case the validation loss. The validation loss is measured in each training epoch through a validation set that has been previously defined. The validation set was defined as the 10% of the training data, and the training procedure will stop when we observe at least two consecutive epochs with no improvements. Each model's training session was between 8-14 runs, which means that they trained on the training data between 8-14 times. Once described the common methodology, is time to start the oriented description of the evaluation methodologies for each specific task and dataset.

The text classification task was performed on LABR dataset with two different experimental. The first one polarity classification it is taking negative and positive sentiment in our case, positive reviews are represented by 4 and 5 rating, and negative reviews are represented by 1 and 2 rating. Also, 3 represent the neutral reviews is not taking in the polarity classification. Moreover, we split the LABR dataset to balanced dataset and unbalanced dataset and we determine both of their results. The second type is the rating one we considering all rating numbers from 1 to 5. Also, we split our dataset to balanced and unbalanced dataset and determine both results.

3.11.2. Training CNN Model

Through the thesis project, several convolutional neural networks were developed. They were developed with the same architecture, but the parameters for each layer were different. The architecture in figure (3.9) is the architecture used in all models. All models have four convolutional neural networks. Moreover, the following parameters we used in the models: the dropout, embedding output dimension and filter numbers.

3.11.3. Input layer & Embedding layer

Input data takes two arguments as input: a shape and the name for the input layer. This will be the component that feeds data to the neural network. The name input is optional, and it will be the name for the input component, so it can be retrieved from TensorFlow’s collections.

The shape is the important input in this layer since it will describe the shape of the data being fed to the neural network. For instance, if a picture would be in the shape of [32, 32, 3], then it is three-dimensional. 32 x 32 pixels and three for the RGB colors. In this thesis, the shape is one-dimensional and consists of the longest sentence.

'None' represents the batch size which will be described later. The output of this component will be a 2D-tensor, the shape of the input data which is: [None, max_words] where None is the batch size.

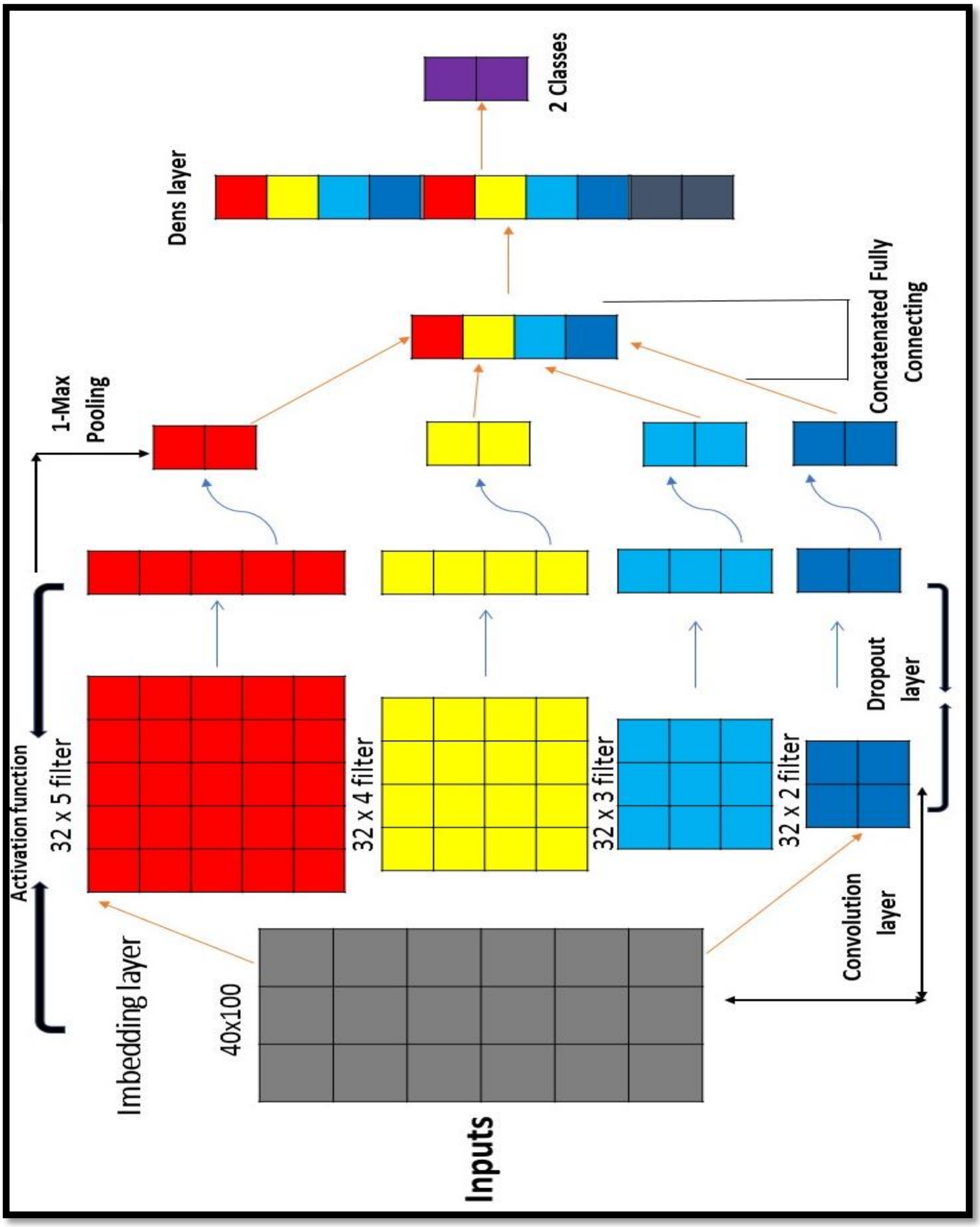

Figure 3.6. CNN Model channel Architecture

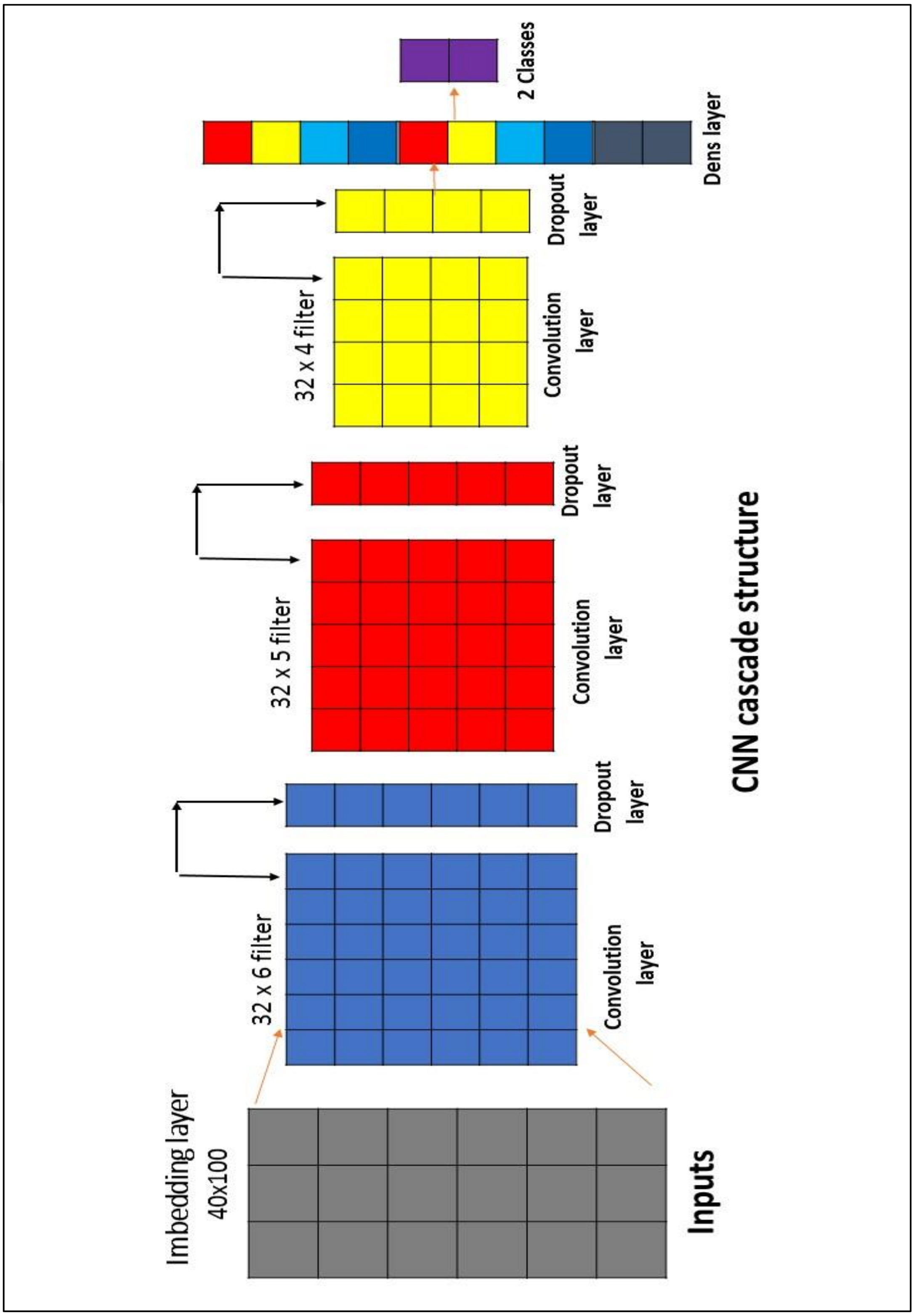

Figure 3.7. CNN Model cascade Architecture

3.11.4. Convolutional layer

As described before, the convolutional layer is the core building block in a Convolutional neural network and is always the first layer. The first parameters are the

2D-tensor from the previous layer. The second and third parameters describe how many filters and their sizes that are used in each layer. So, with four different filters, the sizes will be 32 x 2, 32 x 3, 32 x 4, 32 x 5. The fourth parameter, padding, describes how the matrix is going to be handled while sliding the filter over it. With 'same' the matrix is padded with a border of zeros around it to help the filter to convolve all numbers. If the padding is set to 'valid' the filter is just sliding over the numbers it can reach, this means that some values might be skipped [98].

The fifth parameter describes the activation function which in this thesis is the ReLU operation. The algorithm for ReLU is Max (0, input), and is applied after each pixel is convolved.

The purpose is to introduce non-linearity to the network. ReLU will save all values above zero and change all negative values to be zero. The idea of introducing a non-linear activation function is to prepare the network to handle non-linear data, data which cannot be separated by a single line after they have been classified [98].

3.11.5. Expand Dimension and Max pooling

The convolutional layer returns a 4D-tensor and its able to feed the max_pool function, which performs a max pool computation on the incoming tensor. This computation takes the highest value in each window. In figure 3.7 above is an example of how max pool computation could be done on a sentence.

In this example, a 2x2 filter is going over the matrix and takes the highest number in each filter. This is done to extract features from the result given by the convolutional layers. Each number represents a word from the word indices done by the embedding layer. The result of the max pooling layer is a 2D tensor with the result and it is passed forward to the fully connected layer.

