KARADENİZ TECHNICAL UNIVERSITY

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

COMPUTER ENGINEERING

# COMPREHENSIVE TEXT CLASSIFICATION STUDY FOR ONLINE REVIEWS

Mahmoud MURAD

This thesis is accepted to give the degree of
"MASTER OF SCIENCE"
By
The Graduate School of Natural and Applied Sciences at
Karadeniz Technical University

The Date of Submission : 20 / 05 /2019
The Date of Examination : 21 / 06 /2019

Supervisor : Prof. Dr. MURAT EKİNCİ

Trabzon 2019

KARADENİZ TECHNICAL UNIVERSITY

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

COMPUTER ENGINEERING

Mahmoud MURAD

# COMPREHENSIVE TEXT CLASSIFICATION STUDY FOR ONLINE REVIEWS

Has been accepted as a thesis of

MASTER OF SCIENCE

after the Examination by the Jury Assigned by the Administrative Board of the Graduate School of Natural and Applied Sciences with the Decision Number 1806 dated 28 / 05 / 2019

Approved By

Chairman : Prof. Dr. Mustafa ULUTAŞ ..........................................

Member : Prof. Dr. Kemal BICAKCI ..........................................

Member : Prof. Dr. Murat EKİNCİ ..........................................

Prof. Dr. Asim KADIOĞLU

Director of Graduate School

KARADENIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

COMPUTER ENGINEERING

# COMPREHENSIVE TEXT CLASSIFICATION STUDY FOR ONLINE REVIEWS

MASTER THESIS

**Mahmoud MURAD**

JUNE 2019
TRABZON

# DEDICATION

First, all praise and glory are due to Allah the almighty who provided me with the much-needed strength to successfully accomplish this work.

I would first like to thank my thesis advisor Prof. Dr. MURAT EKİNCİ of the Computer Engineer department at Karadeniz Technical University. The door to Prof. Dr. EKİNCİ was always open whenever I ran into trouble spot or had a question about my research or writing. He consistently allowed this thesis to be my own work but steered me in the right direction whenever he thought I needed it.

I would also like to thank my co-workers, R&D director, and general manager in Mahrek Technology for their endless support and valuable comments.

I would like to thank my family, and I wish to express my heartfelt gratitude to all of them for their encouragement, constant prayers, and continued support.

Finally, my appreciation goes to my mother, the strong and gentle soul who taught me to trust in myself, believe in hard work and that so much could be done with little.

To my father, for earning an honest living for us and for supporting and encouraging me to believe in myself.

To my friends, for their suggestions and encouragement.

Mahmoud MURAD
Trabzon 2019

## THESIS ETHICS STATEMENT

I declare that this Master Thesis, I have submitted with the title "COMPREHENSIVE TEXT CLASSIFICATION STUDY FOR ONLINE REVIEWS" has been completed under the guidance of my Master supervisor Prof. Dr. MURAT EKİNCİ. All data used in this master thesis are obtained by simulation and experimental work done as part of this work in our research labs. All referred information used in the thesis has been indicated in the text and cited in a reference list. I have obeyed all research and rules during my research, and I accept all responsibilities if proven otherwise. 21.06.2019

Mahmoud MURAD

# TABLE OF CONTENTS

**Page Number**

Yüksek Lisans Tezi

Yüksek Lisans
ÖZET

# İNTERNET ORTAMINDA KI İNCELEMELER İÇİN KAVRAMSIZ METİN SINIFLANDIRMA ÇALIŞMASI

Mahmoud MURAD
Karadeniz Teknik Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı
Danışman: Prof. Dr. MURAT EKİNCİ

2019, 42 Sayfa

İnternet ortamında ki incelemeler, makaleler ve haberler gibi yapılandırılmamış metinlerin artması, doğru metin sınıflandırmasını kaçınılmaz bir ihtiyaç haline getirmiştir. Bu tür modellerin oluşturulmasında en kritik adımlardan biri de özellik ağırlıklandırmadır. TF-IDF Term Frequency – Inverse Document Frequency (Terim frekansı – ters metin frekansı) bu amaç için kullanılan en popüler yöntem olmasına rağmen, bazı durumlarda sınıflandırma modelinin doğru performans göstermesini engelleyen bazı kısıtlamaları bulunmaktadır. Bu çalışmada iki alanda geliştirme yapılacaktır. Birincisi; konuşma etiketlemenin bir bölümünü kullanan orijinal TFIDF yönteminin kısıtlarını aşan iyileştirilmiş bir versiyon geliştirilecektir. İkinci geliştirme de iki gizli katmanı olan yeni bir derin sinir ağının eğitilmesidir. Çalışmalar 4 adet çok büyük veri seti üzerinde yapılmaktadır ve sonuçlar oldukça umut vericidir.

**Anahtar Kelimeler:** Metin madenciliği, Özellik ağırlıklandırma, Yapay sinir ağı, Konuşmanın bir kısmı, Doğal dil işleme

Master Thesis

## SUMMARY

## COMPREHENSIVE TEXT CLASSIFICATION STUDY FOR ONLINE REVIEWS

Mahmoud MURAD
Karadeniz Technical University
The Graduate School of Natural and Applied Sciences
Computer Engineer Graduate Program
Supervisor: Prof. Dr. MURAT EKİNCİ

2019, 42 Pages

The need for an accurate text classification model is increasing with the increase of the unstructured text such as online reviews, articles, and news. And one of the most critical steps in building such models is feature weighting. While TFIDF is the most popular method used of that purpose, it has some limitations that prevent the classification model from performing accurately in some cases. In this work we have two main contributions, first we are introducing an improvement version of TFIDF method using part of speech tagging that will overcome the limitations of the original TFIDF equation, second, we trained a new deep neural network with two hidden layers. The studies were conducted on four large datasets and the results were promising.

**Keywords** : Text mining, Feature weighting, Neural network, Part of speech, Natural language processing

## LIST OF FIGURES

**<u>Page Number</u>**

# LIST OF ABBREVIATION

| | |
|---|---|
| *AI* | :Artificial Inelegance |
| *CNN* | :Convolutional Neural Networks |
| *MNN* | :Multilayer Neural Network |
| *ML* | :Machine Learning |
| *NLP* | :Natural Language Processing |
| *BOW* | :Bag of Word |
| *POS* | :Part of Speech |
| *SVM* | :Support Vector Machine |
| *TFIDF* | :Frequency-Inverse Document Frequency |

# 1. GENERAL INFORMATION

## 1.1. Introduction

Welcome to the era of big data, where the growth of data exceeds the limit. Nowadays, we can collect data (text, numerical, binary, or multimedia) almost from everything around us such as web, cars, refrigerators, street lights, body sensors, etc. One of the major data types is text. Moreover, the text could be structured, semi-structured and unstructured. Structured text is the text we could store in tables with rows, columns, and keys to identify the relationship between the attributes, however, structured text represents a small percentage of the whole text data, the rest of the text data around us is actually unstructured, which is not organized and cannot be stored directly to databases; a very popular example of unstructured text is online reviews.

The popularity of online reviews has been increasing rapidly in recent years and taking major responsibility in users' decision making prosses. That is why most of the business websites like Amazon, Alibaba, and Yelp provide different reviewing systems for their users, reviewing a business could benefit both the company and the customer, from the company perspective: reviews point out what customer prefer and disfavor, which helps the company to meet the customer's requirements and expectations, and eventually, leads to business profits. From the customer perspective: prior studies [1,2] have identified that online reviews and recommendations could have an impact on customer decisions when purchasing a product, visit a place, or eat food. Therefore, understanding user reviews is a hot research area [3], where researchers try to understand the hidden factors in reviews that motivate the public to interact with reviews (voting) [4,5,6], writing review and rating [7], or what influence users to participate in online communities [8]. Due to the importance and popularity of online reviews, prediction of good-quality reviews could be quite helpful for increasing business profits and users' experience. A good-quality definition could change based on the nature of the reviews, the problem being discussed, and the perspective of the reader. For example, in Yelp for some users, a review with useful votes can be potential for a good-quality review and can affect the users' opinion.

To understand and analysis the unstructured text, we need first to clean and prepare the text by applying some of the preprocessing data mining techniques, then select our most valuable features, and finally, classify the reviews based on our purpose using linear and non-linear machine learning algorithms. There are a variety of purposes for text classification, in this work, we are focusing on three of the most leading topics in text classification, which is, opinion mining semantic classification, rating classification, and document classification. Still, we could say that all those topics are under one big purpose, which is assigned unlabeled text to predefined classes. Moreover, assigning labels or classes could be hard-classification, where every text belongs to one and only one class, or fuzzy-classification, where a text could belong to more than one class, with a different belonging percentage for each class.

In this work, we also consider opinion mining as the prediction of public opinion about a review, and it can be measured through the provided feedback systems by online communities, such as the helpful voting system in Amazon, in which every review related to the number of people founded that review as a helpful, or the voting system of Yelp, which provide three voting options (useful, funny, or cool) for every review, where the Yelp users could choose one or more option to express their opinion about a specific review. Accordingly, public opinion could be considered as a set of feeling, differed from one online reviewing community to another, in Yelp the set of feeling would be useful, funny, or cool because of the voting system provided by the site. Also, semantic classification in this work is the classification of reviews to positive and negative which is a two-class problem. While rating classification is a five-class problem, where we are trying to predict how many stars a user would give to a review, regardless of the context. Finally, in document classification, we are classifying the text to which topic it belongs, this could be considered as an n-class problem, where n the number of predefined topics.

Generally, in any text classification problem, after feature selection, feature weighting algorithm is needed to produce the final feature vector which will be fed to the machine learning algorithms. Feature weighting is a vital step in machine learning tasks that is used to approximate the optimal degree of influence of individual features [9]. Term Frequency Inverse Document Frequency (TFIDF) is the most popular algorithm for feature weighting, however, it has some limitation (to be discuss in chapter 2). In our research we are introducing a modification to TFIDF that will overcome its limitation. Experiments shows promising result.

We will use datasets obtained from different sources, Amazon, Yelp, Yahoo!, and news sources. And for the classification task, we will build multiple models linear and non-linear, each will be a combination of Natural Language Process (NLP) and Machine learning (ML) algorithms. As any classification problem, we started with data cleaning and preprocessing, and in our baseline case, we used two methods from NLP for feature extraction, the Bag-Of-Words (POW) and part-of-speech tagging (POS). For classification, we used three linear ML algorithms, Naive Bayes (NB), linear Support Vector Machine (SVM), and Logistic Regression (LR), and Multilayer Neural Network (MNN) with hidden layers as non-linear algorithm.

## 1.1. Objectives

### 1.1.1. Main Objective

In this research, we aim to conduct a comprehensive text classification study to understand and analyze unstructured text (online reviews and news documents). A prediction model should be able to take single text or text dataset as input and perform different text mining operation on the input to produce a useful result. In general, because online reviews are unstructured text, every review produces various feature vector in term of dimensionality, in other words, every feature vector contains a different number of features, such problem could affect the machine learning performance and lead to a poor result. In this research, we are using hashing to address this problem. Moreover, we are providing a new improvement to the TFIDF algorithm that will help archive more accrued results in our prediction problems, in our improvement we are using POS tagging of the features in the weighing process. Finally, we are building and training a new MNN model.

### 1.1.2. Specific Objective

The specific objectives of the proposed study are:

- Take single text or text dataset as input.
- Perform preprocessing techniques such as stop-word removing, stemming, tokenization on the input data.
- Extract features from the preprocessed input by using NLP methods, then apply a weighting method to convert from text to numerical.
- Applying the specified machine learning algorithms to classify the input based on the desired purpose.

- Assess the performance of the models with different performance metrics.

## 1.2. Importance of Research and Contributions

Due to the large amount of text data around us, the results of our research can be used in many applications. In general, text classification is very helpful in fields such as search engines, recommendations, marketing, document organization, and spam filtering. In specific, our research can be used in the following areas:

- Using semantic classification (positive and negative reviews), we can present user recommendations in e-commerce platforms based on user's old reviews.
- With rating classification, we can organize online reviews to improve user experience while reading reviews.
- Utilizing users' opinion, we can help businesses understand user trends and act on them.

