SPECIFIC KEYWORD EXTRACTION FROM UNSTRUCTURED CURRICULUM VITAE USING DEEP LEARNING METHODS

MUSTAFA BUĞRA DÜR

FEBRUARY 2021

SPECIFIC KEYWORD EXTRACTION FROM UNSTRUCTURED CURRICULUM VITAE USING DEEP LEARNING METHODS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED
SCIENCES OF
ÇANKAYA UNIVERSITY

BY
MUSTAFA BUĞRA DÜR

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF
MASTER OF SCIENCE
IN
COMPUTER ENGINEERING
DEPARTMENT

FEBRUARY 2021

ABSTRACT

SPECIFIC KEYWORD EXTRACTION FROM UNSTRUCTURED CURRICULUM VITAE USING DEEP LEARNING METHODS

DÜR, Mustafa Buğra
M.Sc., Department of Computer Engineering
Supervisor: Prof. Dr. Hayri SEVER

FEBRUARY 2021, 61 pages

In today's world conditions, technology is developing day by day and the number of data on the internet is increasing considerably. With the increase in the diversity in the informatics sector, which plays an active role with these developments, new positions and new working areas are emerging. Having many data sources on the internet does not indicate that they are meaningful. Due to the rate of increase in data, it becomes increasingly difficult to distinguish between necessary and unnecessary information. The important thing is to be able to extract meaningful data from meaningless data. Data that provide semantic integrity and work is very valuable in every field today. In order to make people's job easier and to seize various opportunities in the computer age, very intensive studies are carried out in this field. When people make job applications in various fields, they send their personal resume information that introduces them to companies. In this way, the company learns the necessary information for the person to be recruited. Companies with many resumes in their hands and pools want to extract keywords with various methods to classify the people to be recruited and to use the data they have in the most efficient way for them, as in other sectors. With the help of keywords, the relevant category required for a text or a resume can be learned, and the shortest summary information about the subject can be obtained. In this way, classification can be made as desired and a semantic integrity can be provided. Keyword definition is not made for each file in each dataset. It can take a lot of time manually to extract keywords correctly, the rate of mistakes is high and manual intervention is very difficult. Therefore, there are many different approaches and studies for keyword extraction.

Statistical methods, linguistic methods, machine learning algorithms, deep learning methods and recently increasing artificial neural network methods have been used in keyword extractions. In the early studies, keyword suggestions were made for the most common words on the resume. In recent studies on artificial neural networks, it is aimed to deepen the learning for keyword extraction, to increase the accuracy rate more and to perform faster processing. In this thesis, approaches have been made on keyword extraction on resumes of the informatics sector. The data pool created from the resumes and the data in the data pool were used together with their explanations in order to create semantic integrity. Deeper learning is aimed with artificial neural networks, and it is used in keyword words studied on the target sector in order to give more accurate results. Thanks to keyword extraction; It is aimed to achieve semantic integrity and classify individuals more accurately than their background. It is aimed to achieve more successful classification with keyword extraction for large data sets.

Keywords: Neural Network, Deep Learning, Resume, LSTM, CV, Keyword Extraction

ÖZ

DERİN ÖĞRENME YÖNTEMLERİ KULLANILARAK ÖZGEÇMİŞLER ÜZERİNDE ANAHTAR KELİME ÇIKARIMI

DÜR, Mustafa Buğra
Yüksek Lisans, Bilgisayar Mühendisliği Anabilim Dalı
Tez Yöneticisi: Prof. Dr. Hayri SEVER