3.11.6. Fully connected layer

A fully connected layer is unlike a convolutional layer fully connected to the previous layers all neurons. The fully connected function takes the previous tensor as one parameter, the number of classes to predict (negative and positive) and an activation function in the form of a SoftMax function. To this fully connected layer there is a regression layer, together these two layers take the output from the previous layers and calculate a classification of the sentences. In figure 3.9 below is an overview how the fully connected layer is built.

The SoftMax function is applied to the fully connected layer to output the probabilities of a class. Its “squashes” the output from each neuron to be equally a number between 0 and 1. For instance, a number of 6.77 could be “squashed” to 0.32. Another task the SoftMax function does beside computing the values in a vector is to divide each “squashed” value from the vector to sum up to a total number of 1.

The new vector contains the probabilities that the data belongs to a class, this means the new vector has the probabilities for all classes to predict [99].

Figure 3.9 shows an example of a SoftMax function predicting the probabilities that the vector belongs to one of three classes. Where class number one gets a probability of 0.70 where 1 is the max of the total sum.

The optimization algorithm used in this model is the Standard descent algorithm (SGD) algorithm, the learning rate in this model is 0.001 which is the defined default number for (SGD) optimizer. The learning rate is one parameter that can be changed to increase the accuracy of the model and decreasing the loss.

A loss function is used to measure the error probability between the network’s prediction and the data's true label. This model uses binary_crossentropy because there are two classes, 0 and 1. The cross-entropy function computes how far away the prediction was from the right answer or the right distribution. This means that the network can predict the right answer, being above 0.51 in the prediction after the SoftMax function, but this means that the network might also think that another label could be true by 0.49. So, this loss function calculates how wrong the network predicts the true labels. The score from the loss function is used in the optimization function (in this case SGD) to improve the weights and parameters to decrease the loss score [100].

A new deep neural network based on the convolutional network that was built now. It also takes two parameters that describe the logging functionality. Tensorboard_verbose takes an integer between 0 - 4, where zero is the minimum level of logging and four is the highest. The last parameters describe in which directory to store the logs.

3.11.7. Instantiate Training

Instantiate the neural network and starts the training procedure and takes several parameters. X_train and Y_train represent 90% of the reviews and their classification, and validation set () takes the rest of the dataset to validate the accuracy. Batch size is the size of how much of the dataset that is used for each training step, in most cases 64 is used, and when setting true to shuffle means that at every epoch the data is shuffled. After this function has finished executing, the model is fully trained and can be used to predict future data.

CHAPTER: FOUR

EXPERIMENTAL RESULTS

In this chapter, all results will be displayed for the training and testing in our experimentation. We will consider the validation accuracy as a test accuracy. We divided the results into two parts. The first one will be sectioned into two parts the first one will show the polarity classification results for the balanced and unbalanced datasets. The second part will show the rating classification results with balanced and unbalanced datasets. The last part for our experimental results will display a comparison between our thesis results and the results from the LABR's one.

4.1. Polarity Classification Unbalanced Dataset Results

In this section we will show the different results for unbalanced datasets with the polarity classification. We are going to display the train accuracy with their loss, and validation accuracy with their loss for both we used unbalanced datasets in our experimental. For unbalanced dataset, we used 51,056 reviews. More than 70% of them are positive, or 40,845 positive reviews total. On the other hand, the negative reviews account for 10,211 reviews.

4.1.1. Train Accuracy and Train Loss Results (Polarity Unbalanced)

We have two different datasets we used in the polarity experiment. The first results will show for the unbalanced dataset. And the second one will show balanced dataset.

Table 4.1. Models Train and Loss Accuracy (Polarity Unbalanced)

Dataset	Distribution	Run Num	Epoch Num	Train Loss	Train Accuracy
Polarity	unbalanced	1	9	0.04	98.00%
Polarity	unbalanced	2	8	0.08	97.00%
Polarity	unbalanced	3	8	0.09	97.00%
Polarity	unbalanced	4	9	0.06	98.00%
Polarity	unbalanced	5	9	0.07	97.00%

In table 4.1 we can see the train accuracy. As shown above, for five models we take the highest ones form each model. For model 1 in the 9th epoch the results are 98.00% which is the same as model 4 in the highest train accuracy for unbalanced polarity classification. Graph 4.1 below shows the train accuracy of the five models.

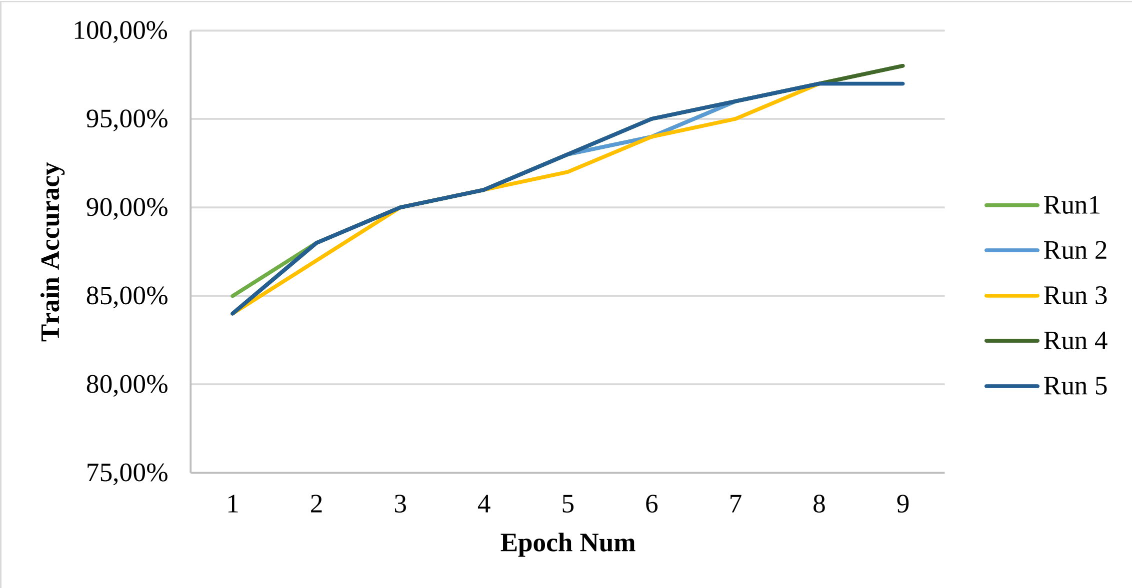

Figure 4.1. Models (Polarity Classification unbalanced Train Accuracy)

Table 4.1 also displays the train loss of the runs on the five models. Model 1 ran on epoch number 9 gave the best train loss rat results. This is because when the lowest rate is reached, it means the train loss is decreased the most, which is the desired result. The train loss for the

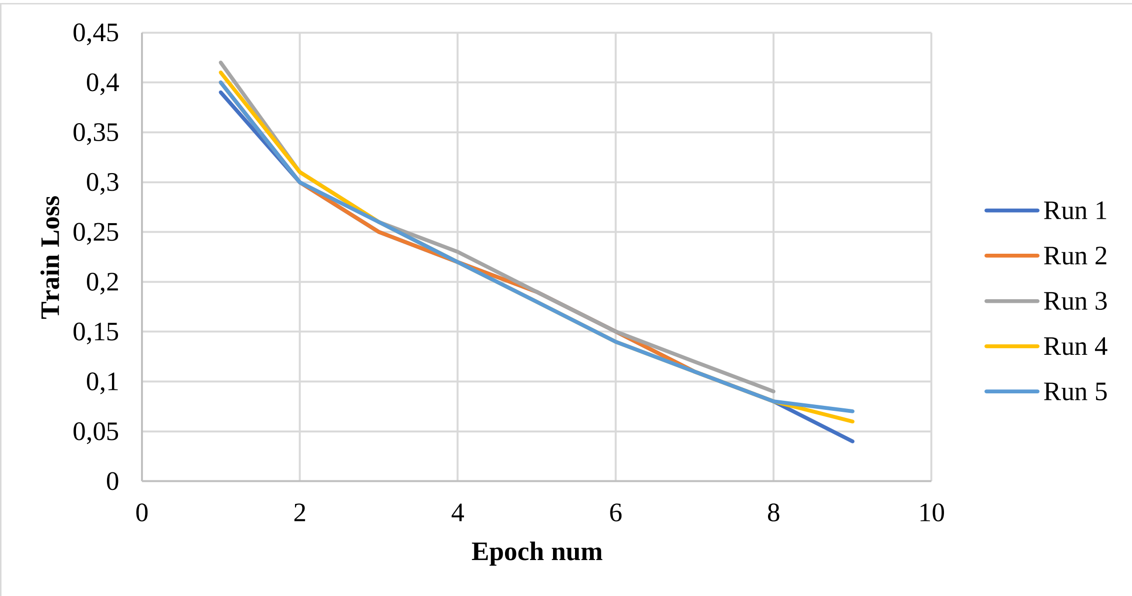

Figure 4.2. Models (Polarity Classification unbalanced Train Loss)

4.1.2. Validation Accuracy and Validation Loss Results (Polarity Unbalanced)

This section shows the validation accuracy for an unbalanced dataset using polarity classification as well as the validation loss on the same dataset.

Table 4.2. Validation Accuracy and Validation Loss (Polarity Unbalanced)

Dataset	Distribution	Run Num	Epoch Num	Validation Loss	Validation Accuracy
Polarity	unbalanced	1	4	0.29	89.00%
Polarity	unbalanced	2	6	0.35	89.00%
Polarity	unbalanced	3	4	0.3	89.00%
Polarity	unbalanced	4	9	0.61	88.00%
Polarity	unbalanced	5	9	0.5	87.00%

As shown above in table 4.2, the highest validation accuracy was for epoch number 4 on the 3rd model at 89%. This is also the smallest validation loss, at 0.3, which means this is the best result from the testing of polarity classification models in an unbalanced dataset.

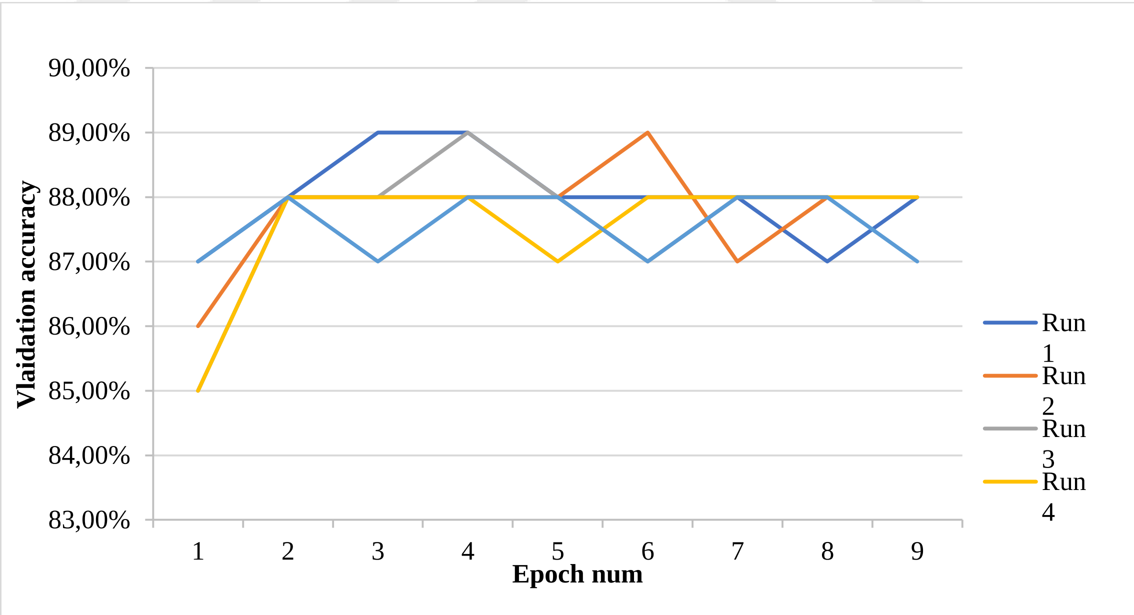

Figure 4.3. Models (Polarity Classification Unbalanced Validation Accuracy)

Figure 4.3 shows the five models for the polarity classification using unbalanced data set. It shows that the 3rd model had the best run of all the models, with the highest validation accuracy and the lowest validation loss.