From a technical perspective, this research has the following contributions:

1. By using POS tagging, we present an improvement to TFIDF that can overcome the limitation of the original algorithm. To the best of our knowledge, we are the first to consider POS in the weighing process.
2. Build a new MNN model for text classification problems, the introduced model can be trained on any unstructured text and predict the class of that text with high accuracy.
3. Comparative study on different term weighting schemes are made both by theoretical analyses and by extensive experiments of text classifications on four commonly used benchmark datasets
4. Our investigation in POS tagging and the results we are presenting from our study could open the door for a new field of research in feature weighting.

## 1.3. Scope and Limitations

Text classification is a wide field with a lot of applications, however, in this research, we are focusing on four of the most important topics in that field, semantic classification, rating classification, and document classification with the following limitations:

- In semantic classification, we consider only positive and negative reviews, neutral to be removed from the dataset. Also, we consider the users' opinion to be a set of feeling, in which the classification is hard-classification where every review belongs to one class only.

### 1.4. Research Methodology

Because our requirements are known up-front and the experiments are consisting of multiple sub-experiment each for a specific purpose, we saw that the Incremental Model is the most suitable methodology to be used in our research.

In the incremental model *figure 1*, we incrementally add sub-systems to the system until completing all requirements. For every cycle of the model, every sub-system is traded as a separated system, where it goes through, analyzing, designing, implementing, and testing.

In our work, the study is divided into four main sub-studies based on the topic discussed.

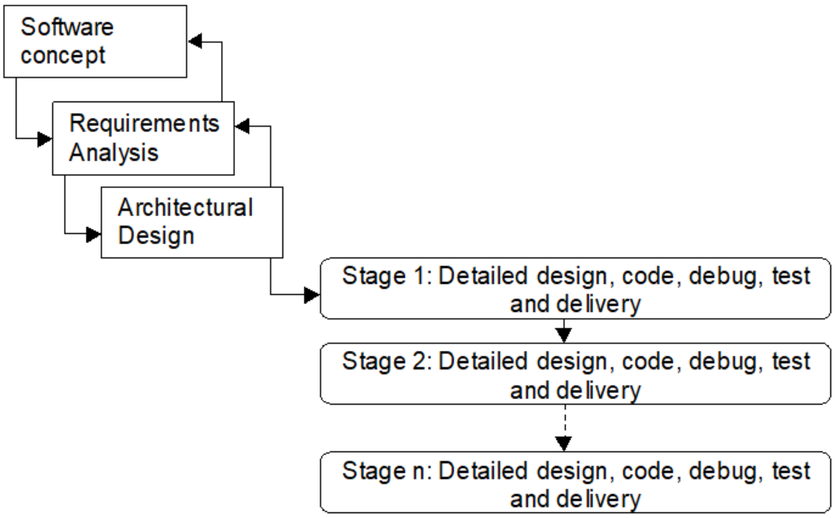
Software concept
Requirements Analysis
Architectural Design
Stage 1: Detailed design, code, debug, test and delivery
Stage 2: Detailed design, code, debug, test and delivery
Stage n: Detailed design, code, debug, test and delivery

Figure 1. Incremental model

## 1.5. Overview of Thesis

The remainder of the thesis contains the following:

Chapter 2: State of the art and related works : This chapter focuses on the current trends and state of art in text mining, and different text classification topics. Then summarize some of the related work.

Chapter 3: Experıments: This chapter explains in detail the conducted experiments, also, the used datasets, technique, and algorithms.

Chapter 4: Results and Dıscussıon: This chapter presents the results of our study with the test dataset, comparing results with baseline, and the used performance metrics.

Chapter 5: Conclusıon: This chapter presents a conclusion of the thesis and a discussion for future works.

# 2. STATE OF THE ART AND RELATED WORKS

In this chapter is divided into two main sections, first, we present the state of the art used, second, we summarize the most related work to our work in this thesis. The first section covers the following topics, Data Mining, Text mining, Semantic Classification, Rating Classification, Document Classification Feature Weighting, Machine Learning, and MNN.

## 2.1. State of the Art

### 2.1.1. Machine Learning

ML is one of the important fields in Artificial Intelligence (AI), it is a set of algorithms and techniques that grants systems the ability to become more intelligent, and to perform tasks that were not programmed to do. The primary goal of ML is to build models that can preforms complex computation based on statistical methods to predict an output from a received input. Furthermore, that process should be iteratively applied, where the model learns from its mistakes by every iteration.

The name ML was first introduced by Arthur Samuel in 1959, and from then ML is widely used in many applications fields, such as recommendations, fraud detection, spam filtering, and network security. Generally, most of the people are familiar with ML in the field of recommendations; particularly in online marketing, where companies like Amazon and Yelp uses ML to predict what users interested in and based on that, they show custom ads and recommendations to their users.

ML algorithms are often categorized as supervised, semi-supervised and unsupervised algorithms. Supervised algorithms trains on a labeled dataset and its main purpose are to predict the label of unseen input; in other words, in supervised algorithms, we train the model to the desired outcome from the input dataset. Supervised algorithms could be used in classification problems where the output is a class label, or in regression problems where the output is a real number. Unlike supervised, unsupervised algorithms do not train on desired output, in the training cycles we provide the model with an unlabeled dataset and based on the background knowledge of the problem space and the understanding of the dataset, unsupervised algorithms such K-means and K-nearest neighbors can classify the input dataset to K cluster. In general, unsupervised algorithms are used in more complex problems such as image recognition. Collecting labeled data considered costlier compared to unlabeled data, therefore, there is a third type of ML algorithms called semi-supervised where the training dataset contains both label and unlabeled samples. MNN is a good example of semi-supervised algorithms, where some hidden layers of the MNN model trained on unlabeled data, and one layer (output) is trained on labeled data.

### 2.1.2. Deep Neural Network

Neural network history goes back to the 1940s, where Warren McCulloch and Walter Pitts [10] wrote a paper on neurons might work based on mathematics and algorithms called threshold logic. Until the late of the 50s neural network wasn't applied to a real-world problem, then Bernard Widrow and Marcian Hoff [11] [12] developed two models called ADALINE and MADALINE. With the evolution of computers and technology, the neural network has become more powerful, and become a hot area of research in Artificial Intelligence (AI). In general, speaking, using multi-layered artificial neural networks can be used in applications such as speech recognition, natural language possessing, object detections and many other applications. Neural network has many similarities and differences with the traditional ML algorithms; for example, ML learns from the training dataset and make decisions based on that dataset, in other words, the output is depending on the dataset. However, in a neural network, the learning is depending more on the algorithm to learn from its own. Moreover, neural network models, in general, does not need human domain-knowledge, unlike the traditional ML models which need a minimum human knowledge of the problem domain.

We can consider the neural network as the simulation of human thinking and problem-solving process. Neural network has a learning process like humans', where both can learn by example and previous knowledge. Because neural network can solve problems traditional ML algorithms cannot, and can achieve a better and more accurate result, has become very popular lately.

### 2.1.3. Convolutional Neural Networks

Convolutional neural networks (CNN) are deep artificial neural networks that are used primarily to classify images (e.g. name what they see), cluster them by similarity and perform object recognition within scenes. The architecture of CNN is different than the regular MNN, wherein MNN the input is being transformed through a sequence of hidden layers, and every layer consist of one or more neurons and the neurons of a layer are fully connected to the layer before, eventually, there is the output layer that represents the predictions. However, CNN has a different architecture (figure 2), the layers in CNN have three dimensions, width, height, and depth; the neurons in the hidden layers are not fully connected to each other, finally, the output layer is a fully connected layer reduced to a single vector of probability scores.

CNN has two main parts, the feature learning, and the classification part. In the first part, the network will perform a series of convolutions and pooling operations during which the features are detected. In the second part, the fully connected layers will serve as a classifier on top of these extracted features and they will assign a probability for the inputted object [26].

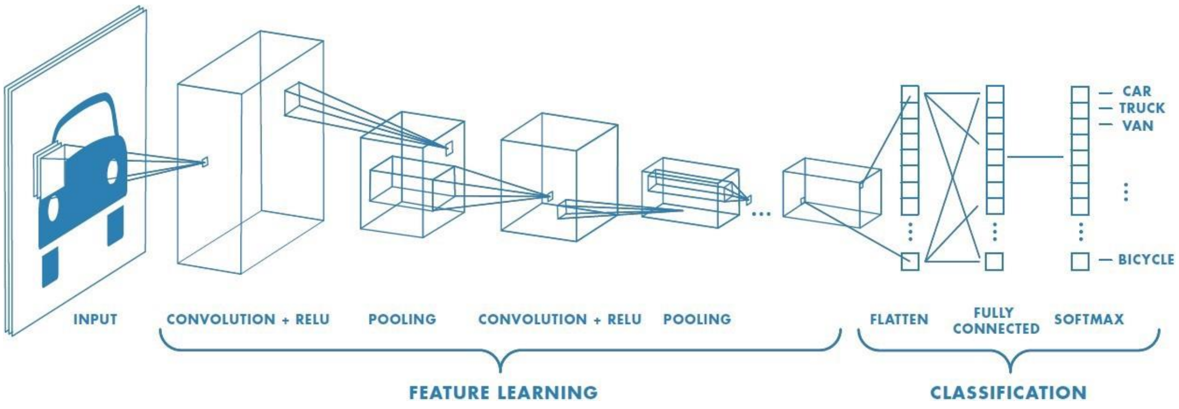

Figure 2. The architecture of a CNN [28]

### 2.1.4. Naive Bayes

NB algorithm [13] is a probabilistic classifier based on the Bayes' theorem with the assumption of independence between the features. Basically, if we have a feature vector

X = [x1,x2,x3,..,xn] and a set of class labels C = [c1,c2,..,cn], to predict the probability of that vector X being belong to class C, we can apply the Bayes' rule (1)

$$P(C \mid X) = P(C)\, P(X \mid C) \,/\, P(X) \tag{1}$$

### 2.1.5. Linear Support Vector Machine

Linear SVM algorithm [14] is a widely used algorithm for classification and regression. SVM built a hyperplane/s to separate the training dataset points, for example, if we have a set of feature vectors [$x_1$,$x_2$,..,$x_n$], we need to find the hyperplane that has the largest distance to the nearest feature vector of any class.

### 2.1.6. Logistic Regression

LR [15] is a popular model used to solve the classification problem, it can be used as a binomial or multinomial model. For the binomial problems, the sigmoid function $f(x)$ =1/1+$e^{-x}$ could be applied to predict the probability of the classes. However, for a multinomial problem (K classes), we can apply the Softmax function (2).

$$f(x) = \frac{e^{x}}{\Sigma_{k=1}^{N} e^{x_k}} \tag{2}$$

### 2.1.7. Data Mining

Data mining is the steps of analyzing datasets to discover patterns that valuable and unknown [16], the goal of data mining is to categorize those patterns into useful information to help in decision making. Data mining is being used in many different fields; for example, many discovering algorithms are being used in marketing related fields, such algorithms could help to discover potential customers and improve provided services which lead to increase the profits.

The process of data mining for different purposes has some common steps;

- Collecting related data to the domain of the problem, the data could be numerical, text, or any type of data.

- Understanding and analyzing the collected data.
- Based on the data analyzing, cleaning the data from any noise and handlining missing information.
- Extracting the most valuable features from the data.
- Build a model using a suitable data mining algorithm.
- Apply the model to unseen data and evaluate the model.

A data mining algorithm is a combination of steps and statistical calculation to produce a model that can discover unknown patterns inside the dataset. Classification and regression algorithms are the most used data mining algorithms; in general, they are used for prediction based on labeled dataset. Furthermore, clustering algorithms are another type of data mining algorithm used for identifying relationships among attributes in order to group the dataset to k-number of clusters.