ŞUBAT 2021, 61 sayfa

Günümüz dünya şartlarında, teknoloji günden güne gelişmekte ve internet ortamında veri sayısı bir hayli artmaktadır. Bu gelişmeler ile beraber aktif rol oynayan bilişim sektöründe ki çeşitliliğin artmasıyla beraber, yeni pozisyonlar yeni çalışma alanları ortaya çıkmaktadır. İnternet ortamında birçok veri kaynağının olması bunların anlamlı olduğuna işaret etmemektedir. Verilerdeki artış hızı nedeniyle gerekli ve gereksiz bilgiyi ayırt etmek gitgide zorlaşmaktadır. Önemli olan anlamsız verilerden anlamlı bir veri çıkarabilmektir. Anlamsal olarak bütünlük sağlayan ve işe yarayan veriler günümüzde her alanda çok değerlidir. İnsanların işini kolaylaştırarak ve bilgisayar çağında çeşitli fırsatları yakalayabilmek adına bu alanda çok yoğun çalışmalar ortaya atılmaktadır. İnsanlar çeşitli alanlarda iş başvuruları yaparken, şirketlere kendilerini tanıtan kişisel özgeçmiş bilgilerini göndermektedirler. Şirket işe alacağı kişi için gerekli bilgileri en kısa yoldan bu şekilde öğrenmektedir. Ellerinde ve havuzlarında birçok özgeçmiş bulunan şirketler, işe alacağı kişileri sınıflandırmak ve diğer sektörlerde olduğu gibi elinde bulunan verilerin kendisi için en verimli şekilde kullanmak için çeşitli yöntemler ile anahtar kelime çıkarmak istemektedirler. Anahtar kelimeler sayesinde bir metin veya özgeçmiş için gerekli olan ilgili kategoriyi öğrenilebilir, konu hakkında en kısa özet bilgiye sahip olunabilir. Bu şekilde istebildiği gibi sınıflandırma yapılabilir ve anlamsal olarak bir bütünlük sağlanabilir. Her verisetinde bulunan her dosya için anahtar kelime tanımlaması yapılmamaktadır. Anahtar kelime çıkarımını doğru yapabilmek el yöntemiyle bir hayli zaman alabilir, hata yapılma oranı büyüktür ve el ile müdahale çok zordur. Bu yüzden anahtar kelime çıkarımı için birçok farklı yaklaşım ve çalışma söz konusudur. Daha önce yapılmış anahtar kelime çıkarımlarında istatiksel yöntemler, dilbilimsel yöntemler, makine öğrenmesi algoritmaları, derin öğrenme methodları ve son zamanlarda artan yapay sinir ağları methodları kullanılmıştır. Başlarda yapılan çalışmalarda, özgeçmiş üzerinde en çok geçen kelimeler için anahtar kelime önerisi yapılmıştır. Son zamanlarda yapay sinir ağları üzerinden yapılan çalışmalarda ise, anahtar kelime çıkarımı için öğrenmenin daha derinleştirilmesi, doğruluk payının daha çok arttırılması ve daha hızlı işlem yapılması amaçlanmıştır. Bu tez çalışmasında, bilişim sektörüne ait özgeçmişler üzerinde anahtar kelime çıkarımı üzerine yaklaşımlarda bulunulmuştur. Özgeçmişlerden oluşturulan veri havuzu ve veri havuzunda bulunan verilere ait anlamsal bütünlük oluşturması amacıyla açıklamalarıyla beraber kullanılmıştır. Yapay sinir ağları ile daha derin öğrenme amaçlanmış, daha doğru sonuçlar verebilmesi için hedef sektör üzerinde çalışılan anahatar kelimelerde kullanılmıştır. Anahtar kelime çıkarımı sayesinde; anlamsal bütünlük elde etmek, kişilerin özgeçmişlerine göre daha doğru sınıflandırılması hedef alınmıştır. Büyük veri setleri için anahtar kelime çıkarımı ile daha başarılı sınıflandırma elde edilmesi amaçlanmıştır.