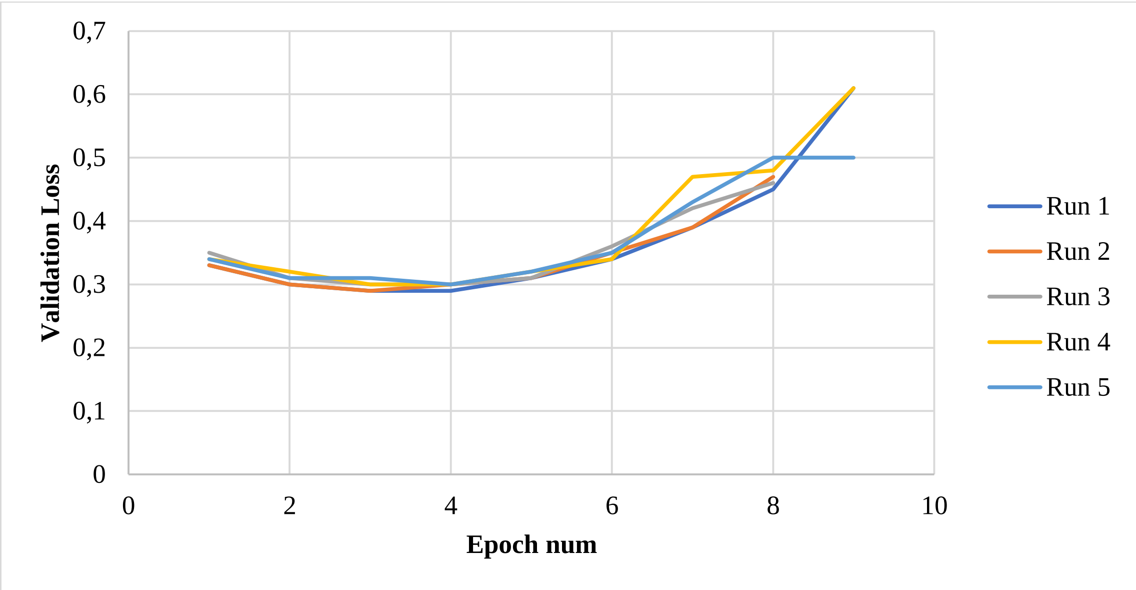

Figure 4.4. Models (Polarity Classification Unbalanced Validation Loss)

Figure 4.4 shows the chart of the five models for the validation loss in polarity classification using an unbalanced dataset.

4.2. Polarity Classification for Balanced Dataset Results

This section shows the results of the polarity classification using a balanced dataset, with the highest accuracy results shown for all the five models. We are going to show in tables and figures both train accuracy and the validation accuracy, and the loss results for the train and test which is represented by validation in our experimentation. In the balanced dataset, we are going to use 16,448 book reviews from all the datasets we have. Half of them are positive reviews and half are negative reviews.

4.2.1. Train Accuracy and Train Loss Results (Polarity Balanced)

The results displayed below relate to train accuracy and train loss for polarity classification on the unbalanced dataset. Table 4.3 below shows that the highest result was found on the 3rd model at 98%, better than the other models. The result comes from epoch number 14 and is the best result not considering the train loss. The smallest train loss results are in the 4th and 5th models on different epoch numbers.

Table 4.3. Models Train Accuracy and Train Loss (Polarity Balanced)

Dataset	Distribution	Run Num	Epoch Num	Train Loss	Train Accuracy
Polarity	balanced	1	9	0.11	96.00%
Polarity	balanced	2	9	0.11	96.00%
Polarity	balanced	3	14	0.06	98.00%
Polarity	balanced	4	10	0.1	96.00%
Polarity	balanced	5	13	0.1	96.00%

The epoch number 14. This is the best result without considering the train loss. The smallest train loss result is show up in the 4th and 5th models with different epoch numbers.

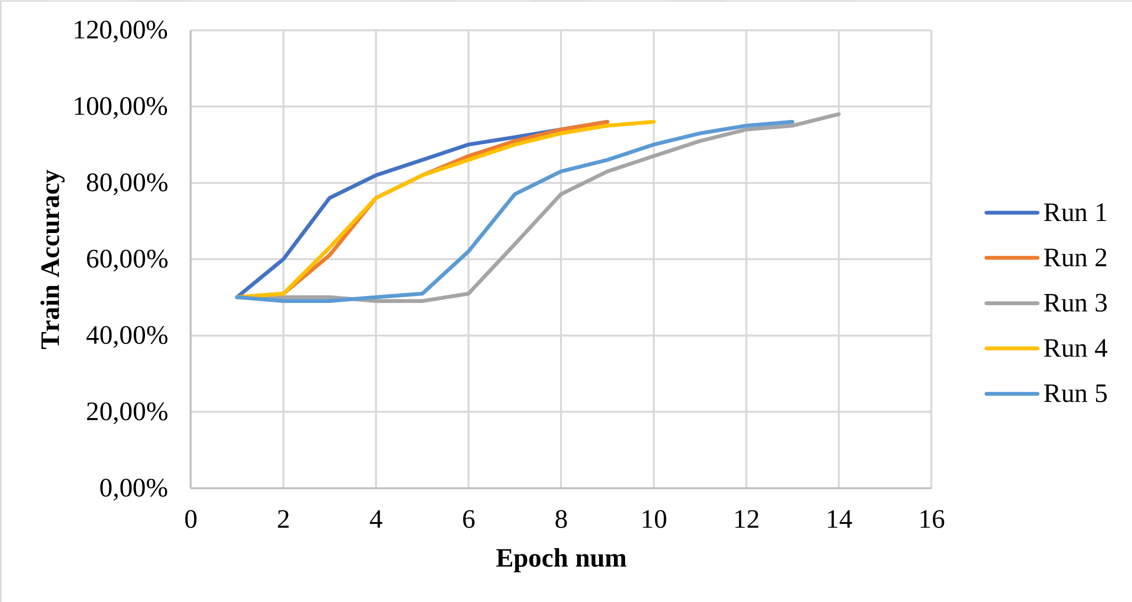

Figure 4.5. Models (Polarity Classification Balanced Train Accuracy)

Figure 4.5 above displays the five train accuracy models for the polarity classification using the balanced dataset.

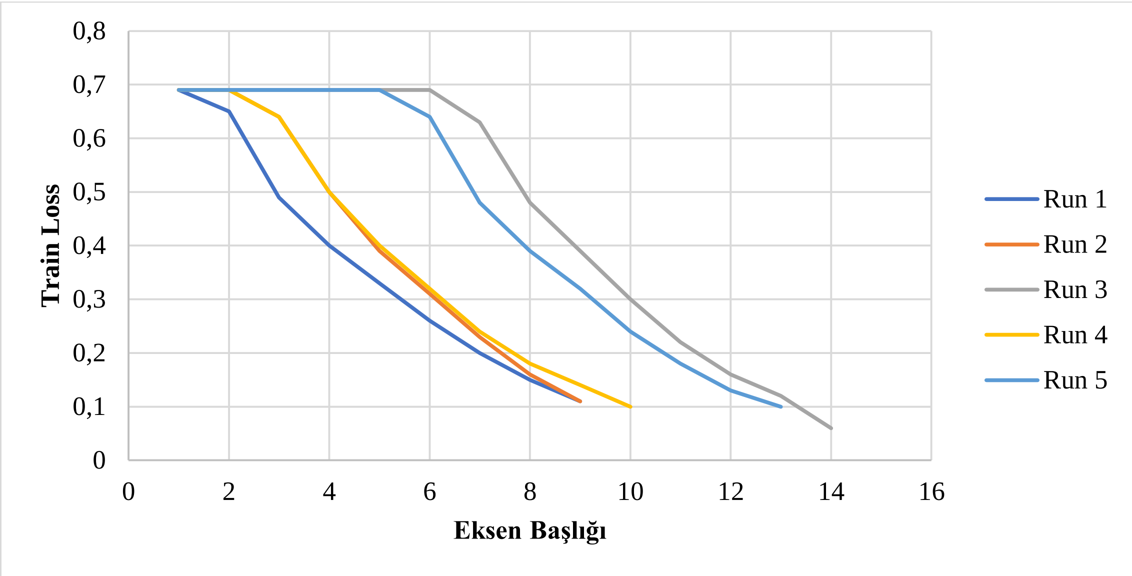

Figure 4.6. Models (Polarity Classification Balanced Train Loss)

Figure 4.6 above displays the train loss for the five models, providing a clear vision to the train loss models decreasing in the last epochs.

4.2.1. Validation Accuracy and Validation Loss Results (Polarity Balanced)

As shown in Table 4.4 below, the highest validation accuracy was 78%, obtained for model 3 in the 9^{th} epoch. This means that after nine runs, we obtained the best accuracy results for our models.

Table 4.4. Models Validation Accuracy and Validation Loss (Polarity Balanced)

Dataset	Distribution	Run Num	Epoch Num	Validation Loss	Validation Accuracy
Polarity	balanced	1	8	0.6	77.00%
Polarity	balanced	2	4	0.48	77.00%
Polarity	balanced	3	9	0.46	78.00%
Polarity	balanced	4	5	0.47	78.00%
Polarity	balanced	5	8	0.48	77.00%

The highest result for the validation accuracy also depends on the lowest validation loss. Figure 4.7 displays a chart of the validation accuracy for the five model runs.