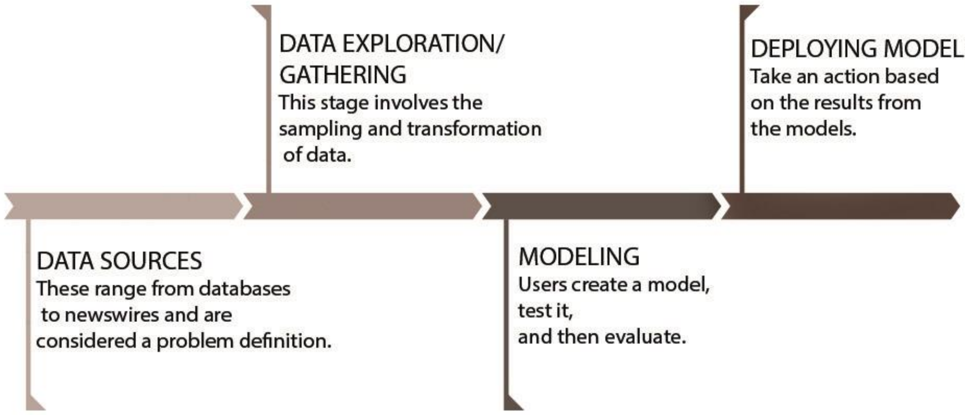
DATA EXPLORATION/ GATHERING
This stage involves the sampling and transformation of data.
DEPLOYING MODELS
Take an action based on the results from the models.
DATA SOURCES
These range from databases to newswires and are considered a problem definition.
MODELING
Users create a model, test it, and then evaluate.

Figure 3. Data mining stages

### 2.1.8. Text Mining

Text mining is a sub-field of data mining, whereas data mining is discovering useful patterns from datasets [17], text mining is discovering useful and unknown patterns from textual datasets. In general, text mining refers to extracting non-trivial patterns from the unstructured text; such as news article and online reviews.

As mentioned before, text mining is considered very important in the business field. However, working with text and especially unstructured text is more complex and difficult than working with the numerical dataset for example. Unstructured text is fuzzy and contained a lot of noise, therefore it requires more prepossessing than the numerical data.

Text mining intersects with Information Retrieval, NLP, Data Mining, ML and Information Extraction, therefore, to build a successful model, knowledge from those fields is required. Usually, using NLP methods we can perform the first step in building a text mining model which is reorganization of the text in a way that we can apply quantitative and qualitative analysis on the text, NLP methods are a combination of statistical computations and language's set of rules and their main goal is to make the humane language (text in our case) understandable by machines.

### 2.1.9. Semantic Classification

Semantic classification [18] or opinion mining is one of the important areas of research in text mining, and is very popular in advertising and recommendation systems; semantic classification is aiming to predict user's feeling, in other words, what users like, dislike, love, hate, prefer and need, once the classification model can predict those feeling, the system can suggest suitable ads to users. Furthermore, semantic classification models trained on keyword or phrases using machine learning algorithms to predict and categorize the text contents. In such approach, users will see ads they are interesting in, which will improve their experience with the system, leading the company providing those ads to increase their profits.

In traditional text classification, a document is present as a vector of features extracted by methods such as bag-of-word, in which, every word in the document is a feature regardless to the context or the positions of the word in the document. However, semantic classification of text is taking into consideration the positions of the word and the part-of-speech tagging of the word, so that not every word in document present as a feature.

### 2.1.10. Document Classification

To assign a document to one or more predefined labels [19], we have two methods: manually assign the document to label by reading and interpreting the text, then based on human judgment, a document assigned to one or more label; however, such prosses comes with high costs and time wasting. The second method is automatically assigning the labels to document which is document classification.

Document classification is another field of text mining and more focusing on documents such as news articles, blogs, and online reviews. While semantic classification focusing on the word and its context and position of the text, document classification is a content-based classification aiming to assign one or multiple predefined labels to the whole document making the set of documents more manageable for tasks such as searching, reporting, sorting, and filtering. Financial and insurance, writer and publishers, and any industry which operates on unstructured text can find document classification very useful because a classification system can improve the speed and performance of document processing in those industries. Document classification is originally used in information retrieval where using it helped to understand and manage large volumes of unstructured text; then it combined with ML and NLP to become a part of text mining and data mining.

To automatically classifying documents there are three methodologies: Supervised classification, unsupervised classification (mentioned in section 2.1.1), and Rule-based classification, unlike the first two methodologies, Rule-based classification does not rely on statistics, it builds relationships between texts and labels and tries to think more like humans by associate roles to results. This method considered more accurate and has high performance in term of predictions.

### 2.1.11. Feature Selection

Feature Selection or attribute selection is a middle and primary step in the text mining prosses, it’s the automatic selection of a subset of dataset’s features, this selection is very important in cases when the number of features is very large because in general, it will contain a lot of domain-unrelated features that will mislead the classification model. Feature selection has many benefits for classification models, first it reduces the complexity of the model, second it makes the training for the model faster, third it improves the performance of the model, finally, it can be a solution for overfitting.

There are three main categories of feature selection methods: Filter methods, wrapper methods, and embedded methods. Filter methods are statistical methods that used as a preprocessing step and it considers independent from the classification algorithms. Chi-Square, LDA, and ANOVA are an example of filter methods. Wrapper methods search for the best subset of the feature, it trains a model on a subset of the features, then based on the results more features are added or removed. Such methods are considered very expensive and time-consuming. Forward selection, backward selection, and recursive elimination are an example of wrapper methods. Embedded methods are a combination of both filter and wrapper methods, LASSO, RIDGE regression, and decision tree, are an example of embedded feature selection methods.

For text mining in specific there are some especial feature selection methods, Bag-of-Word and Part-of-Speech are two of the most popular text feature selection.

### 2.1.12. Bag-of-Word

As mentioned before, in general, ML algorithms takes numerical input for training, and in text classification cases we need more preprocessing steps to represent text as a numerical vector. One of the most popular techniques in feature preparing for text mining is BOW; it is a very simple to understand and implement, also, very efficient way of representing text as features.

In BOW, we tokenize every text (document, online review, article) and consider each token as a feature excluding stop-words like (in, a, the, to, on, etc.). a document is represented as a bag of its words, moreover, the structure and order of the words are neglected which make his approach simple and flexible. BOW proved its efficiency in many text classification problems.

However, BOW cannot capture the relationships between words, which in some cases can improve the accuracy of prediction or classification model. Therefore, an improvement of BOW called n-gram can be used. N-gram is a sequence of n tokens for some integer n. For example, consider the following sentence:

“*In general speaking, Deep learning uses multi-layered artificial neural networks*”

If *n* equals 1, this is typically the BOW where every word in the sentence is a token, Moreover, if *n* equals 2, the output tokens would be: “In general”, “general speaking”,” speaking Deep”, “Deep learning”... etc. obviously, one token like “Deep learning” in a 2-gram model is more accurate than two tokens “Deep” and “learning” in a BOW model. However, the n-gram is more expensive than BOW in term of memory and feature size.

### 2.1.13. Part-of-Speech

In any language, a word is categorized into one of several classes in the smallest level of the language, those classes are the POS tags of the language. In the English language, there are seven main tags which are noun, pronoun, verb, adverb, adjective, conjunction, preposition, and interjection as the following:

- Noun (N)- Daniel, London, table, dog, teacher, pen, city, happiness, hope
- Verb (V)- go, speak, run, eat, play, live, walk, have, like, are, is
- Adjective (ADJ)- big, happy, green, young, fun, crazy, three
- Adverb (ADV)- slowly, quietly, very, always, never, too, well, tomorrow
- Preposition (P)- at, on, in, from, with, near, between, about, under
- Conjunction (CON)- and, or, but, because, so, yet, unless, since, if
- Pronoun (PRO)- I, you, we, they, he, she, it, me, us, them, him, her, this
- Interjection (INT)- Ouch! Wow! Great! Help! Oh! Hey! Hi!

PoS tagging is a technique in NLP, and it can be adapted in text mining problems as a feature selection method. For example, for some cases, just the Noun can be selected as features and discard the rest of words in the document. Such a method needs a very good analysis of the domain of the problem and the dataset.

### 2.1.14. Feature Weighting

Feature weighting is a technique used to approximate the optimal degree of influence of individual features using a training set. When successfully applied relevant features are attributed to a high weight value, whereas irrelevant features are given a weight value close to zero. Feature weighting can be used not only to improve classification accuracy but also to discard features with weights below a certain threshold value and thereby increase the resource efficiency of the classifier.

Most of the machine learning algorithms take numeric vectors as input, in cases like text classification (the input feature vector is a vector of strings), we must convert the text to numeric vector, the most popular algorithm for the task is TFIDF.

### 2.1.14.1.Term Frequency-Invers Documents Frequency

TFIDF is a weighting algorithm often used in information retrieval and text mining. This weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus.

Typically, the TFIDF weight equation (3) is composed of two terms: the first computes the normalized Term Frequency (TF), aka. the number of times a word appears in a document, divided by the total number of words in that document; the second term is the Inverse Document Frequency (IDF), computed as the logarithm of the number of the documents in the corpus divided by the number of documents where the specific term appears.

$$\text{TF*IDF} = \text{TF(t,d)} * \log \frac{|D|}{DF(t,D)} \qquad (3)$$

where,

- TF: Term Frequency**,** which measures how frequently a term occurs in a document. Since every document is different in length, it is possible that a term would appear much more times in long documents than shorter ones. Thus, the term frequency is often divided by the document length as a way of normalization: TF(t,d) = (Number of times term **t** appears in document d) / (Total number of terms in document d).
- IDF: Inverse Document Frequency**,** which measures how important a term is. While computing TF, all terms are considered equally important. However, it is known that certain terms, such as "is", "of", and "that", may appear a lot of times but have little importance. Thus we need to weigh down the frequent terms while scaling up the rare ones, by computing the following: IDF(t,D) = log(Total number of documents / Number of documents with term t in it), where the Logarithms is base 10 (common log).

Consider a document containing 100 words wherein the word *cat* appears 3 times. The term frequency (i.e., tf) for *cat* is then (3 / 100) = 0.03. Now, assume we have 10 million documents and the word cat appears in 1000 documents. Then, the inverse document frequency (i.e., idf) is calculated as log(10,000,000 / 1000) = 4. Thus, the TF-IDF weight is the product of these quantities: 0.03 * 4 = 0.12 [20].

#### 2.1.14.2.Limitation

The main problem with TFIDF is that it only considers the relationship between the term and the whole text but neglects the relationship between different terms in different classes. Therefore, it cannot understand that if a term appears in Class A more than in Class C, so that term should not have the same weight in both classes. And IDF is based on the BOW model, therefore it does not capture position in text, semantics, and co-occurrences in different documents.

## 2.2. Related Works

As discussed in the previous section, TFIDF has some limitation. Therefore, over the years, many improvements have been presented.

Kuang and Xu [21] in their research, they add a new weight $C_i$ that consider the frequency of a term in a specific category to the original equation of TF-IDF. In other words, the original TF-IDF considers the frequency of a term in the whole dataset, but TF-IDF*$C_i$ also consider in which category this term appears in. If $N$ is the total number of documents, $n$ is the number of documents containing term $t$ in the whole dataset, and $m$ is the number of documents containing term $t$ in a specific category. Then $C_i$ = 1/(n-m+1) could be used to give a better weighting to a term based on its category. The final equation would be (4):

$$\begin{aligned} TF \bullet IDF \bullet C_i &= TF \times IDF \times C_i \\ &= TF \times \log \frac{N}{n} \times 1/(n-m+1) \end{aligned} \tag{4}$$

When the number of the documents in a category containing term $t$ is large, the number of the documents in the other categories containing term $t$, i.e.（n-m）, is small. Then the term can represent the category of the documents containing the largest number of term $t$, so the weighting value should be large, i.e. the feature representation ability of term t is inversely proportional to the number of the documents in all categories except the category containing the largest number of term *t*.