Anahtar Kelimeler: Yapay Sinir Ağları, Derin Öğrenme, Özgeçmiş, Uzun kısa süreli bellek derin öğrenme, Anahtar Kelime Çıkarımı

ACKNOWLEDGEMENTS

Throughout the study, I would like to thank Prof. Dr. Hayri SEVER, who always guided me with his knowledge and experiences, and who did not refrain from his support and who tried to help me in all matters. It is a pleasure to express my special thanks to my family and my friends for their valuable support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

INTRODUCTION

Internet has become an indispensable need in our World. The number of people using the internet is increasing and age of internet usage is decreasing. People's desire to follow the innovations, the developments in the world, to access news and information are among the reasons that direct people to use the internet. According to the world internet usage statistics for 2020, the number of users using the internet is 4.57 billion people in total. This number refers to constitute 59% of the world population. Number of using internet increased by 7% compared to the previous year. This number will increase day by day when looking at the general picture.

Thanks to the use of the internet, new information and data are being created and their number is gradually increasing. The number of data increases considerably through social media and question-answer sites, forums, and sharing sites, and people want to find the right one for themselves.

Scientific research is the work done by collecting, interpreting and evaluating data to contribute to science. One of the main tasks of academics, who are members of the academic world, is to generate knowledge for the development and advancement of science and to guide the next research. The first aim of academicians is to publish publications to contribute to the world of science.

The development of technology has led to development and innovation in many areas and many disciplines and practices have forced them to adapt to these developments. It followed this process in digital publishing and adapted itself to technological development in order to exist in the digital environment. Thanks to the e-book, digitalization steps have been taken and people have gradually turned to digitalization. Increasing internet usage has also increased the demand for e-book usage. In this way, the information in the digital environment has also increased. As a result of increase in knowledge, lack of information and pollution has also emerged. Not all information on the Internet is true. Incomplete data may lead to a decrease in the success rate of research conducted while publishing academic papers. Due to missing data, the researcher carries out his research with missing data. Deficiencies in the publications can cause serious problems in the researcher's research such as information loss, decrease in statistical power, increase in standard errors. With Correction missing data, the researcher will make more successful research. Information loss will be reduced by finding missing data in the publications. The standard mistakes made by the researcher will decrease and the accuracy of the findings will increase.

As a necessity of today's age, people are on various platforms; They communicate in many types over the internet. When all of these provide integrity, they become an important value. Important inferences can be drawn from all of the information accumulated here. Two methods come forward to analyze the data collected here. Manual analysis comes to the fore as the first approach. Manually processing the information here is to suggest keywords. The other approach is to process large data more easily thanks to the algorithms taught to the computer. Data Mining is generally defined: It is the process of obtaining potentially useful, organized and structured data from unstructured and irregular text stacks. In the light of the obtained information, relationships, hypotheses and trends that are not clearly seen in the analyzed text sources are determined.

The keyword contains the main information that helps people understand text content. When you see keywords, it's easier for users to determine if they need to read the text, which will increase productivity. Scientific articles serve as a means of communication between researchers.

Researchers and academics benefit from previous studies while doing scientific research. One of the problems faced by the researcher due to the increase in data such as books, journals and articles in the digital environment is that the researcher cannot find the information. The shortest information about the text is expressed with the keyword in the shortest way what the text is about.

Keywords are the most obvious word or group of words that represent a text. Keywords give the reader a preliminary idea of the data they are looking for. At the same time, the use of keywords enables the reader to access information quickly and accurately. Usually, keywords are selected manually by the authors or publishers of a document. This situation brings the human factor to the forefront at the rate of accuracy in the extracted keywords and increases the possibility of error that may occur and may cause the reader not to access the information they are looking for.

Automatic keyword extraction is the process of extracting the shortest phrase, namely the keyword, that provides the shortest information about the document, directs what it is related to, expresses and categorizes the text best [1]. The purpose of this process is to get the shortest expression about the text. For the text to be used for various purposes, it is a process that enables the shortest information about the text without reading the text [2]. Keywords constitute 17% proportionally when the original text is considered. It provides the main theme that can be learned while reading the original article.