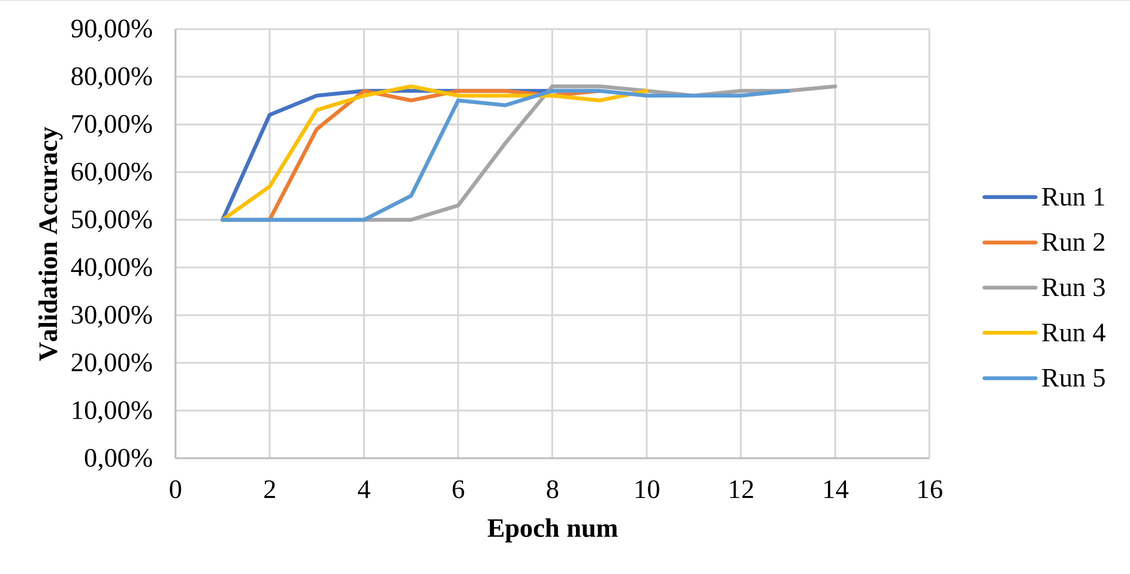

Figure 4.7. Models (Polarity Classification Balanced Validation Accuracy)

It is important to measure validation loss because it provides insight into the loss rating for each model in every run and for each iteration of the models. As such, Table 4.4 shows the highest validation accuracy related to the smallest validation loss. However, as shown in Figure 4.8, it is clear that the lowest and best validation loss was in model 1 at the 8th epoch.

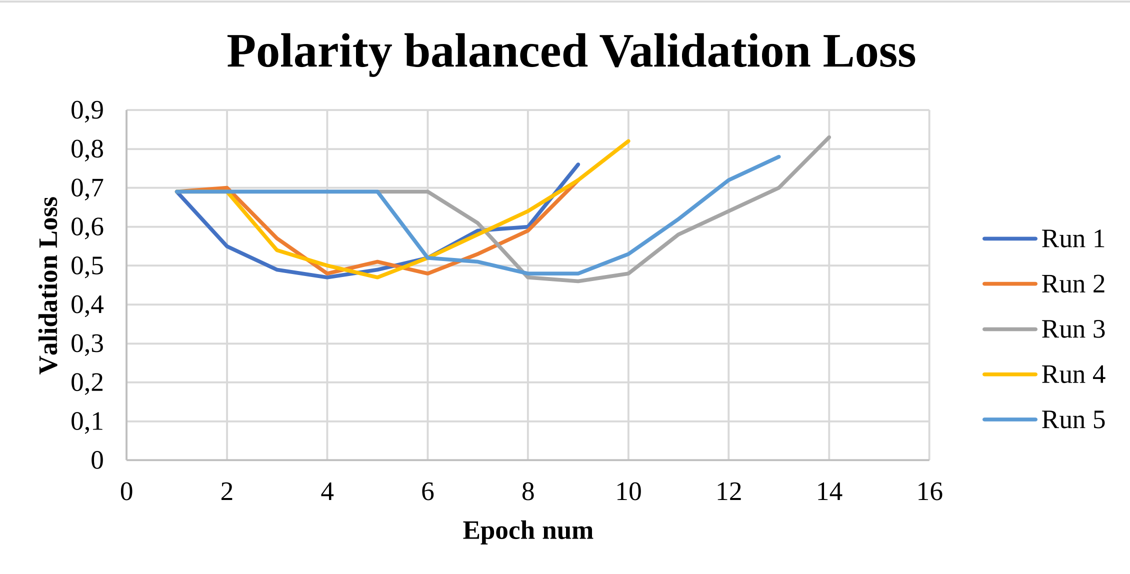

Figure 4.8. Models (Polarity Classification Balanced Validation Loss)

4.3. Rating Classification Unbalanced Dataset Results

This section displays the results of rating classification on the unbalanced dataset, using figures and charts to provide a clear vision of the rating classification and how it was applied on the unbalanced dataset. For the unbalanced dataset, the entire dataset in our corpus was used that contains 63,257 book reviews.

4.3.1. Train Accuracy and Train Loss (Rating Unbalanced Dataset)

Table 4.5 below displays the rating classification on the unbalanced dataset. From the information, it can be seen that the highest train accuracy for our rating classification was 66%, found on the 5th model and 10th epoch.

Table 4.5. Models Train Accuracy and Train Loss (Rating Unbalanced)

Dataset	Distribution	Run Num	Epoch Num	Train Loss	Train Accuracy
Rating	unbalanced	1	9	0.83	63.00%
Rating	unbalanced	2	8	0.91	59.00%
Rating	unbalanced	3	9	0.81	64.00%
Rating	unbalanced	4	9	0.83	63.00%
Rating	unbalanced	5	10	0.79	66.00%

Figure 4.9 below displays a chart of the train accuracy for the five model runs. From the chart, it can be seen that the accuracy ratings increase, beginning at the first epoch until the fifth. It stopped because there is no enhancement in the training accuracy for the last two epochs. This depends on the stopping function.

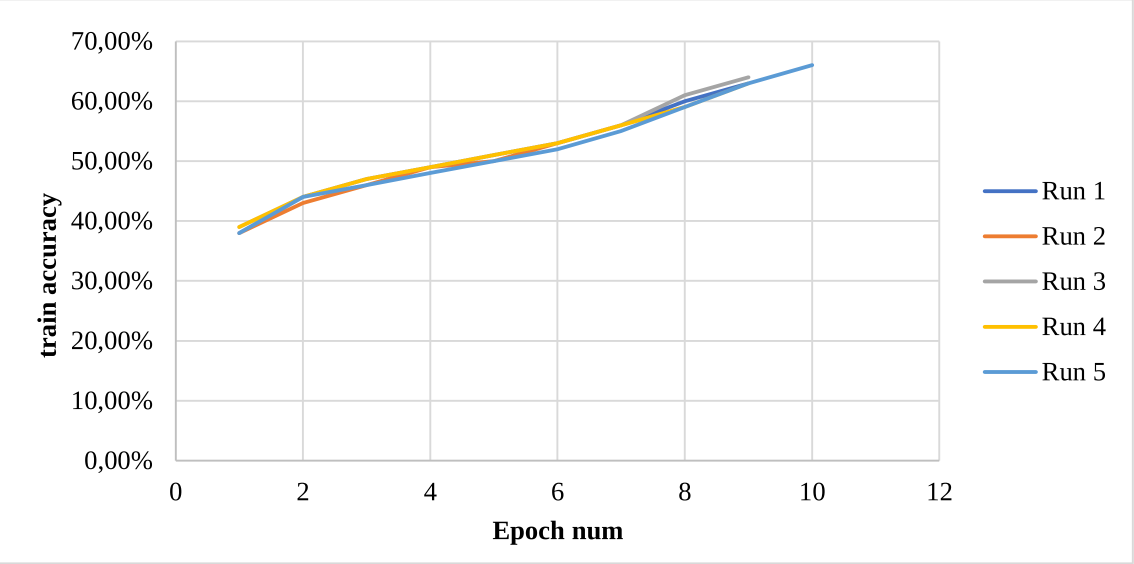

Figure 4.9. Models (Rating Classification Unbalanced Train Accuracy)

Figure 4.10 above displays the train loss for the five models using rating classification on an unbalanced dataset, with the lowest figure at 0.79.

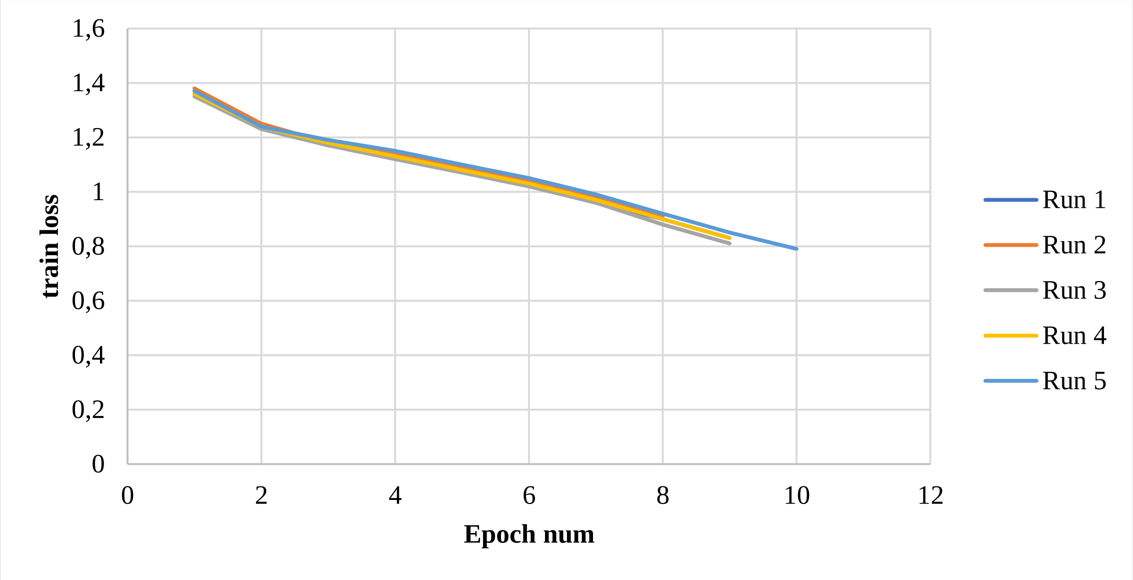

Figure 4.10. Models (Rating Classification Unbalanced Train Loss)

4.3.2. Validation Accuracy and Validation Loss (Rating Unbalanced Dataset)

Table 4.6 displays the results of the five models for validation accuracy and validation loss on the unbalanced dataset. The highest result was 46%, which was for model 1 and 2. However, they received this rating at different epochs: the 4th epoch for model 1 and the 3rd epoch for model 2.

Table 4.6. Models Validation Accuracy and Validation Loss (Rating Unbalanced)

Distribution	Run Num	Epoch Num	Validation Loss	Validation Accuracy
unbalanced	1	4	1.2	46.00%
unbalanced	2	3	1.2	46.00%
unbalanced	3	4	1.19	46.00%
unbalanced	4	6	1.25	46.00%
unbalanced	5	4	1.21	45.00%

Figure 4.11 below displays a chart of the validation accuracy for rating classification on the unbalanced dataset.