Wang and Zhang [22] based on Information Theory they presented another improvement for TF-IDF. To solve the limitation of the original equation, they study the information distribution in information theory and applied it to TF-IDF, in specific they study the distribution among classes DI$_{\text{DA}}$ and the distribution inside one class DI$_{\text{DC.}}$ Where in DI$_{\text{DA}}$ represent the distribution of term *t* in *A*ll classes, and DI$_{\text{DC}}$ the distribution of term t in a specific class *C*. They add the following values (5), (6):

$$DI_{DC}(t_k) = \frac{\sqrt{\sum_{j=1}^{n}(tf_j(t_k) - \overline{tf'(t_k)})^2 \Big/ (n-1)}}{\overline{tf'(t_k)}} \tag{5}$$

where, $tf_i(t_k)$ represent the frequency of feature $t_k$ in category *i*, $\overline{tf(tk)}$ is the average frequency of feature $t_k$ in all categories, and *m* is the number of categories.

$$DI_{DA}(t_k) = \frac{\sqrt{\sum_{i=1}^{m}(tf_i(t_k) - \overline{tf(t_k)})^2 \Big/ (m-1)}}{\overline{tf(t_k)}} \tag{6}$$

where, $tf_i(t_{jk})$ represent the frequency of feature $t_k$ in document *j*, $\overline{tf'(tk)}$ is the average frequency of feature $t_k$ in all documents, and *n* is the number of documents inside a class.

According to the first equation, if a term appears in just in one category, it has strong classification ability and DI$_{\text{DA}}$ = 1, if the term appears in all the categories in the same frequency, that's mean it has no classification ability and DI$_{\text{DA}}$ = 0. Moreover, when a term appears in one document inside the class, the distribution of the term inside the class DI$_{\text{DC}}$ = 1, and that means the term is nose and has no classification ability, and if the term appears in all the documents in the same class, that means the term has classification ability for that class, and DI$_{\text{CD}}$ = 0. The final formula (7) will be a combination between the classic TF-IDF and the distribution information:

$$\omega(t_{i,k}) = f \times \min(DI_{DA}, 1 - DI_{DC}) \tag{7}$$

Another improvement was proposed by Chen et al., [23] they adapt the *Gravity Moment GM* from physics to the original equation. According to [14], the weight of a term should be determined mainly by its class distinguishing power, which is embodied primarily by its uneven distribution across different classes. First, sort all the frequencies of term t in all classes C in descending order. The resulting sorted list is $f_{k1} \geq f_{k2} \geq \ldots \geq f_{km}$ where $f_{kr}$ ($r = 1, 2, \ldots, m$ ) is the frequency of $t_k$ occurring in the r-th class.

Lan et al. [24] proposed a new simple supervised term weighting method, i.e., *tf:rf*, to improve the terms' discriminating power for text categorization task, they improve the original TF-IDF by adding the *rf* term as follows (8):

$$icf(t_i) = \log(\frac{|C|}{cf(t_i)}) \tag{8}$$

where, *a* is the number of documents in the category *c* that contain term *t*. and *c* is the number of documents in the rest of the categories that contain this term.

Deqing and Zhang [25] follow on the work of Lan et al. and introduced a new equation (9),

$$icf-based(t_i, d_j) = tf(t_i, d_j) \times \log(2 + \frac{a}{\max(1,|c)} \times \frac{|C|}{cf(t_i)}) \tag{9}$$

where, *cf(t*$_i$*)* is the number of categories in which term *t* occurs, *|C|* denotes the total number of categories in the training corpus. And then they combine their equation with *Lan*'s to have the following equation (10):

$$rf = log\left(2 + \frac{a}{max(1,c)}\right) \tag{10}$$

In the other hand, Zhang et al. [27] built CNN for text classification at character-level, they constructed large datasets from different data sources, and use those datasets to compare the results from CNN, traditional TFIDF, and other state-of-art algorithms. Figure 4 illustrates the CNN model [27] built, which is a 9 layers deep with 6 convolutional layers and 3 fully-connected layers.

Based on the results they had, CNN in character-level could superior the traditional TFIDF in some cases. However, CNN is originally used in the computer vision field, and consider to be very powerful in image classification, using it in text classification in some cases might be useful, but it needs a large dataset to be trained and considered very costly compared to MNN, which is simpler than CNN and could fit to text classification problem more than CNN.

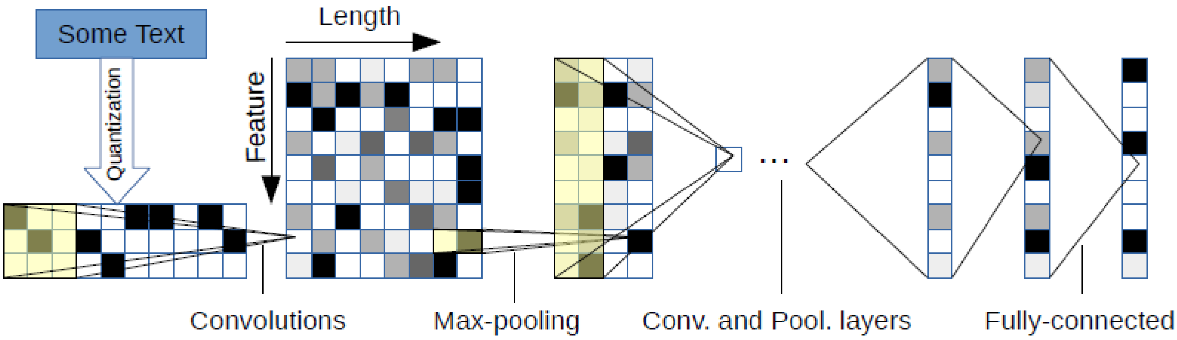

Figure 4. Zhang’s [27] CNN model architecture

## 3. DEVELOPED ARCHITECTURE

### 3.1. Datasets

For the purpose of this experiments, we use four large datasets [27] that originally collected for text classification experiments, the datasets in general, have two main columns, text and label, were the text could be an online review or an arbitrary document, and the label can be multinomial in some datasets and binomial in other. The datasets details as follows:

1. Amazon_review: Amazon is well-known e-commerce and cloud computing company, the dataset contains a review of products from Amazon website, every sample contains three columns, review rate from 1 to 5 which is the label, review title, and review text, in our experiments, we ignored the review title. Moreover, the dataset contains 3.65 million samples divided into two datasets, train dataset 3 million samples, 600,000 samples for every label, and test dataset contains 650,000 samples, 130,000 for every label. This dataset will be used for rate classification purposes in which we have a 5-class problem. We also use the dataset as a 2-class problem where the samples with rate 1 and 2 considered as negative, and the samples with 4 and 5 as positive, samples with rate 3 were ignored, this version of the dataset is used for the semantic classification experiments.
2. Yelp_review: Yelp is a search service powered by a crowd-sourced review, its dataset has the same purpose of Amazon’s and the same structure, However, this dataset is smaller than Amazon’s, in total it contains 650,000 training samples, 130,000 for every label, and 50,000 testing samples, 10,000 for every label. Furthermore, this dataset will be used first as a 5-class problem, then the same as Amazon’s it will be used for 2-class classification.
3. Yahoo_answer: the purpose of the first two datasets was rate classification and semantic classification. However, this dataset used for document classification. Yahoo_answer dataset contains questions and their answers for ten categories collected from Yahoo, each sample has the four parameters, label, question title, question content and best answer. For the aim of this research we neglect the question title and question content and focus on the answer. In other words, we

have ten categories, for every category we have 140,000 training samples and 6,000 testing samples, in total, 1,400,000 training samples and 60,000 testing samples, for every sample we have the label (category) and the best answer. The categories are *Society & Culture, Science & Mathematics, Health, Education & Reference, Computers & Internet, Sports, Business, Finance, Entertainment & Music, Family & Relationships*, and *Politics & Government*.

4. AG_news: is a collection of news articles gathered from more than 2000 news source, it contains news from 4 categories. In total, we have 120,000 training samples and 7,600 testing, 30,000 training samples and 1,900 testing samples for each category. Moreover, every sample has three columns, label (category), title and description, we neglected the title and focus on the description. The categories as follows, *World, Sports, Business, and Sci/Tech*. This dataset used for document classification as the same as Yahoo'.

### 3.2. Experiments Description

In our experiments, we focus on how to improve the accuracy of classification models, every experiment is consisting of three main steps as follows:

1. Preprocessing: preparing the data for classification is a very important step in every text maiming problem [12]. Many preprocessing methods could be used, in our experiments we used stop-words removing [16] stop-words is a commonly used word such as the, a, an, and in, those words generally do not have any effect on the prediction model because it can occur in every sample of the training set across different classes, also, we used tokenization which is breaking up a sequence of strings into pieces such as words, keywords, phrases, symbols and other elements called tokens, in our experiments we use Bag-of-Word and Part-of-speech as tokenization methods.
2. Feature weighting: for every sample in the dataset, the output from the first step would be a vector of strings represents the input text. However, ML algorithms understand numerical vectors as discussed, therefore, in our second step we take the vector of strings and convert it to a vector of numbers. For that purpose, generally TFIDF is

used, in our experiments, besides TFIDF we are also using an improvement version of TFIDF we develop for feature weighting.

3. Prediction: after having the vector of numbers -feature vector- the final step in our experiments is to predict the right label of that vector. To predict, many ML algorithms are available, some are linear and other non-linear. NB, linear SVM, and LR are an example of linear algorithms that we used, also we trained a new DL model for the purpose of these experiments.

### 3.3. Improvement of TF-IDF

As mentioned before, TF-IDF is a well-known weighting algorithm, used to weight text features and extract the numerical representation of that text features for machine learning purposes. However, it has some limitations as explained in chapter two. Many improvements were introduced to overcome their limitations. Nevertheless, none of them consider the word itself, most of them use statistical methods to improve the original equation. In our improvement, we are looking at the problem in a new and different perspective. In our approach, we are adopting the Natural Language Processing (NLP) field to the problem, in specific, we are using the Part-of-Speech tagging of a word to improve the original TF-IDF.

A word can have one Part-of-Speech tag (noun, verb, adjective, ..) which is valuable knowledge in the process of term weighting, and by combine it with original equation, we could have better and more accurate weighting which will led to better classification performance. Let us consider the following case: If we have a dataset contains documents from different topics (Science, Sports, Art, .. ), naturally the documents from Sport class will contains a lot of verbs (play - do – go – run – catch – dribble …) that could be an indicator that verbs in this class affect the prediction model more than other POS tags, therefore, verbs should have higher weight than other. Based on this assumption, we introduce a new factor to the original equation, our factor will cover the limitation of TF-IDF and build a more accurate prediction model.

#### 3.3.1. New factor for feature weighting

Let suppose we have a dataset *D* contains documents $d_j$, every document contains terms $t_i$, for a specific Part-of-Speech tag *p* (verb, noun, adjective, adverb, .. ) :

First we compute *PF(p,d)* the *f*requency of that tag *p* in document *d*.

Second we compute *PC(p,c)* the frequency of *p* in a class *c* in the dataset.

Then the weighting equation (11) for a term $t_i$ in a document $d_j$ in class *C* would be:

$$W(t_{ic},d_{jc}) = f * (PF * log \frac{PC}{|C|} + 1 ) \qquad (11)$$

where *f* is the original equation of TF-IDF and |C| is the total number of terms in class *C*, another representation for the equation would be:
$W(t_{ic},d_{jc})$ = TF.IDF * PF.IPC where,
TF.IDF is the original equation and IPC is the inverse of the frequency of the Part-of-speech *p* in class *c*. Finally, the complete equation (12) for feature weighting is:

$$W(t_{ic},d_{jc})= \left(TF * log \frac{|D|}{DF(t,D)} + 1\right) * (PF * log \frac{PC}{|C|} + 1) \qquad (12)$$

## 3.4.New Multilayer Neural Network Model

MNN is used in many applications with different purposes, text classification is one of them. For our experiments we built and trained a new MNN model, the MNN model we build is based on a multi-layer feedforward artificial neural network that is trained with stochastic gradient descent using back-propagation. Moreover, the length of the input feature vector is variable due to the length of the input text, therefore, a hash function was used as a layer between the text and the MNN input layer, the hash function will hash the variable feature vector into a fixed-size feature vector and provide it as input for the model. MNN has many parameters and to find the best model for a problem, those parameters need to be adjusted in order to have a model that can be appropriate for the problem. The new model (figure 5) we built has more than 40 parameters, the most important are the following:

1. Number of layer and node: the model consist from 4 layers, input layer, two hidden, and the output layer. Furthermore, we have 200 nodes in each hidden layer and one output node in the output layer.
2. Activation Function: for hidden layers we used Rectifier activation function, and because we build a classification model, for the output layer we used Softmax.