The first keyword extraction studies started in the 1950s and many methods used today are based on these studies. The term frequency (Term Frequency - TF) concept introduced by Hans Peter Luhn in 1957. This study is still an important part of keyword extraction studies and the most used statistical approach. Luhn said in his article that, “The more a word or phrase is repeated in the article, the more the author attaches importance to that word.” [3]. As a result of the studies, this approach is wrong, inadequate. It has been observed to give results. Because with this method, frequently repeated prepositions, conjunctions and so on. Although the sentence elements do not have a semantic significance in the text, they can rise to the top in terms of word repetition. A pioneering study on keyword extraction studies was published by K. S. Jones in 1972. He introduced the concept of Inverse Document Frequency (IDF) [4]. Unlike the high-frequency list of words, it enabled a healthier normalized keyword list. While decreasing the weight values of frequently mentioned words, it increased the weight values of less repeated words and created a more homogeneous scheme. The aim of this study is to obtain more accurate results.

CHAPTER II

APPROACHES USED TO KEYWORD EXTRACTION

As shown in the Figure 2.1 below, the classification of automatic keyword extraction based on approaches used in the current literature is as follows.

- Simple Statics
- Linguistic
- Machine Learning
- Hybrid

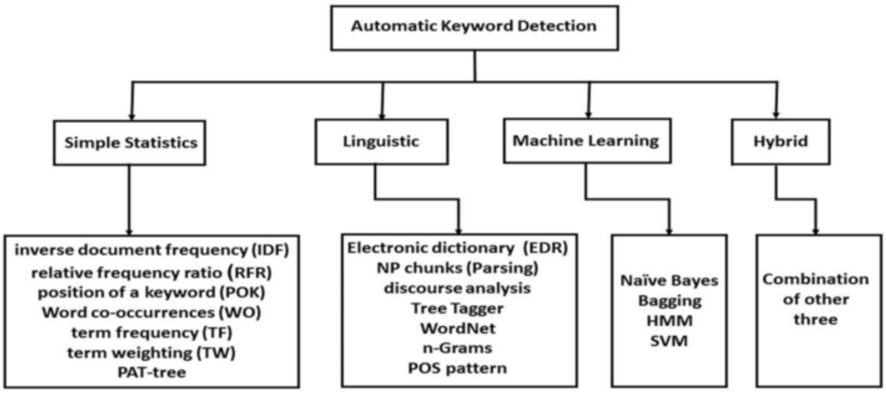


Figure 2.1 Approaches Used To Keyword Extraction[5]

2.1 SIMPLE STATISTIC APPROACHES

There are many methods to extract keywords. One of them and one of the first examples; Based on the statistics of individual words in the dataset. Since the process performed here is a simple statistical approach, it is a simple process and does not require training data. Any language is not important because it is considered statistically. In order to select the words in the text as keywords, the statistics of the words are checked: N-gram statistics, word frequency, term frequency–inverse document frequency (TFIDF) can be given as examples of these methods. When a meaningful keywords is selected for the document, it is defined by comparing it with other words in the text and comparing it with a more comprehensive text content. Continuously repeated words in the referenced text are not defined as keywords. Kumar and Srinathan developed an automatic keyword extraction method for English texts [6]. They use an algorithm based on the N-Gram filtering technique. Turney has made improvements to Keyphrase Extraction Algorithm (KEA) keyword extraction, thanks to statistical data among keywords. In this way, it aimed to get more accurate results [7].

It can be divided the approaches in this section into four different parts:

2.1.1 Free Indexing Based Approach

Keyword extraction does not depend on the support of other words in the text to be extracted. Based on the use of AI techniques;

(a) Based On Learning[8][9].

(b) Non-Learning Based Approach[10][11][12][13].

2.1.2 Controlled Indexing Based Approach

Keyword selection is chosen within a specific phrase. To give an example: Keyphrase Extraction Algorithm (KEA) is an algorithm for extracting keyphrases from text documents [14][15].

2.1.3 Term Frequency - Inverse Document Frequency

This is a technique used to measure a word in documents, usually calculating a weight for each word that point outs importance the word contained in the document and in compilation. This method is a technique commonly used in Information Retrieval and Text Mining.