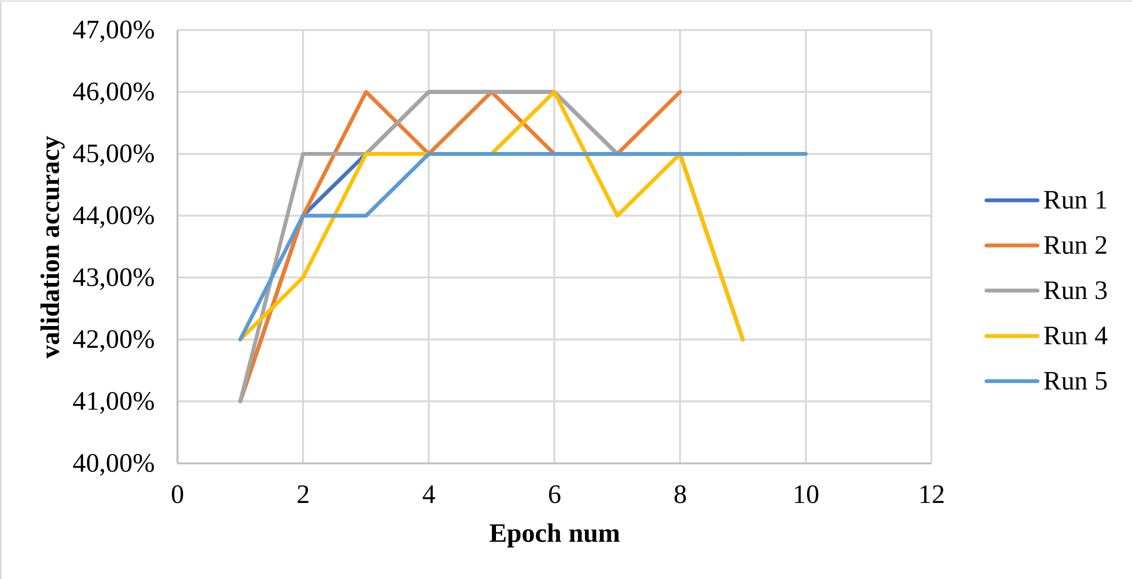

Figure 4.11. Models (Rating Classification Unbalanced Validation Accuracy)

Figure 4.12 below displays a chart of the validation loss for the unbalanced dataset using rating classification. The lowest validation loss was obtained in model 1 and 2, at 1.2 rate.

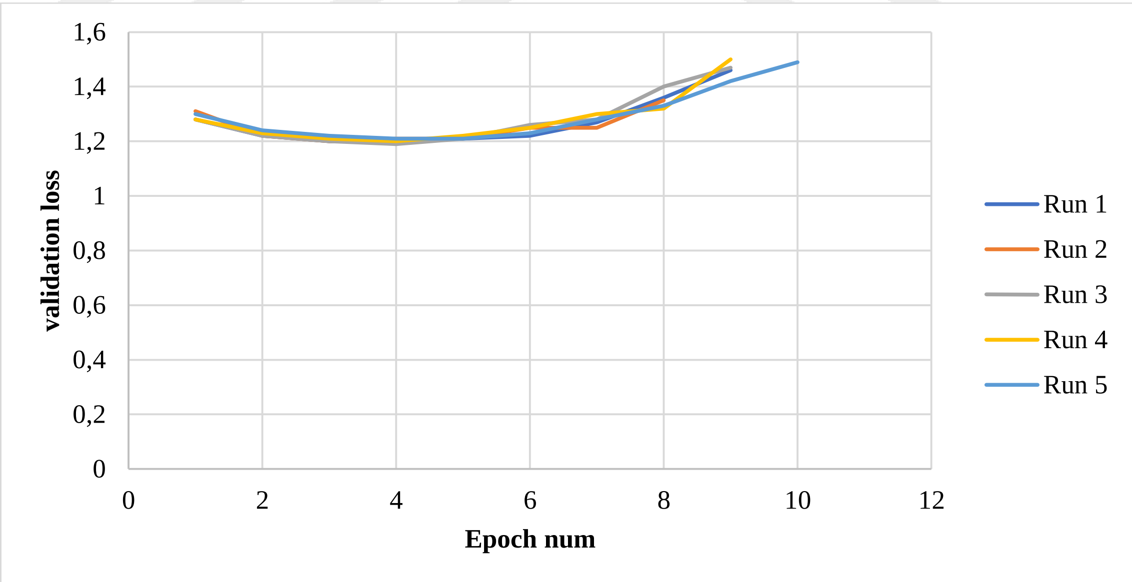

Figure 4.12. Models (Rating Classification Unbalanced Validation Loss)

4.4. Rating Classification Balanced Dataset Results

This section shows the results for rating classification using a balanced dataset through figures and charts. A total of 14,695 book reviews were used as the balanced dataset.

The dataset was divided into train and test data. For the experimentations, the validation accuracy was tested for our models.

4.4.1. Train Accuracy and Train Loss (Rating Balanced Dataset)

Table 4.7 below shows the train accuracy and validation loss for the balanced dataset using rating classification. The highest train accuracy was recorded at 73%, which was found in the 12th epoch of model 4.

Table 4.7. Models Train Accuracy and Train Loss (Rating Balanced)

Dataset	Distribution	Run Num	Epoch Num	Train Loss	Train Accuracy
Rating	balanced	1	11	0.9	61.00%
Rating	balanced	2	12	0.79	66.00%
Rating	balanced	3	12	0.7	71.00%
Rating	balanced	4	12	0.67	73.00%
Rating	balanced	5	12	0.7	71.00%

Figure 4.13 below shows a chart of the train accuracy for rating classification using the balanced dataset.

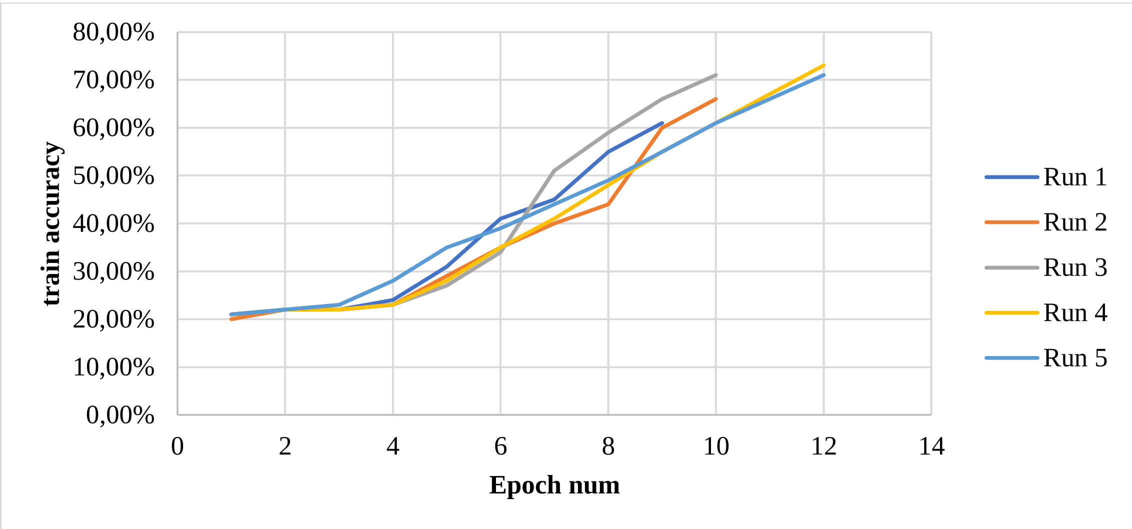

Figure 4.13. Models (Rating Classification Balanced Train Accuracy)

Table 4.7 above also shows the train loss rat for all models. The lowest train loss for rating classification was found at 0.7 rat. This figure was found in both model 3 and 5, both during the 12th epoch.

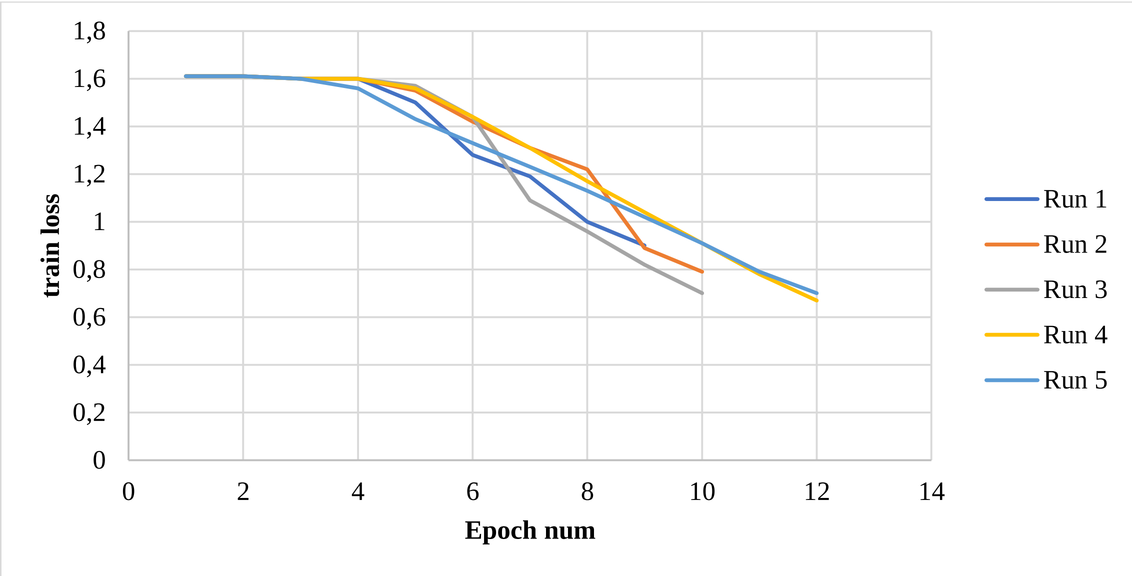

Figure 4.14. Models (Rating Classification Balanced Train Loss)

Figure 4.14 above shows the train loss rat for rating classification using the balanced dataset. From the figure, it can be seen that there is a train loss decrease for all five models.

4.4.2. Validation Accuracy and Validation Loss (Rating Balanced Dataset)

This section shows the validation accuracy and validation loss using rating classification applied to the balanced dataset.

Table 4.8. Models Validation Accuracy and Validation Loss (Rating Balanced)

Dataset	Distribution	Run Num	Epoch Num	Validation Loss	Validation Accuracy
Rating	balanced	1	6	1.4	**39.00%**
Rating	balanced	2	7	1.4	37.00%
Rating	balanced	3	8	1.42	**39.00%**
Rating	balanced	4	9	1.6	36.00%
Rating	balanced	5	8	1.5	37.00%

Table 4.8 above shows the highest validation accuracy for all five models was 37.00%.