3. Epochs: one Epoch is when an entire dataset is passed forward and backward through the neural network only once, our model passed over the dataset 100 time.
4. Minimum batch size: the whole dataset cannot be passed to a neural network at once, therefore, datasets are divided into number of batches, the minimum number of training example in every batches in our model equals 1.
5. Train samples per iteration: Iterations is the number of batches needed to complete one epoch, for every iteration we set the training sample to be random.
6. L2: adding L2 regularization to improve generalization and stability of the model, we set L2 to 0.005.
7. Learning rate: adjusting the weights of the network with respect the loss gradient, our learning rate equals 0.001.
8. Stopping round: stops training when the option selected for stopping metric does not improve for the specified number of training rounds, we set this parameter to 5.
9. Stopping metric: we use *logloss* as stopping metric for early stopping.

The determinations of the value of those parameters was done by conducting multiple tests on the model. For example, after we built three networks, the first with one hidden layer, second with two hidden layers, and the third with three hidden layers; the results show that a model with one layer can be trained fast, but with low accuracy, and a model with three layers becomes too complex and very slow in training with no

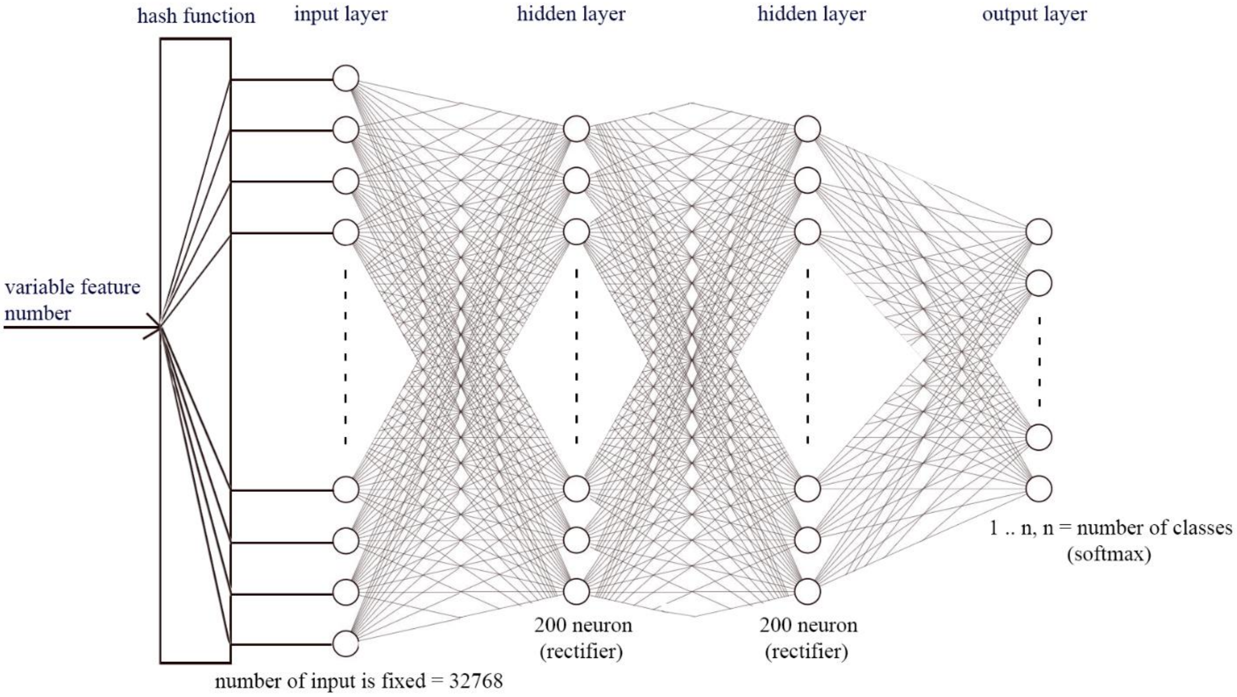

Figure 5. The new MNN model architecture

improvement in the accuracy, so we decided that two hidden layers were the optimal number of layers for the model because with two hidden layers the model trained in an acceptable time with high accuracy. The same goes for the number of neurons in every layer.

### 3.5.Constructed Models

Several models have been built with different combinations, we conduct many experiments based on the previous introduced improvement of TF-IDF and the new MNN model.  As descripted in section 3.2, every model consists from three steps, the following table explain in detail the experiments:

Table 1. Experiments details

| **id** | **model** | | |
|---|---|---|---|
| | *Prepressing* | *Feature weighting* | *Prediction* |
| **1** | Bag of Word | TFIDF | NB |
| **2** | Bag of Word | TFIDF | LR |
| **3** | Bag of Word | TFIDF | Linear SVM |
| **4** | Bag of Word | TFIDF | MNN |
| **5** | Part of Speech | Improved TFIDF | NB |
| **6** | Part of Speech | Improved TFIDF | LR |
| **7** | Part of Speech | Improved TFIDF | Linear SVM |
| **8** | Part of Speech | Improved TFIDF | MNN |

The eight models were applied to each dataset described previously.

### 3.6.Implementation

All our experiments are implemented in a cluster consists of one master and four workers, using Apache Spark with Java and Scala programming languages. Spark is a fast-general engine for large-scale data processing. Powered by big companies and organizations such as NASA, Amazon, Alibaba, eBay, IBM, Samsung, and Yahoo!. For MNN, we used Sparkling Water, which is a library for deep learning powered by H2O with support to Apache Spark. H2O is an open source, in-memory, and scalable machine learning platform that allows you to build machine learning and deep learning models on big data.

Spark and H2O enabled us to apply our experiments not only on one machine but on a cluster, which made the implementation and run time faster and easier.

# 4. RESULTS AND DISCUSSION

## 4.1. Results

In this research, we introduced a new improvement to TF-IDF that will help in the feature weighting process and a new MNN model that will help in the classification process. To verify those two contributions, we build multiple models as described in the previous chapter; the results were promising (see Figure 6,7,8).

Four different datasets (section 3.1) were used; each dataset was split into two datasets, training and testing. Amazon and Yelp datasets were used as a 5-class rating classification problem where the label is the rang of rating for every online review from 1 to 5. Also, the datasets were used as a 2-class semantic classification problem. Yahoo and AG news datasets were used for document classification. The eight models we built were applied to each of the training and testing datasets.

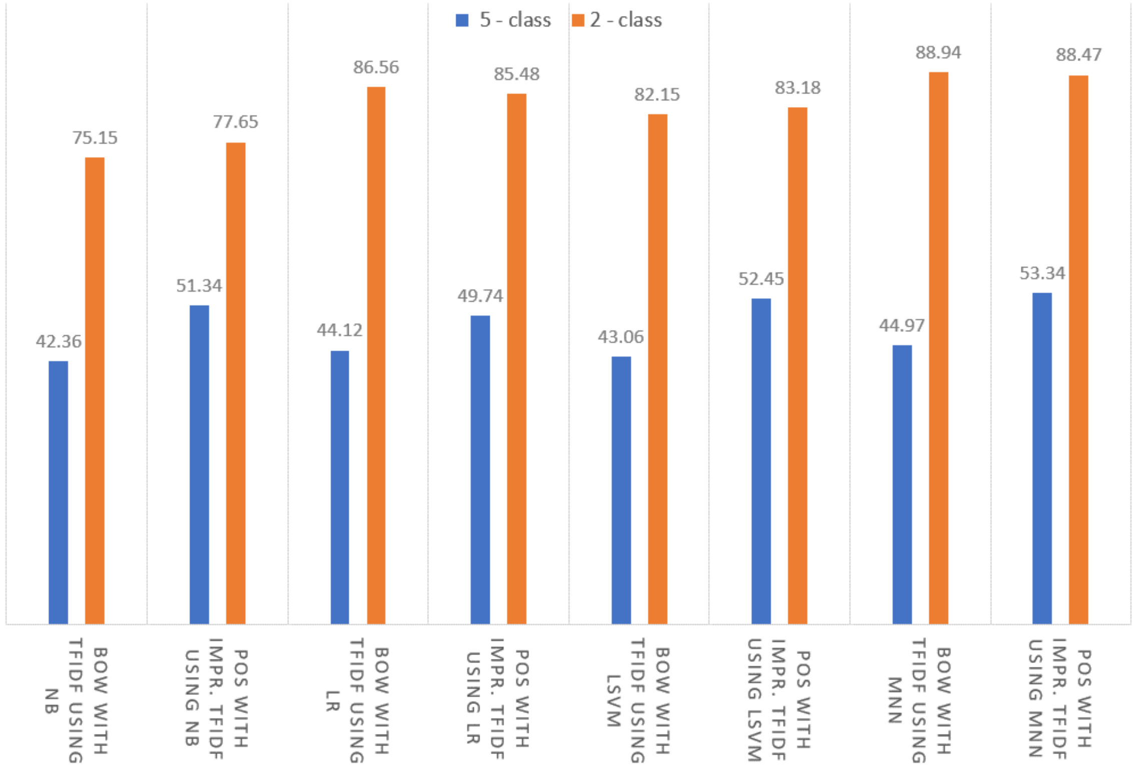

Figure 6. Results for the Amazon dataset.

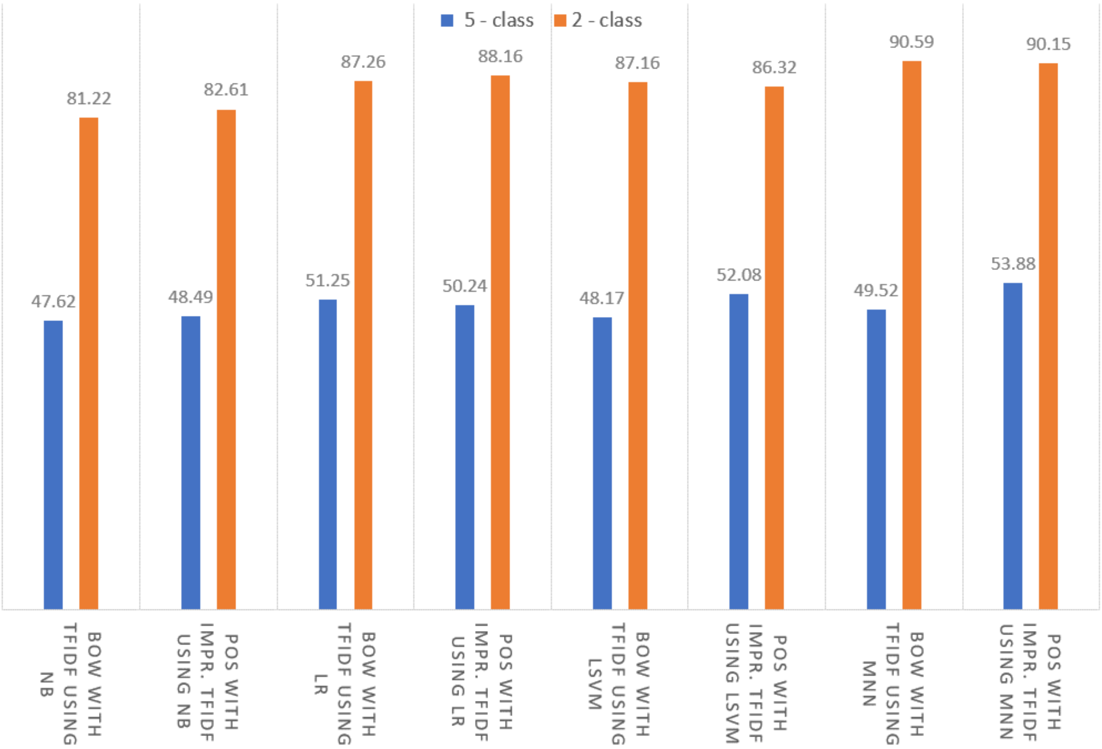

Figure 7. Results for Yelp dataset.