To determine if sentence is a keyword, it followed the following properties: Term Frequency and INverse Document Frequency are recommended in combination with the Neural Networks that appear in the visible and paragraphs of the given document frequency [16].

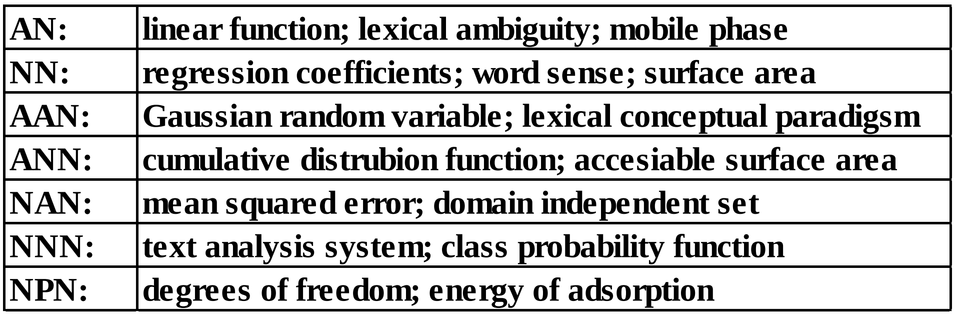
Zhou investigated predominantly complex web-based keyword extraction that includes network structure and grammar [17]. The focus is on reasonable node selection, accurate definition of relationships between words, simple weighted network and the construction of lexical network including TF-IDF. From the point of view of a linguistic approach, the greater the link between the keyword and its contexts taken from within the text and correctly defined, it tries to represent the texts as a network of these generated words.

2.1.4 N-Gram

The N-gram algorithm which has been developed in studies on language modeling since ancient times, is a preferred algorithm model for keyword prediction [18].

2.2 LINGUISTIC APPROACH

Linguistic approaches work on sentences and documents to learn word features. This approach consists of word analysis, syntax analysis, speech analysis, and analysis of such situations. The biggest advantage of this approach is that it can be applied to many languages, not just one. In other words, it can be said that it is open to analysis independently. Although these analyzes did not give very accurate results compared to linguistic methods, they had good results on statistical analysis. Nguyen and Kan came up with an algorithm to extract keywords from grammar. This algorithm is a type of algorithm that captures the conditions under which the keyword is extracted from the data. According to Justeson [19], keywords have some familiar patterns. These patterns depend on the way the expression speaks part of speech (POS).



AN:	linear function; lexical ambiguity; mobile phase
NN:	regression coefficients; word sense; surface area
AAN:	Gaussian random variable; lexical conceptual paradigsm
ANN:	cumulative distrubion function; accesiable surface area
NAN:	mean squared error; domain independent set
NNN:	text analysis system; class probability function
NPN:	degrees of freedom; energy of adsorption

Figure 2.2 Phrases in English[2]

The table used in his study and categorized by himself is shown in Figure 2.2.

N represent : “Noun”.

A represent : “Adjective”.

P represent : “Pronoun”.

Bao [20] worked on the Chinese keyword in algorithms. In this study the features of the Chinese language are discussed. Examining keyword properties, these researchers divided the syntactic functions of keywords into four areas: common nouns, modifiers, noun and verb expression. When looking at the basic characteristics of Chinese keywords, it is discussed that some words emerge from the synthesis of other words. With this inference, twenty-two types of rhetorical methods are presented to my suggestion. These types are; It is called singular, double and triple grammar. A single grammar contains 16 types of methods, including main noun and modifier categories. Triple grammar consists of a noun phrase.

Hu et al. [21] proposed an algorithm for extracting a new keyword that incorporates the features of the language to indicate the position importance of the word in a document. The name of this algorithm is Position Weight PW. This algorithm includes common categories before and after words. Three methods were used for this; Term Frequency Inverse Term Frequency (TFITF), CHI-Square and Position Weight Inverted Position Weight (PWIPW). Based on these methods, original words were combined for word formation. Three main factors were used in this; paragraph weight, word weight and sentence weight.