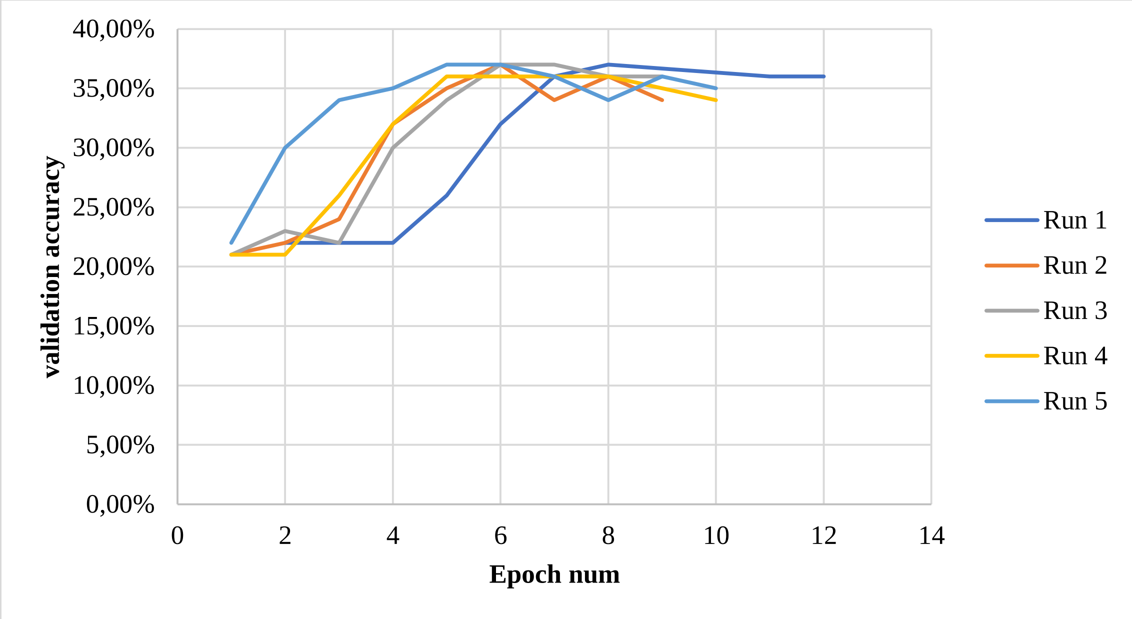

Figure 4.15. Models (Rating Classification Balanced Validation Accuracy)

Figure 4.15 above shows a chart of the validation accuracy using rating classification for the balanced dataset. Table 4.8 above also shows the validation loss, with the lowest being found in model 2 at the 7th epoch.

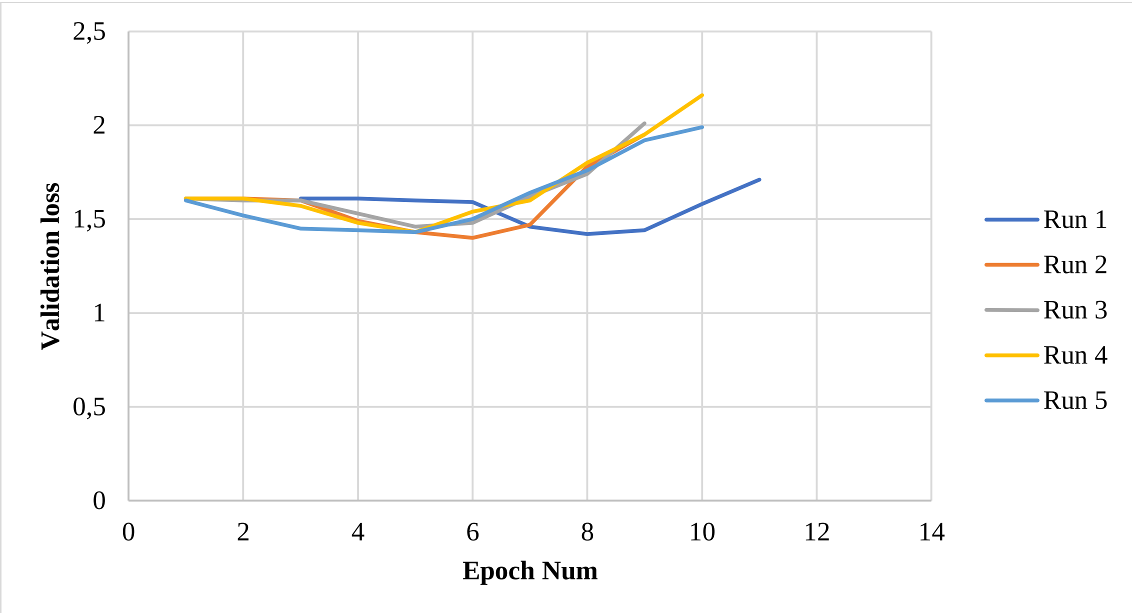

Figure 4.16. Models (Rating Classification Balanced Validation Loss)

Figure 4.16 above shows a chart of validation loss rat using rating classification on the balanced dataset.

4.5. Comparison Between Thesis Experimental Results and LABRs Results

We applied the CNN model under TensorFlow packing onto the LABR dataset in our experiments. In their experiments, the same corpus and same methodology of splitting the dataset was used. They also used a balanced and unbalanced dataset through their experiments. LABRs authors apply two different type of NB algorithm one of them MNB (Multinomial Naive Bayes) and BNB (Bernoulli Naive Bayes). In addition, they applied SVM in their experiments. The information below shows the highest results obtained in our experiments compared to the highest results from other LABRs experiments.

Table 4.9. Validation Accuracy (Polarity Balanced, Unbalanced)

Polarity		
	Avg. validation Accuracy	
Algorithm Name	Balanced	Unbalanced
LABR BNB	80.70%	88.90%
LABR-MNB	80.10%	88.70%
LABR-SVM	76.60%	88.00%
Proposed CNN Model	78.00%	89.00%

As show in the table 4.9 the results for our CNN models for the balanced datasets using polarity classification it's near the LABR's results and it get 78.00%. Also, for the same

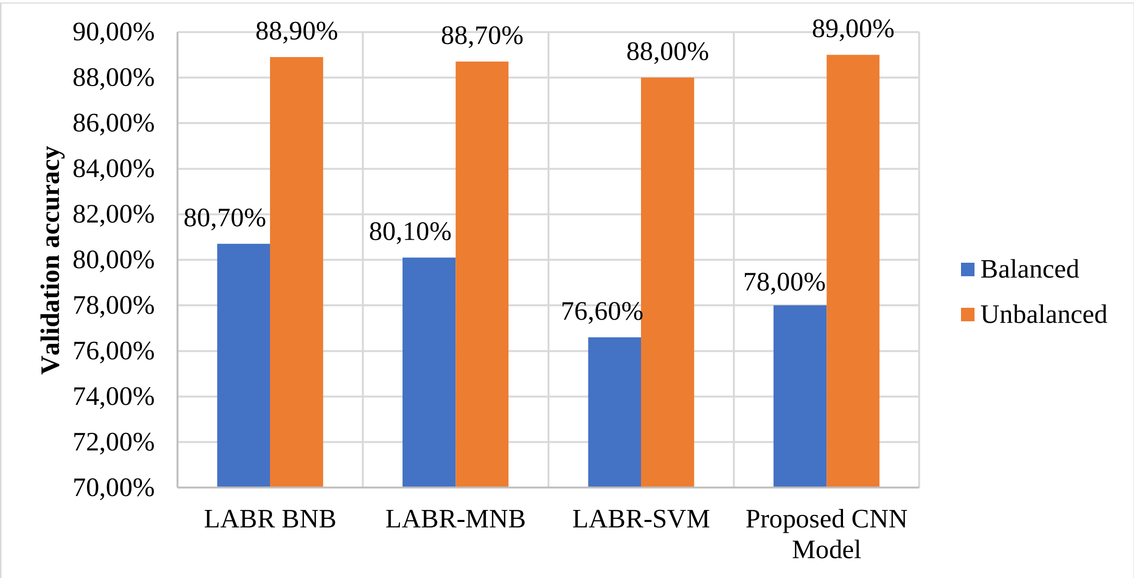

Figure 4.17. Models (Polarity Classification Balanced, Unbalanced)

Classification type but with unbalanced dataset we get 89.00% this result is the highest one from all the models and, higher than the LABR's results but, there is no big different between their results.

Figure 4.17 display the chart of CNN results comparing with the LABR's results using balanced and unbalanced datasets classify by polarity classification.

Table 4.10. Show Validation Accuracy (Rating Balanced, Unbalanced)

Rating		
Algorithm Name	**Avg. validation Accuracy**	
	Balanced	Unbalanced
LABR BNB	39.50%	46.40%
LABR-MNB	39.30%	46.50%
LABR-SVM	36.70%	46.00%
Proposed CNN Model	39.00%	46.00%

Table 4.9 show as the validation results for our CNN models comparing with LABR's results in this table we are display the results for using balanced and unbalanced datasets classify by rating classification.

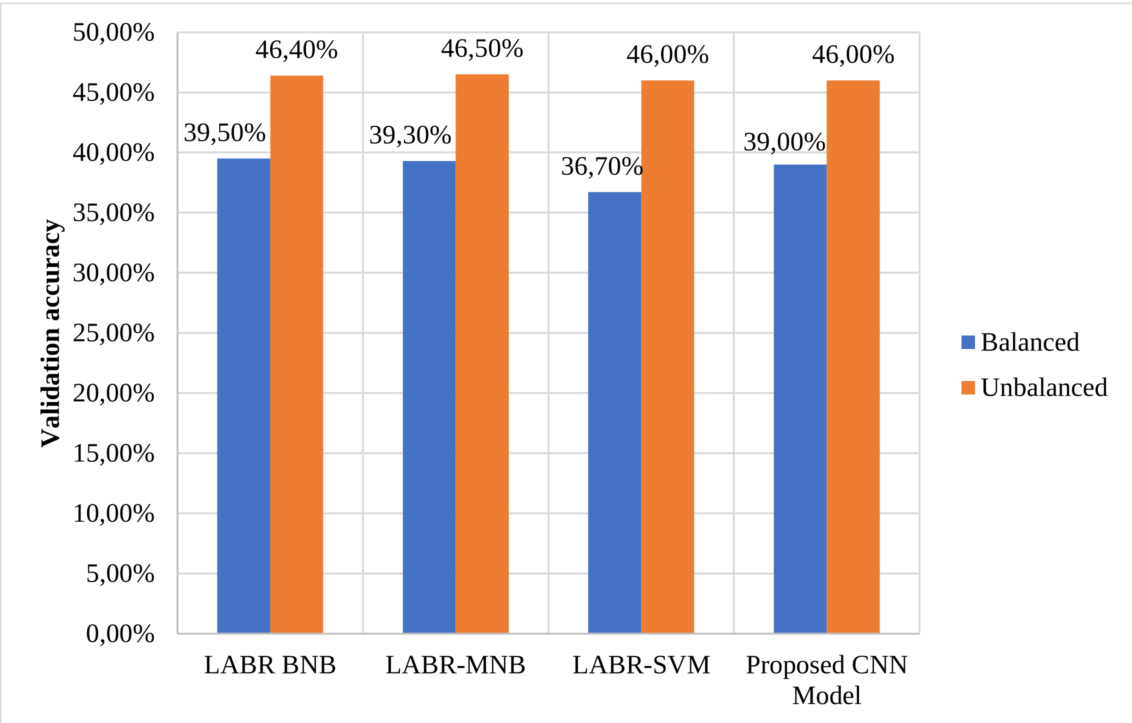

Figure 4.18. Models (Rating Classification Balanced, Unbalanced)

Figure 4.18 display to us the chart of using rating classification with balanced and unbalanced datasets. these result for our CNN models comparing with LABR's. from the chart above we can see our CNN models get 46.00% for the unbalanced datasets. this result very close from the LABR's results, they get also, 46.40%.

For the balanced dataset we get 39.00%, also, close from the LABR's they get 39.50%.

CHAPTER: FIVE

CONCLUSION AND FUTURE WORK

5.1. Conclusion and Future Work

In this chapter, a brief summary of the most important issues in our thesis experiments is outlined, which describes the research work of the master's thesis above. The thesis addressed the problem of analyzing whether deep learning could be applied to analyze sentiment on Arabic text. It was carried out through the implementation of a convolutional neural network because this type of neural network had shown stellar results in previous studies.