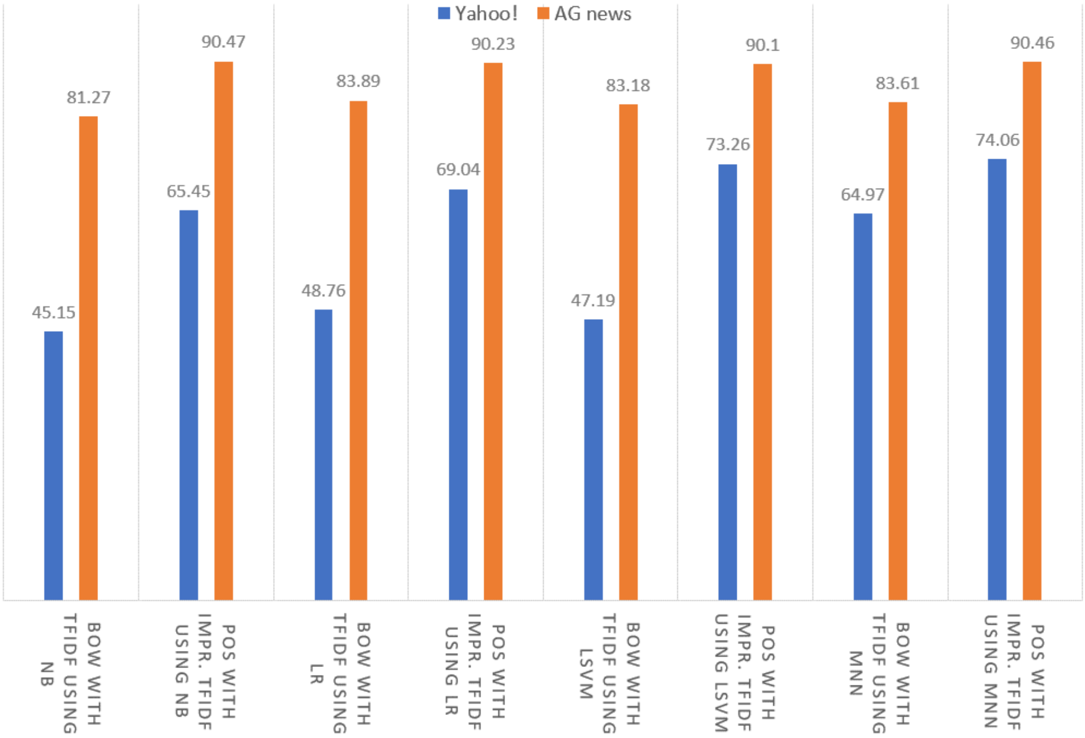

Figure 8. Results for the Yahoo and AG datasets.

In our experiments, we evaluate the models by its confusion matrix, in specific, by using the accuracy measurement. Accuracy is used to describe the performance of a classification model in term of true and false predictions, where accuracy is the summation of the Ture Positive and True negatives divided by the total number of predictions. All the results in the following figures are represented based on the accuracy of the related model.

As descripted previously, Zhang et al. [27] collected the large datasets we used in our experiments and conducted their own studies on the datasets. They discuss the use of character-level CNNs for text classification and compare the results with multiple traditional and deep learning models, the following models are similar to our models:

- BOW with TFIDF: this is a traditional and popular model used in text classification, they select the 50,000 most frequent words to construct the bag-of-word model.
- N-grams with TFIDF: this also considered as a traditional model, the N in their experiments was up to 5.
- Character-level ConvNets: they constructed ConvNet with 9 layers, 6 convolutional layers and 3 fully-connected layers.
- Word-based ConvNets: for comparison, they constructed a word-based ConvNets with the same structure and layers as the character-level ConvNets.
- RNN: They also built a word-based long-short term memory (LSTM) RNN model.

The results that have been presented by Zhang [27] shows that the character-level CNN model scores the highest accuracy among the other state of art models. Specifically, they achieved 59.97% classification accuracy on the Amazon 5-class dataset when the character-level CNN model was applied, while the other state of art models achieved less accuracy on the same dataset. Furthermore, for Amazon 2-class dataset, the character-level model scores 95.07% classification accuracy, which again was the best among the state of art models. The results from Yelp dataset were the same as Amazon, were character-level model scores 62.05% and 95.64% on Yelp 5-class and Yelp 2-class consecutively. Finally, for Yahoo! and AG news datasets, the character-level model scores 71.02% and 92.36% consecutively.

Figure 9 presents a complete comparison between the results Zhang et al. [27] obtained using the character-level model and the best results we obtained using POS tagging.

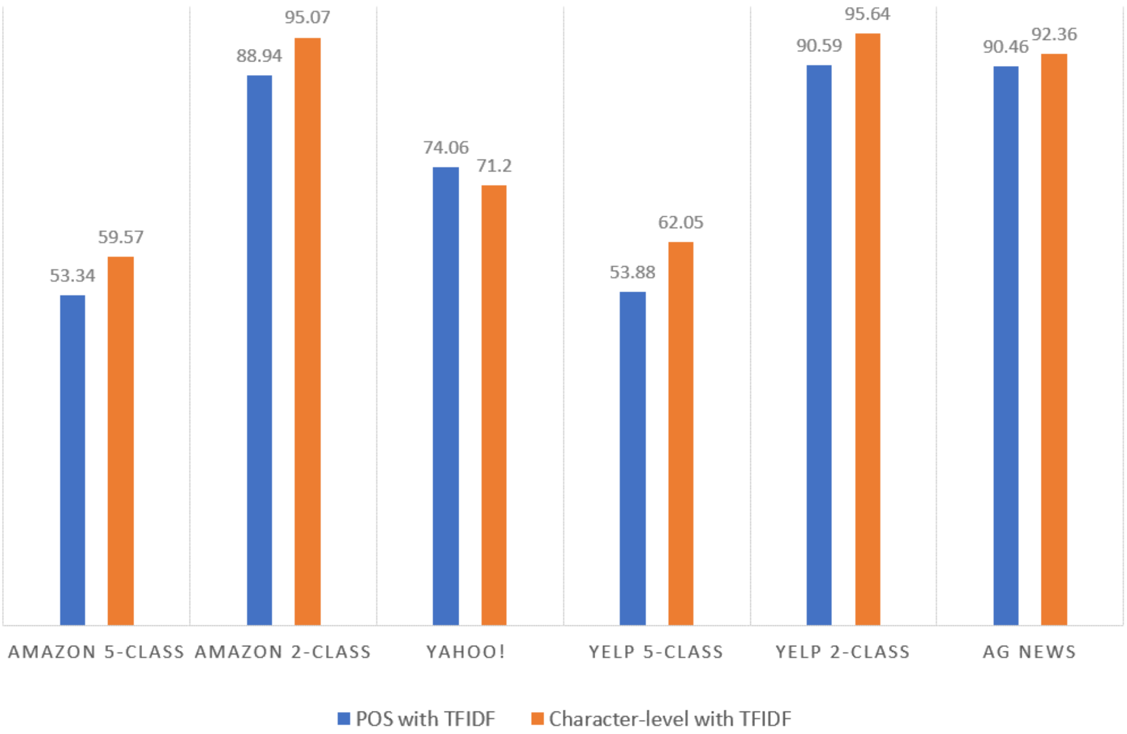

Figure 9. Comparison between Zhang et al.'s (Character-level) best results and our best results (POS).

### 4.2. Discussion

For classification models to successfully classify the input datasets, there must be some distinction between the attributes of each class in the dataset, otherwise, the model will not be able to classify the data correctly. In some cases, the differences between classes are clear and the model easily can find it. However, it is not the case most of the time.

The nature of each of the four datasets we used in this work are different, the Amazon and Yelp datasets are a collection of users' reviews on online products or services, which we used for rating and semantic classification. In the other hand, Yahoo! And AG datasets are a collection of documents each belong to a specific category. That diversity in the nature of the dataset leads to diversity in the POS distribution among classes in each dataset. We argue that the variation of the POS tag distribution can lead to better classification in some cases.

From figures 6,7,8 we can notice the following points:

- The introduced improvement of TFIDF outperforms the original method in term of accuracy in some cases. However, our improvement dose not shows constant and stable performance across different classification and regression algorithms and datasets.
- The MNN model we built preform very well against the classic machine learning algorithms. We can notice that the highest accuracy was scored by the MNN model.
- Expect the MNN model, LR had the best results, then linear SVM come second, and finally the worst was the NB model.
- Clearly using Part of speech tagging in those datasets provide more accurate and high-performance results than using the classic Bag of word method.
- The results were varying from the 2-class datasets to the 10-class datasets; in the 2-class Yelp dataset (figure 7) the liner machine learning algorithms with POW score a good accuracy in the test datasets range from 81% to 87%, which are in some cases acceptable scores. In the other hand, the same model scores a very low accuracy in the 5-class yelp dataset, range from 47% to 54%. This retreating in the accuracy because of the correlation between the five classes in Yelp; the same results were obtained from the Amazon dataset.

Furthermore, the results clearly show that using POS tagging improves the performance of the machine learning algorithms. POS tagging cover the limitations of the original equation of TFIDF because in addition to the statistical information about features, POS adding information about the class of the feature. For example, in the Yahoo datasets has 10 class, the classes were varied from fields such health, education, sport, and finance; this diversity in classes fields led to diversity in the type of features and the POS tags related to those features, therefore, adding the information about POS tagging helps the machine learning models perform better.

Another important note from the results was the new MNN model we built, the model performs as expected and superior the linear machine learning models. In term of accuracy, the model scores the highest results, however, unlike the linear algorithms, building MNN models is time-consuming. Our MNN model was built using a cluster with 64 core and 128-gigabyte ram, it takes nearly four days to train on the amazon dataset. The rest of the models trained on the same cluster but with less training time, for example, the NB model is completed in 10 minutes, yet its accuracy was the worst.

Figure 9 shows that Zang et al. models had better results in some of the datasets and our models in some other. To explain that, let us look at the distribution of the POS tags inside every dataset.