2.3 MACHINE LEARNING APPROACH

Researchers are extensively developing new methods of machine learning. Processes such as researching new types of knowledge, multidimensional machine learning problems, and integrated models with techniques from other disciplines can be given as examples to these developing fields. Some mining methods are also exploring how data miner performance values can be used to evaluate the significance of discovered models and guide the discovery process. Machine learning methodologies include different applications such as machine learning in one-dimensional problems and machine learning in multi-dimensional problems.

Machine learning is used to predict a result (predictor) or to describe a specific result (descriptive) in the analysis. For this reason, machine learning functions are gathered under two main tasks: predictive and descriptive. The purpose of predictive functions is to predict a property of an object based on its other properties. The predicted value can be either a continuous numerical value or a categorical value. In the foresight stage, the model is trained using historical data and therefore these functions are known as Supervised Learning methods.

Descriptive tasks, on the other hand, aim to reveal the relationships between features through machine learning techniques, and while doing this, the model is not subjected to a training process. Therefore, the descriptive tasks are known as unsupervised Learning. The machine learning type Figure is shown in 2.3 [22].

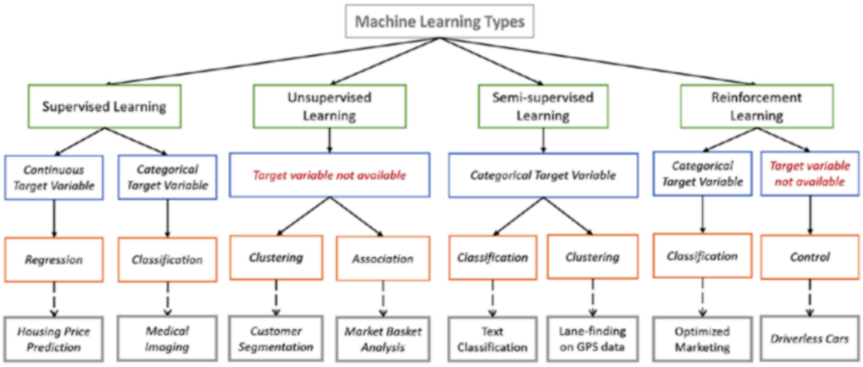


Figure 2.3 Types Of Machine Learning[22]

2.3.1 Supervised Learning

They are functions that use historical data to predict a future outcome. During the analysis of the attributes of a new variable and classification of this variable into a specified group, it makes use of the information to which class the variables belonged to in the past. The factors and results that are used to predict the possible outcome are included in their modeling. While establishing the model, the results obtained in the historical data according to the values taken by the factors are used as input. Expected result; It is a binary or categorical value or numerical value in the form of "Yes-No". The performance value of the predicted results is as important as the predicted result. For the quality of this result, often with the predicted result; confidence interval, probability, accuracy rate, etc. values are determined. In predictive models, it is aimed to develop a model based on the data with known results and to estimate the result values for data sets with unknown results by using this model. The purpose of the predictive functions is to estimate some fields in the database in relation to other fields. If the area to be predicted is a numerical (continuous) variable, the prediction problem will be a regression problem. If the area to be estimated is a categorical variable, it is a classification problem.

There are many variables used for classification and regression. The problem with predictive models; In other fields (inputs), the training data set for each observation where the target variable value is given and the set of assumptions that reflect the knowledge previously possessed about the problem can be interpreted as determining the possible value of the variable to be predicted.

2.3.1.1 Classification

The process of dividing data sets into classes is called classification. This process is a type of process that can be implemented on structured and unstructured data. The process of separation into classes occurs as a result of estimates in line with point calculations of the data. Classes are generally grouped under labels, targets, and categories. This estimation method is performed by modeling the data and calculating the input and output approaches of the objects in the data. Here, the main goal is to decide which class the data belongs to. Classification Terminology In Machine

Learning as expressed: that uses an algorithm that decides which category the data should be mapped to.