Many tasks were presented throughout this research. Five models were implemented and trained on two categories (positive and negative). The models had the same architecture but ran on different parameters.

The first task was how to evaluate text classification methods on an Arabic corpus. As such, the research was based around the LABRs dataset of 63,257 book reviews. We prepared this corpus to prepare it for the application of deep learning methods. We tried many different models on this corpus before settling on the CNN models. After, we had to deal with the encoding issue because the Arabic text had to work with the UTF-8 encoder. We also had to observe most of the sentiment given to the book reviews to detect any sentiment that did not meet the actual opinion.

After the encoding task, we began evaluating the preprocessing steps needed for NLP of Arabic text. For this, we applied different models, beginning with tokenization and the conversion of the content of the Arabic corpus into tokens.

The second preprocessing step was the normalization task that had to deal with different roles of the Arabic text. Once normalization was finished, we had to remove stop words from the corpus. Stemming was the final task in the preprocessing phase. This task deals with the roots of the words in the Arabic text corpus. This means giving all words their root, without exception.

Out experiments were divided into two sections. The first was the polarity classification applied on the Arabic text, using two different datasets: balanced and unbalanced. The evaluations showed that the models that only classify two categories had an average of 89% for the unbalanced dataset and 78% for balanced, both using polarity classification. The second section applied rating classification onto the balanced and unbalanced datasets. For rating classification, the average accuracy for the unbalanced dataset was 46% and 37.8% for balanced.

As previously mentioned, some thoughts were sparked about the worsened performance of the models when classifying categories of the Arabic corpus.

It would be interesting to see if the accuracy improved if more data was gathered, making the datasets more evenly spread among the categories to check if that is the reason of the lower accuracy.

Another idea is to rebuild the models with a completely different architecture as neural networks can be designed in many ways. For instance, you can add multiple dropout layers between each convolutional layer and add more fully connected layers at the end, but also adjust the parameters further. Because it takes a lot of time to get the data and to train the models.

Another aspect is to train the models in more epochs than we did, for instance with 200 epochs. The number of epochs could be doubled or even quadrupled to let the models learn more about the data, but as it takes about six hours per model when 40 epochs were used. Another issue that might occur in the future is new words should be entered into our Arabic corpus.

To keep the models up-to-date, they need to be re-trained on newer data in the future to keep up with the society and the language usage. Moreover, we should try different embedding methods by using (glove vector, word2vec). These two methods could positively impact the accuracy.

Another issue we have to consider is the metrics we could use for our experiments, such as: (tf-idf, n-grams, bi-garam, tri-garam). These metrics can also affect the line of the accuracy.

To conclude, if there was more time, it would be ideal to reach the measurement value and minimum threshold of 85% accuracy on the models that were trained for the two categories, as the other models did.

REFERENCES

1. F.Sebastiani, (1999). “Machine Learning in Automated Text Categorization,” **34** (1), 1–47.
2. L. Khreisat, (2009). “A machine learning approach for Arabic text classification using N -gram frequency statistics,” **3**, 72–77.
3. M. H. Dunham, “Data Mining: Introductory and Advanced Topics. Prentice Hall,” 2003.
4. F. Sebastiani, “Text Categorization,” pp. 1–5, 2005.
5. A. Ng *et al.*, “Deep Learning Tutorial,” *Univ. Stanford*, 2015.
6. K. Kowsari, D. E. Brown, M. Heidarysafa, K. J. Meimandi, M. S. Gerber, and L. E. Barnes, (2017). “HDLTex: Hierarchical Deep Learning for Text Classification,” 364–371.
7. T. Mikolov, G. Corrado, K. Chen, and J. Dean, “Vector Space,” pp. 1–12.
8. M. Aly and A. Atiya, “LABR: A Large Scale Arabic Book Reviews Dataset,” *51st Annu. Meet. Assoc. Comput. Linguist.*, pp. 494–498, 2013.
9. H. M. Harmain, H. El Khatib, and A. Lakas, “Arabic text mining,” pp. 33–38.
10. J. Geraci, P. Wilansky, V. De Luca, A. Roy, J. L. Kennedy, and J. Strauss, “Applying deep neural networks to unstructured text notes in electronic medical records for phenotyping youth depression,” pp. 83–87.
11. Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” 2015.
12. A. Agarwal *et al.*, “TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems,” 2015.
13. TensorFlow, “TensorFlow.” Online.. Available: **https://www.tensorflow.org/.** Accessed: 31-Oct-2018..
14. J. Carlos, D. Santos, and O. Violante, “Sentiment Analysis with Deep Neural Networks,” no. November, 2016.
15. G. Roffo, “Ranking to Learn and Learning to Rank: On the Role of Ranking in Pattern Recognition Applications,” degli Studi di Verona, 2017.

16. J. Deriu, T. Hofmann, M. Jaggi, and A. Lucchi, "Sentiment Analysis using Deep Convolutional Neural Networks with Distant Supervision," 2016.

17. K. O'Shea and R. Nash, "An Introduction to Convolutional Neural Networks," no. November, 2015.

18. Anish Singh Walia, "Activation functions and it's types-Which is better?" Online.. Available: **https://towardsdatascience.com/activation-functions-and-its-types-which-is-better-a9a5310cc8f.** Accessed: 14-Nov-2018..

19. G. Chowdhury, "Natural language processing," **Annu. Rev. Inf. Sci. Technol., 37** (1): 51–89.

20. A. Farghaly and K. Shaalan, (2009). "Arabic natural language processing : Challenges and solutions," **ACM Trans. Asian Lang. Inf. Process., 8** (4): 1–22.

21. S. K. Ray and K. Shaalan, (2016). "A Review and Future Perspectives of Arabic Question Answering Systems," **IEEE Trans. Knowl. Data Eng., 28** (12):3169–3190.

22. T. A. A. Bani-khaled, (2014). "Standard Arabic and Diglossia : A Problem for Language Education in the Arab World," 4(8), 180–189.

23. E. Refaee and V. Rieser, "An Arabic twitter corpus for subjectivity and sentiment analysis," *Proc. Lang. Resour. Eval. Conf.*, no. spring 2013, pp. 2268–2273, 2014.

24. K. Shaalan, "A Survey of Arabic Named Entity Recognition and Classification," no. July 2013, 2013.

25. N. El-makky *et al.*, "Sentiment Analysis of Colloquial Arabic Tweets," *2014 ASE BigData/SocialInformatics/PASSAT/BioMedCom 2014 Conf. Harvard Univ. December 14-16*, no. October 2015, pp. 1–9, 2014.

26. N. a Abdulla, N. a Ahmed, M. a Shehab, and M. Al-ayyoub, (2013). "Arabic Sentiment Analysis," **Jordan Conf. Appl. Electr. Eng. Comput. Technol.,** 6(12):1–6.

27. A. Abbasi, H. Chen, and A. Salem, (2008). "Sentiment analysis in multiple languages: Feature selection for opinion classification in Web forums," **ACM Trans. Inf. Syst. …,** 26 (3):1–34.

28. M. A.-M. Mohammed Korayem1, 3, David Crandall1, "Subjectivity and sentiment analysis of arabic: A survey," *Commun. Comput. I1. M. A.-M. Mohammed Korayem1, 3, David Crandall1, "Subjectivity Sentim. Anal. Arab. A Surv. Commun. Comput. Inf. Sci., vol. 322, no. December, pp. 0–10, 2012.nformation Sci.*, vol. 322, no. December, pp. 0–10, 2012.

29. N. Al-Twairesh, H. Al-Khalifa, and A. Al-Salman, (2014). "Subjectivity and sentiment analysis of Arabic: Trends and challenges," *Proc. IEEE/ACS Int. Conf. Comput. Syst. Appl. AICCSA*, vol. 2014, pp. 148–155.

30. T. alOwisheq, A., alHumoud, S., alTwairesh, N., alBuhairi, "Arabic Sentiment Analysis Resources: A Survey," 9742, no. March 2018, 2016.

31. M. Oakleaf, "OCA: Opinion Corpus for Arabic," **Commun. Inf. Lit., 3** (2): 80–90, 2009.

32. M. Abdul-Mageed and M. Diab, (2012). "AWATIF: A Multi-Genre Corpus for Modern Standard Arabic Subjectivity and Sentiment Analysis.," *Lang. Resour. Eval. Conf. (LREC'12), Istanbul*, November, pp. 3907–3914.

33. M. Nabil, M. Aly, and A. Atiya, (2015). "ASTD: Arabic Sentiment Tweets Dataset," *Proc. 2015 Conf. Empir. Methods Nat. Lang. Process.*, no. September, pp. 2515–2519.

34. H. S. Ibrahim, S. M. Abdou, and M. Gheith, (2015). "MIKA: A tagged corpus for modern standard Arabic and colloquial sentiment analysis," *2015 IEEE 2nd Int. Conf. Recent Trends Inf. Syst. ReTIS 2015-Proc.*, no. May 2016, 353–358.

35. H. Elsahar and S. R. El-beltagy, (2015). "Building Large Arabic Multi-domain Resources for Sentiment Analysis," 3406, 23–34.

36. M. Abdul-Mageed, M. T. Diab, and M. Korayem, (2011). "Subjectivity and Sentiment Analysis of Modern Standard Arabic," *Proc. 49th Annu. Meet. Assoc. Comput. Linguist. Hum. Lang. Technol.*, 27, (1), 587–591.

37. B. Liu and S. M. Mohammad, (2017). “A Practical Guide to Sentiment Analysis,” 5:11–39.

38. Z. Huang, W. Xu, and K. Yu, (2015). “Bidirectional LSTM-CRF Models for Sequence Tagging,”.

39. K. S. Tai, R. Socher, and C. D. Manning, (2015). “Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks,” pp. 1556–1566.

40. J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling,” 2015.

41. A. Kumar *et al.*, “Ask Me Anything: Dynamic Memory Networks for Natural Language Processing,” 2015.

42. M. El Kourdi, A. Bensaid, and T. Rachidi, “Automatic Arabic document categorization based on the Naïve Bayes algorithm,” *Proc. Work. Comput. Approaches to Arab. Script-based Lang. - Semit. ’04*, p. 51, 2004.

43. A. Mourad and K. Darwish, (2013). “Subjectivity and Sentiment Analysis of Modern Standard Arabic and Arabic Microblogs,” **Proc. 4th Work. Comput. Approaches to Subj. Sentim. Soc. Media Anal., 3**, 55–64.

44. A. Shoukry and A. Rafea, (2012). “Sentence-level Arabic sentiment analysis,” **Proc. 2012 Int. Conf. Collab. Technol. Syst. CTS 2012,** 546–550.

45. M. Abdul-Mageed and M. Korayem, “Automatic Identification of Subjectivity in Morphologically Rich Languages: The Case of Arabic.”

46. M. Bautin, L. Vijayarenu, and S. Skiena, (2008). “International Sentiment Analysis for News and Blogs,” *Proc. Int. Conf. Weblogs Soc. Media ICWSM*, pp. 19–26.

47. L. Jiang, M. Yu, M. Zhou, X. Liu, and T. Zhao, (2011). “Target-dependent Twitter Sentiment Classification,” *Comput. Linguist.*, 151–160.