### 3.6.1. Distribution of POS

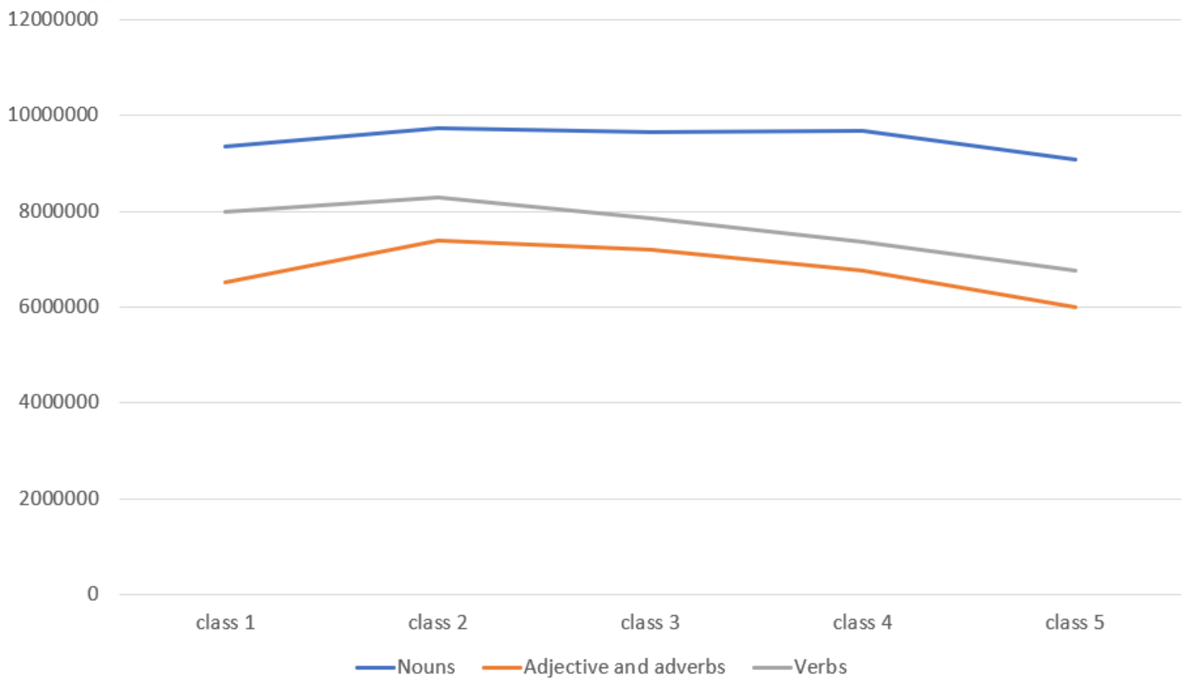

Figure 10. Distribution of POS in the Amazon 5-class dataset.