There are two types learners in classification as lazy learners and eager learners.

Lazy Learners: in this type of learning, the data remains on hold until the test data arrives. Then these pending data are classified according to where they belong. They have the ability to make more guessing calculations (K-nearest neighbor).

Eager Learners: this type of learning is also revealed as a classification model based on the training given before it is taken. A single hypothesis for data is enough for them. For this reason, they take a lot of time in training, but the conclusion is reached in little time for prediction (Naive Bayes, Artificial Neural Networks).

2.3.1.2 K-Nearest Neighbor *(KNN)*

K-Nearest Neighbor Algorithm (KNN) is an algorithm that is widely used among the classification methods in terms of the learning time is powerful and convenient and the implementation is simple to apply. There is a study on keyword extraction for Chinese words using KNN [23].

K value determined in this algorithm belongs to the closest neighbor number to be used to indicate to which class the object belongs. Object belongs to the class which is most close to determined K value.

If it can be summarized the calculation steps in order to make the method more understandable;

- The K-Value is determined for the object to be determined.
- The distance of this object to the target point is calculated using alternative methods.
- The closest neighbors are listed according to the determined distance calculations.
- The categories of the closest neighbors are determined.
- The nearest neighbor determined from within the nearest neighbor now means the new class of the specified object.

2.3.1.3 Naive Bayes

Naive Bayes argues that a particular feature in one class has nothing to do with the feature existing in another class. That is, even if the properties appear to be the same, it assumes that they are not related to each other. The assumption that every feature is independent; probability-based method is widely used [24]. It is easy to apply and is more compatible on large sets. Naive Bayes is a probabilistic model and also the algorithm is easily applicable. Various studies have been conducted for keyword extraction using the naive bayes method [25].

Listed applications of Naive Bayes Algorithms;

▪ Real time Prediction: The classifier used for real time prediction is a very fast method.

▪ Multi-class Prediction: It is a multi-class prediction algorithm used for class prediction of more than one variable.

▪ Text Classification / Spam Filtering / Sentiment Analysis: NB classifiers, which are based on independence, have very high successful classification results in spam filtering and sensitivity analysis compared to other algorithms.

▪ Recommendation System: The NB classifier is also quite common for classifying invisible information. It has very good results for a machine's prediction system.

2.3.1.4 Support Vector Machine Algorithm

When complex categorization problems are desired to be solved, SVM is one of the most preferred and one of the most powerful algorithms thanks to its method. This work has been discussed academically by Zhang [26]. Originally, SVM was used to linearly separate data between two classes. SVM is based on estimating the hyperplane, in other words, the decision function that can distinguish two classes in the best way and make a classification decision.

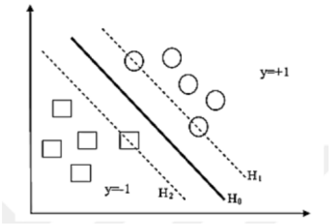


Figure 2.4: Linearly Separable Data

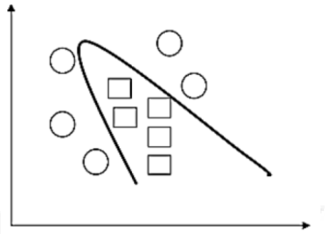


Figure 2.5: Linearly Non-Separable Data

In linearly separable classes, suppose that the data set is expressed as "((x_1, y_1), (x_2, y_2),.., (x_n, y_n),)" with "n" being the total number of classes. The form of "y ∈ {+1, -1}" until "i = 1,2,., n"; It holds the class label of the "x_i" values. The "x_i" values that can be divided linearly into two classes and the "H0" plane showing that they are divided into two classes linearly are shown in Figure 2.4.

Hyperplanes can be drawn in many ways to separate the data set on the plane. The important thing here is that the distance between the closest points of the hyperplane separating the two classes is max.