48. M. Abdul-Mageed, M. Diab, and S. Kübler, (2014). “SAMAR: Subjectivity and sentiment analysis for Arabic social media,” **Comput. Speech Lang., 28**(1), 20–37.

49. X. Wang, F. Wei, X. Liu, M. Zhou, and M. Zhang, (2011). “Topic sentiment analysis in twitter: a graph-based hashtag sentiment classification approach,” *Proc. 20th ACM Int. Conf. Inf. Knowl. Manag.*, 1031–1040.

50. M. Elhawary and M. Elfeky, “Mining Arabic business reviews,” *Proc. - IEEE Int. Conf. Data Mining, ICDM*, pp. 1108–1113, 2010.

51. O. A.Ghanem and W. M. Ashour, (2012). “Stemming Effectiveness in Clustering of Arabic Documents,” **Int. J. Comput. Appl., 49** (5): 1–6.

52. R. Duwairi, M. Al-Refai, and N. Khasawneh, (2008). “Stemming versus light stemming as feature selection techniques for arabic text categorization,” *Innov. 4th Int. Conf. Innov. Inf. Technol. IIT*, no. September 2007, pp. 446–450.

53. F. Thabtah, “VSMs with K-Nearest Neighbour to categorise Arabic text data,” *Proc. World Congr. Eng. Comput. Sci.*, no. WCECS 2008, October 22-24, 2008, San Francisco, USA, pp. 22–25, 2008.

54. R. Al-Shalabi, G. Kanaan, and M. H. Gharaibeh, “Arabic Text Categorization Using kNN Algorithm,” *Proc. 4th Int. Multiconference Comput. Sci. Inf. Technol.*, no. February, pp. 5–7, 2006.

55. D. A. Said, N. M. Wanas, N. M. Darwish, and N. H. Hegazy, “A Study of Text Preprocessing Tools for Arabic Text Categorization,” *Int. Conf. Arab. Lang. Resour. Tools*, no. April, pp. 330–336, 2009.

56. H. Sawaf, J. Zaplo, H. Ney, and A. Ag, “Statistical classification methods for Arabic news articles,” *Nat. Lang. Process. ACL2001, Toulouse*, p. 6, 2001.

57. M. M. Syiam, Z. T. Fayed, and M. B. Habib, (2006). “An Intelligent System for {A}rabic Text Classification,” ***Int. {J}ournal {I}ntelligent {C}omputing {I}nformation {S}ciences,*** 6 (1):1–19.

58. A. Mesleh, (2008). “Support Vector Machine Text Classifier for Arabic Articles : Ant Colony Optimization- Based Feature Subset Selection,” ***Arab Acad. Bank. Financ. Sci.,*** pp. 143–148.

59. F. Harrag and E. El-Qawasmah, (2009). "Neural Network for Arabic text classification," **2009 Second Int. Conf. Appl. Digit. Inf. Web Technol.,** 778–783.

60. F. K. Hammouda and A. A. Almarimi, (2010). "Heuristic lemmatization for arabic texts indexation and classification," **J. Comput. Sci., 6**(6), 660–665.

61. J. Neumann, C. Schnörr, and G. Steidl, (2005). "Combined SVM-based feature selection and classification," **Mach. Learn., 61**(1–3): 129–150.

62. J. and A. M. Novoviˇ, (2005). "Information-Theoretic Feature Selection Algorithms for Text Classification," 3272–3277.

63. N. Sanchez-Marono and a Alonso-Betanzos, (2005). "A new wrapper method for feature subset selection," *Symp. Artif.*, 27–29.

64. H. Peng, F. Long, and C. Ding, (2005). "Feature selection based on mutual information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy," **IEEE Trans. Pattern Anal. Mach. Intell., 27**(8):1226–1238.

65. E. Gasca, J. S. Sánchez, and R. Alonso, (2006). "Eliminating redundancy and irrelevance using a new MLP-based feature selection method," **Pattern Recognit., 39**, (2):313–315.

66. S. K. Singhi and H. Liu, (2006). "Feature subset selection bias for classification learning," *Proc. 23rd Int. Conf. Mach. Learn.*, 849–856.

67. Y. Liu and Y. F. Zheng, (2004). "FS SFS : A Novel Feature Selection Method for Support Vector Machines," 1, 797–800.

68. Z. Zhao and H. Liu, (2007). "Searching for interacting features," **IJCAI Int. Jt. Conf. Artif. Intell.,** 1156–1161.

69. G. Vanessa, (2007). "Information-Theoretic Feature Selection for the Classification of Hysteresis Curves," 522–529.

70. D. Zhang, S. Chen, and Z. H. Zhou, (2008). "Constraint Score: A new filter method for feature selection with pairwise constraints," **Pattern Recognit., 41** (5): 1440–1451.

71. H.W. Cho, (2009). “A data mining-based subset selection for enhanced discrimination using iterative elimination of redundancy,” **Expert Syst. Appl., 36** (2): 1355–1361.

72. Y. F. Li, M. Xie, and T. N. Goh, (2009). “A study of mutual information based feature selection for case based reasoning in software cost estimation,” **Expert Syst. Appl., 36,** (2), 5921–5931.

73. M. F. Akay, (2009). “Support vector machines combined with feature selection for breast cancer diagnosis,” **Expert Syst. Appl.,** 36 (2): 3240–3247.

74. M. Baglioni, B. Furletti, and F. Turini, (2005). “DrC4.5: Improving C4.5 by means of prior knowledge,” *Proc. 2005 ACM Symp. Appl. Comput. - SAC ’05*, no. May 2014, 474.

75. J. Li, (2006). “Robust rule-based prediction,” **Knowl. Data Eng. IEEE Trans., 18**(8): 1043–1054.

76. A. Aizawa, “The Feature Quantity: An Information Theoretic Perspective of Tfidf-like Measures,” no. Sigir, pp. 104–111, 2000.

77. M. Rogati, (2002). “High-Performing Feature Selection for Text Classification,” **Proc. Elev. Int. Conf. Inf. Knowl. Manag. - CIKM ’02,** 16:659.

78. P. Frasconi, G. Soda, and A. Vullo, (2002). “Hidden Markov models for text categorization in multi-page documents,” **J. Intell. Inf. Syst., 18**(2–3):195–217.

79. G. Forman, (2003). “An Extensive Empirical Study of Feature Selection Metrics for Text Classification,” **J. Mach. Learn. Res.,** 3:1289–1305.

80. T. Liu, S. Liu, Z. Chen, and W. Ma, (2003). “An evaluation on feature selection for text clustering,” *Icml*, 488–495.

81. D. D. Lewis, Y. Yang, T. G. Rose, and F. Li, (2004). “RCV1: A New Benchmark Collection for Text Categorization Research,” **J. Mach. Learn. Res., (5):** 361–397.

82. S. Bergsma, (2004). “Title Similarity-Based Feature Weighting for Text Categorization,” **Cmput,** 1–22.

83. M. Lan, C.-L. Tan, H.-B. Low, and S.-Y. Sung, (2005). “A comprehensive comparative study on term weighting schemes for text categorization with support vector machines,” **Spec. Interes. tracks posters 14th Int. Conf. World Wide Web - WWW ’05,** 1032.

84. S. Tan, X. Cheng, M. M. Ghanem, B. Wang, and H. Xu, (2005). “A novel refinement approach for text categorization,” *Proc. 14th ACM Int. Conf. Inf. Knowl. Manag. - CIKM ’05*, p. 469.

85. N. M. Wanas, D. A. Said, N. H. Hegazy, and N. M. Darwish, (2006). “A study of local and global thresholding techniques in text categorization,” **Conf. Res. Pract. Inf. Technol. Ser.,** 61:91–101.

86. H. Xu and C. Li, (2007). “A Novel Term Weighting Scheme for Automated Text Categorization,” **Seventh Int. Conf. Intell. Syst. Des. Appl. (ISDA 2007),** 759–764.

87. J. Chen, H. Huang, S. Tian, and Y. Qu, (2009). “Feature selection for text classification with Naïve Bayes,” **Expert Syst. Appl., 36** (3):5432–5435.

88. A. A. Obeidat, (2017). “Arabic text steganography using Unicode of non-joined to right side letters,” **J. Comput. Sci., 13**(6): 184–191.

89. I. A. Alkharashi, “Processing large Arabic text corpora : Preliminary analysis and results Processing Large Arabic Text Corpora :,” July, 2015.

90. C. D. Manning, P. Ragahvan, and H. Schutze, (2009). “An Introduction to Information Retrieval,” **Inf. Retr. Boston.,** 1–18.

91. M. Y. Almusaddar, (2014). “Improving Arabic Light Stemming in Information Retrieval Systems,” **1**.

92. M. M. Syiam, Z. T. Fayed, and M. B. Habib, (2006). “An Intelligent System for Arabic Text Classification,” **Int. {J}ournal {I}ntelligent {C}omputing {I}nformation {S}ciences, 6**(1):1–19.

93. T. M. T. Sembok, B. M. A. Ata, and Z. A. Bakar, (2011). “A rule and template based stemming algorithm for Arabic language,” **Int. J. Math. Model. Methods Appl. Sci., 5**(5):974–981.

94. M. Sawalha and E. S. Atwell, (2008). “Comparative evaluation of Arabic language morphological analysers and stemmers,” ***Int. Conf. Comput. Linguist.,*** 3–6.

95. M. Aljlayl, (2002). “On Arabic Search : Improving the Retrieval Effectiveness via a Light Stemming Approach,” **Proc. Elev. Int. Conf. Inf. Knowl. Manag.,** 340–347.

96. L. Larkey, L. Ballesteros, and M. Connell, (2007). “Light stemming for Arabic information retrieval,” **Arab. Comput. Morphol.,** 221–243.

97. M. Saad, (2010). “The Impact of Text Preprocessing and Term Weighting on Arabic Text Classification,” **Comput. Eng.,** no. August, 172.

98. Ujjwalkarn, (2017). “An Intuitive Explanation of Convolutional Neural Networks,” **Data Sci. Blog,** 1–24.

99. Ji Yang, “ReLU and Softmax Activation Functions · Kulbear/deep-learning-nano-foundation Wiki · GitHub.” Online.. Available: **https://github.com/Kulbear/deep-learning-nano-foundation/wiki/ReLU-and-Softmax-Activation-Functions.** Accessed: 31-Oct-2018..

100. James McCaffrey, “Test Run - L1 and L2 Regularization for Machine Learning,” 2015. Online. Available: **https://msdn.microsoft.com/en-us/magazine/dn904675.aspx.** Accessed: 31-Oct-2018..

.

CURRICULUM VITAE

Name and Suriname	: Bilal Shakir Faraj ALKHASAWE
Nationality	: IRAQI
Birth Date and Place	: 21-12-1991/IRAQI-BAGHDAD
Marital Status	: Single
Mobile	: 05387472152
Email	: bilalshakir59@gmail.com
Correspondence Address	: Kayseri, Talas, Anayurt

EDUCATION

Degree	Institution	Date of Graduation
MSc	Erciyes University	2018
Undergraduate	Madinet AL-Alem University/Baghdad	2013
High school	Al-Jwaheri / Baghdad	2009

Languages

Arabic, English, Turkish