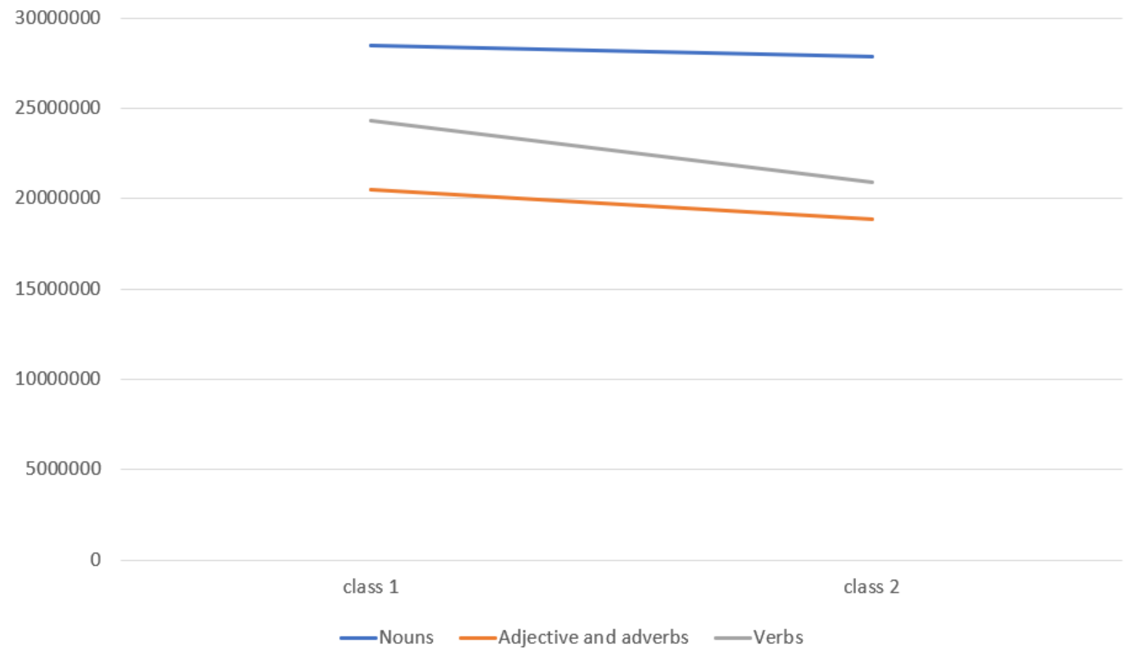

Figure 11. Distribution of POS in the Amazon 2-class dataset

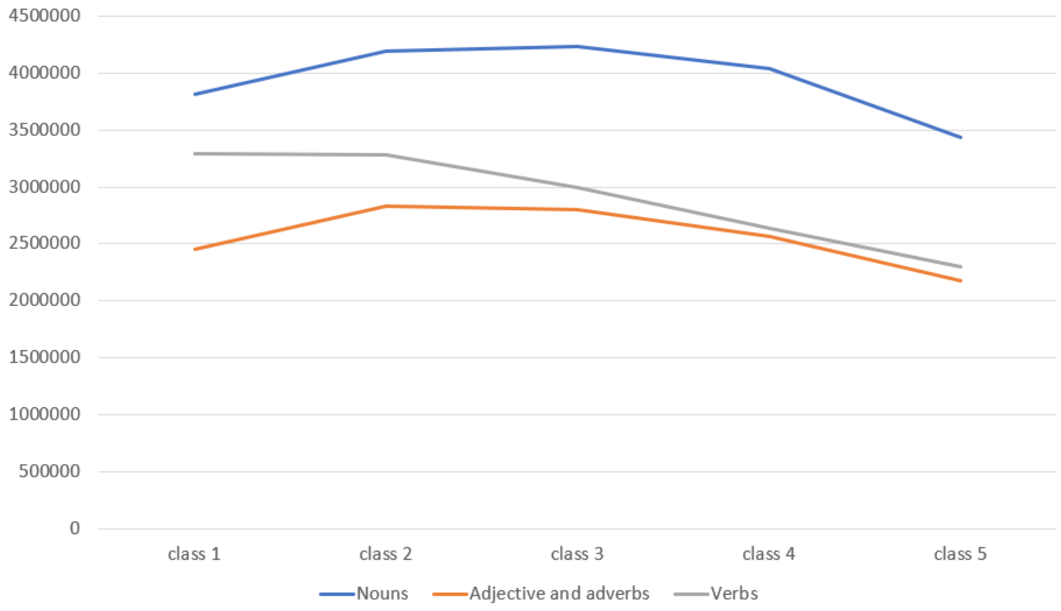

Figure 12. Distribution of POS in the Yelp 5-class dataset

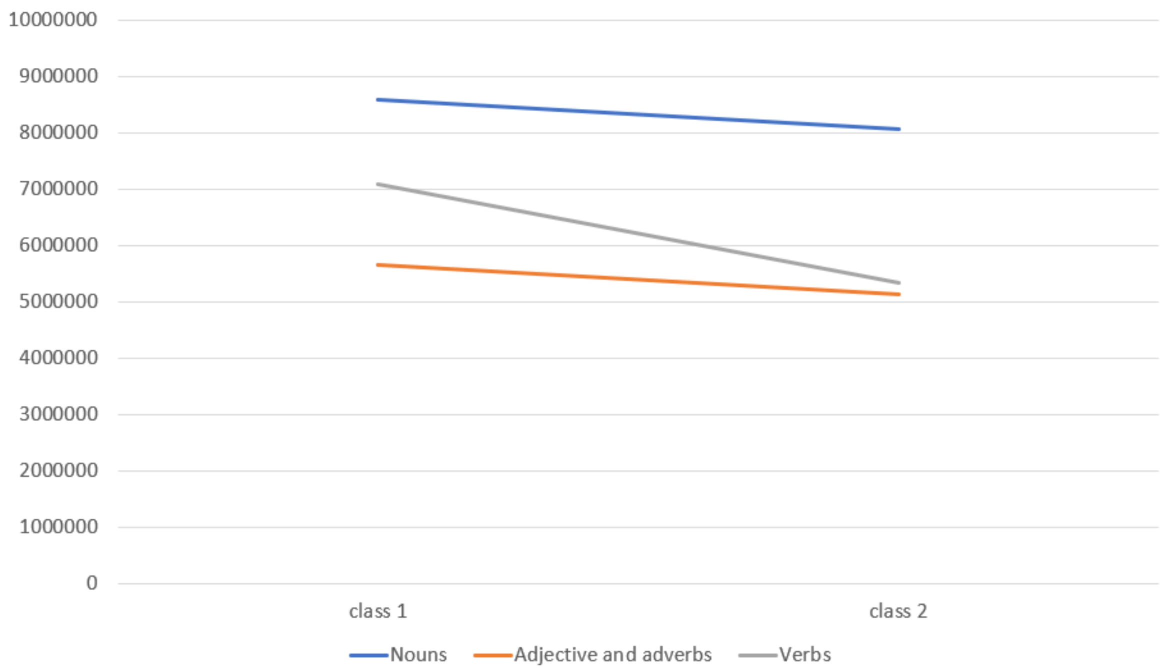

Figure 13. Distribution of POS in the Yelp 2-class dataset

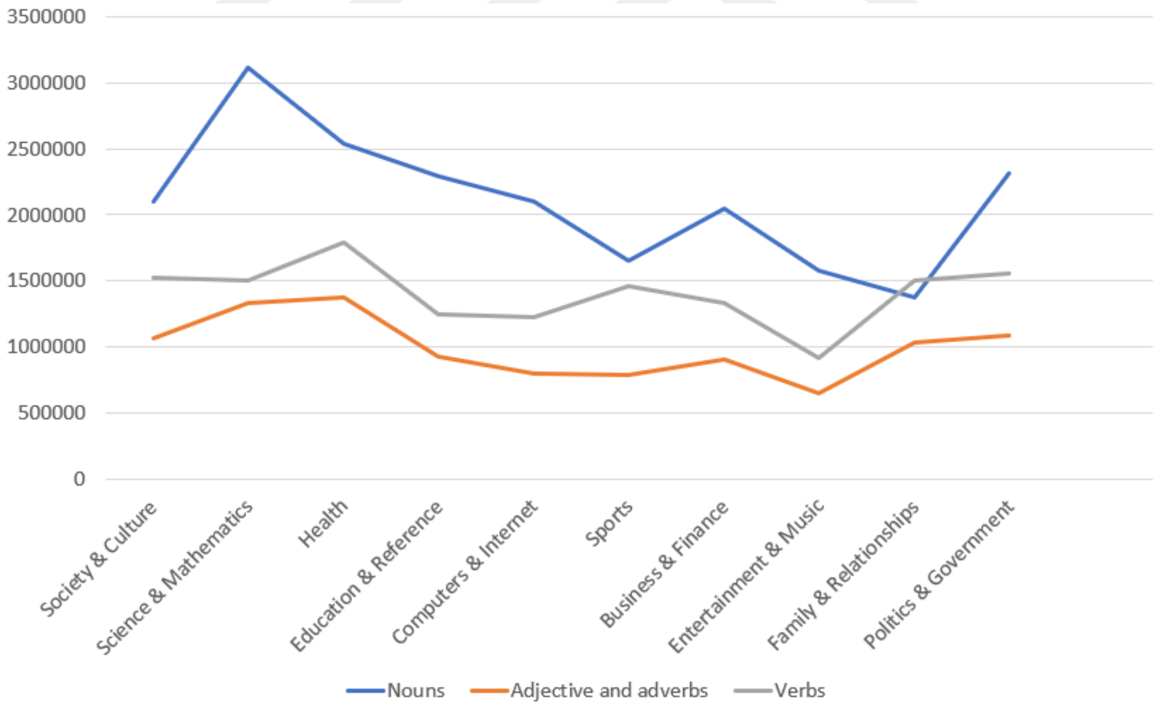

Figure 14. Distribution of POS in the Yahoo! dataset

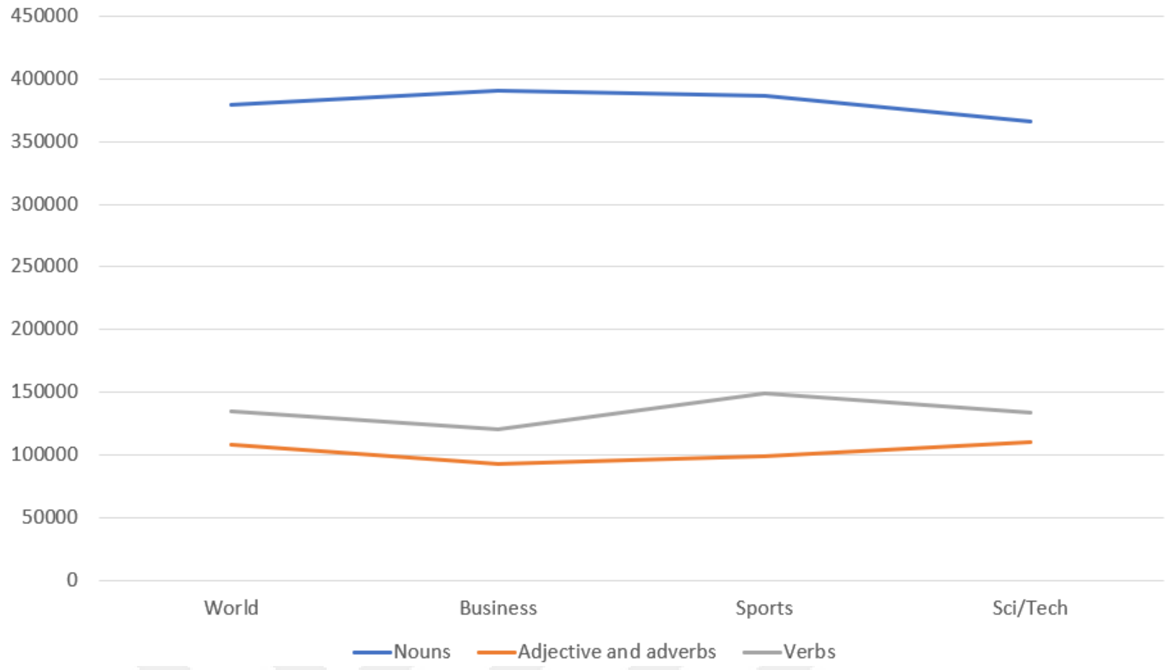

Figure 15. Distribution of POS in the AG dataset

From the previous figures (10, 12, 13, 14, 15), we can see clearly the difference in the distribution of POS tagging in the datasets. Furthermore, this difference in distribution leads our classification model to perform better than Zhang et al.'s in some cases as shown in figure 9.

From the distribution of POS tagging we can notice the following points:

- In datasets like Amazon and Yelp, the POS tagging distribution has a stable percentage across the different categories, therefore, our new feature weighting POS-based algorithm did not affect the results of the classification models.
- Unlike Amazon and Yelp, the POS tagging distribution in Yahoo! dataset were variable across the different classes in the dataset. Which leads the classification model to score high accuracy using the new POS-based feature weighting.
- In Yahoo! dataset, we can notice that the Verbs are more than the Nouns in the Family and Relationship class, this is a very important indication to the importance of Verbs in that specific class. Such information from the dataset must be used in the classification model, which is exactly what our POS-based feature weighting does.

- Also, the distribution of POS tagging in Yahoo! was not stable across classes, for example, in the Science and Mathematics class, the gap between Nouns and Verbs is approximately 1.5 million. In the other hand, the gap in the Sports class is approximately 200,000. Which prove that the importance of Nouns is higher in Science and Mathematics than Sports and the importance of Verbs is higher in Sports than Science and Mathematics.
- Furthermore, Due to the nature of the AG dataset, which is a news dataset, the Nouns distribution was the higher in all classes.
- Another side observation from the Amazon and Yelp datasets, from figures 10, 12 we can notice that costumers tend to write fewer words when they like (class 5) or dislike (class 1) a product or service.

To summarize the results, as shown in the above observations, we can clearly say that our new POS-based feature weighting method works well in document classifications such in Yahoo! dataset and does not have much effect in rating and semantic classification.

## 5. CONCLUSION

The huge amount and importance of unstructured text motived us to look deep into the process of extracting knowledge from it. Because of the nature of the text, the understanding unstructured text is more difficult than structured text, however, extracting useful information from unstructured text can be very helpful for businesses and costumers in general. With this motivation we first analysis the sources of unstructured text and found that one of the most sources is online reviews and news articles; therefore, we look for datasets related to those sources and found four datasets as descripted in section 3.1.

The process of extracting useful information from text generally called text mining, and when we try to put a label or predict to which class the unstructured text belongs then it called text classification which we focused on in the experiments. Text classification models are usually starting with preprocessing methods to clean the datasets from any noise or unwonted features, then the feature selection to select the most related and useful features, and because the feathers are text, another step is needed before train the machine learning algorithm, which called feature weighting, where every text feature convert to a numerical representation based on statistical measurements and the importance of the feature to the dataset.

One of the most popular feature weighting methods is TF-IDF, where every term (feature) is weighted based on two statistical calculation, which is frequency of the term in the document, multiplied by the logarithms of the total number of documents divided by the number of documents with term t in it. TF-IDF is a very useful method, however, it has some limitations; mainly the method neglects the class information of the text and treats all the documents as if they were belonging to one class. To overcome this limitation, we introduced a new improvement to TF-IDF, where we include the POS tagging frequency in the documents and classes, thereby we calculate the weighting of a term by using the stander TF-IDF with addition to the POS tagging frequency of the term. This improvement helps the machine learning algorithms to achieve high accuracy in the train and test datasets.

After the feature weighting algorithm, the final step is the actual classification or prediction, for this step, many machine learning algorithms could be used. In our work, we used three of the most popular algorithms, NB, linear SVM, and LR. And for the purpose of this work, we trained a new MNN model specific for text classification problems. The two main contributions presented in this work perform very well in the conducted experiments. We build eight models each was a combination of diffident method and algorithms, then those models were train and test on four large datasets. The obtained results were in favor of the new imperilments of TFIDF we present, also, the MNN had the best accuracy among all machine learning algorithms.

In our future work we will be focusing on investigation the POS tagging deeply. Moreover, transfer the MNN model to a deep learning model and combine it with our POS-based feature weighting.

## 6. REFERENCE

1. Chen, P., Wu, S., and Yoon, J., The Impact of Online Recommendations and Consumer Feedback on Sales, in ICIS 2004 Proceedings, 58.

2. Ye, Q., Law, R., and Gu, B., The impact of online user reviews on hotel room sales, International Journal of Hospitality Management, 28,1(2009) 180-182.

3. Qu, L., Ifrim, G., and Weikum, G., The bag-of-opinions method for review rating prediction from sparse text patterns, in Proceedings of the 23rd International Conference on Computational Linguistics, 2010, Beijing, China, 913-921.

4. Bakhshi, S., Kanuparthy, P., and Shamma, D., If It Is Funny, It Is Mean: Understanding Social Perceptions of Yelp Online Reviews, in Proceedings of the 18th International Conference on Supporting Group Work, 2014, Sanibel Island, Florida, USA, 46-52.

5. Bakhshi, S., Kanuparthy, P., and Shamma, Understanding Online Reviews: Funny, Cool or Useful, in Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing, 2015, Vancouver, BC, Canada. 1270-1276.

6. Danescu-Niculescu-Mizil, C., Kossinets, G., Kleinberg, J., and Lee, L., How opinions are received by online communities: a case study on amazon.com helpfulness votes, in Proceedings of the 18th international conference on World wide web, 2009, Madrid, Spain, 141-150.

7. Vasa,R., Hoon, L., Mouzakis, K., and Noguchi, A., A preliminary analysis of mobile app user reviews, in Proceedings of the 24th Australian Computer- Human Interaction Conference, 2012, Melbourne, Australia, 241-244.

8. Bakhshi, S., Kanuparthy, P., and Gilbert, E., Demographics, weather and online reviews: a study of restaurant recommendations, in Proceedings of the 23rd international conference on World wide web, 2014, Seoul, Korea, 443- 454.

9. Dialameh, M., and Jahromi, M., A general feature-weighting function for classification problems, Expert Systems with Applications Journal, 72 (2017) 177-188.

10. McCulloch, W.S. and Pitts, W., A logical calculus of the ideas immanent in nervous activity, Bulletin of Mathematical Biophysics, 5 (1943) 115-133.

11. Widrow, B., Generalization and Information Storage in Networks of Adaline `Neurons'," in Self-Organizing Systems, symposium proceedings, M.C. Yovitz, G.T. Jacobi, and G. Goldstein, eds., 1962, Spartan Books, Washington, DC, 435-461.

12. Widrow, B., and Hoff, M.E., Associative Storage and Retrieval of Digital Information in Networks of Adaptive `Neurons', Biological Prototypes and Synthetic Systems, 1 (1962) 160.

13. Hand, D. J., and Yu, K. Idiot's Bayes—Not So Stupid After All, International Statistical Review, 69,3 (2001) 385-398.

14. Hearst, M., Support Vector Machines, IEEE Intelligent Systems, 13 (1998) 18-28.

15. Penh, C., Lee, K., and Ingresoll, G., An Introduction to Logistic Regression Analysis and Reporting, The Journal of Educational Research, 96 (2002) 1-14.

16. Han, J., Pei, J., and Kamber, M., Data Mining Concepts and Techniques, Third Edition, Morgan Kaufmann, 2011.

17. Mostafa, M., More than words: Social Networks' Text Mining for Consumer Brand Sentiments, Expert Systems with Applications, 40 (2013) 4241-4251.

18. Dave, K., Lawrence, S., and Pennock, D., Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews, Proceedings of the 12th international conference on World Wide Web, 2003, Budapest, Hungary, 519-528.

19. Baker, D., and McCallum, A., Distributional Clustering of Words for Text Classification, Proceedings of SIGIR, 1998, Melbourne, Australia, 96-103.

20. http://www.tfidf.com/ Term Frequency-Inverse Document Frequency, 10 June 2018.

21. Kuang, Q., and Xu, X., Improvement and Application of TF•IDF Method Based on Text Classification, International Conference on Internet Technology and Applications, 2010, Wuhan, China.

22. Wang, N., Wang, P., and Zhang, B., An Improved TF-IDF Weights Function Based on Information Theory, International Conference on Computer and Communication Technologies in Agriculture Engineering, 2010, Chengdu, China.

23. Chen, K., Zhang, Z., Long, J., Zhang H., Turning from TF-IDF to TF-IGM for Term Weighting in Text Classification, Expert Systems with Applications, 66 (2016) 245-260.

24. Lan, M., Tan, C., Su, J., and Lu, Y., Supervised and Traditional Term Weighting Methods for Automatic Text Categorization, Transactions on Pattern Analysis and Machine Intelligence, 31 (2009) 721-735.

25. Wang, D., and Zhang, H., Inverse-Category-Frequency based Supervised Term Weighting Schemes for Text Categorization, Journal of Information Science and Engineering, 6 (2012).

26. Wang, D., and Zhang, H., Gradient-based Learning Applied to Document Recognition, Proceedings of the IEEE, 86 (1998), 2278-2324.

27. Zhang, H., Zhao, J., and LeCun, Y., Character-level Convolutional Networks for Text Classification, Advances in Neural Information Processing Systems, 2015, Montreal, Canada, 649-657.

28. Raja Chakrabarty, How did Neural Networks Get So Good?, Online Imgae, retrieved April 22, 2019 from https://medium.com/@rajachak127/how-did-neural-networks-get-so-good-d25a5593a6d4.

# CURRICULUM VIATE

Mahmoud MURAD was born in 20th of February 1993 in Gaze, Palestine. He has a bachelor’s degree in computer science from the Islamic University of Gaza, in 2015, Palestine. Mahmoud MURAD has been studying at Karadeniz Technical University since 2016 for a master’s degree in computer engineering. Murad speaks Arabic, English, and Turkish languages.

Publication:

Predict Public Opinion in Online Reviews with Imbalance Dataset and Feature Number. Mahmoud Murad, Murat Ekinci. 2017. Konya: s.n., 2017. International Conference on Engineering Technologies. p. 6.