"H0" seen in Figure 2.4 is this optimum hyperplane sought; "H1" and "H2" are vectors that are the determinants of the boundary width, known as support vectors. The current data set may not be linearly separated as seen in Figure 2.5. In order to solve the classification process, it will be easier to move the data to a larger space.

2.3.1.5 Neural Network

The neural network model is a language model application with neural networks and neurons that have the ability to learn by imitating the human brain. Artificial neural networks (ANNs) are computer systems developed with the aim of automatically realizing the capabilities of the human brain, such as the ability to generate and discover new information through learning, without any assistance. Artificial neural networks have emerged as a result of mathematical modeling of the learning process by taking the human brain as an example. During the learning, rules are set by giving input and output information. Artificial neural networks are mainly used in areas such as diagnosis, classification, prediction, control, data association, data filtering, interpretation.

In artificial neural networks, information is held by the weights of the connections of the nerves in the network. For this reason, how the weights are determined is important. A 3 layer (or layered) feed forward neural network model consisting of Input, Hidden and Output layers is seen in Figure 2.6.

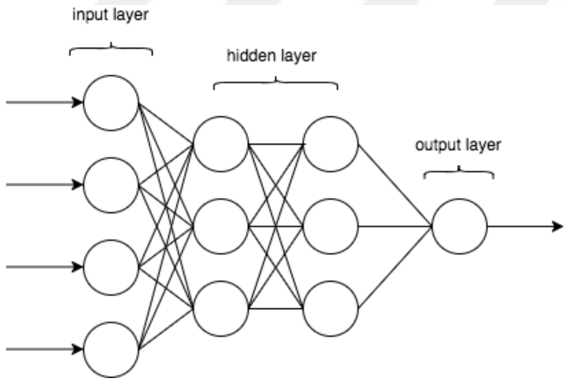


Figure 2.6: Neural Network Layers

It can be categorized Feed-forward neural networks and Feedback artificial neural networks.

In feedforward neural networks, neurons are in the form of regular layers from entry to exit. The information coming to the entrance of the artificial neural network is transmitted to the middle point, in other words to the cells in the hidden layer, without being changed. It then passes through the output layer, respectively, and transferred to the external environment. An example is shown in the Figure 2.7.

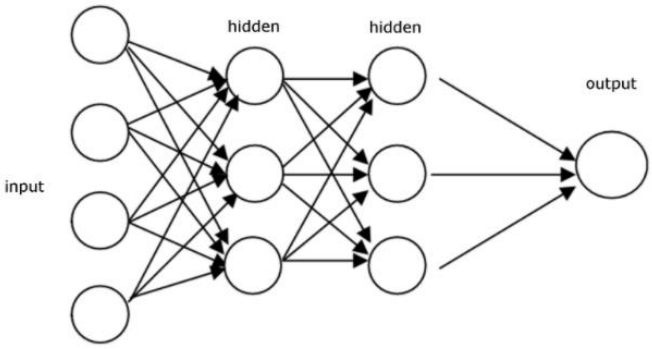


Figure: 2.7 FeedForward NN

A feedback neural network is a network structure in which outputs and outputs at the intermediate floors are feedback to the input units or to previous intermediate layers. Thus, the inputs are transferred both in the forward and backward direction. An example is shown in the Figure 2.8.

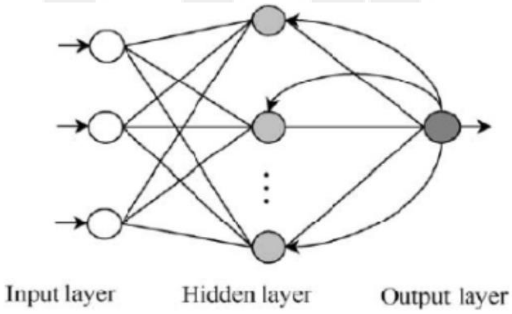


Figure 2.8: FeedBack NN

Artificial neural networks are structures formed by connecting artificial nerve cells to each other. Artificial neural networks are examined in three main sections; input, hidden and output layers. It shown below in Figure 2.9.

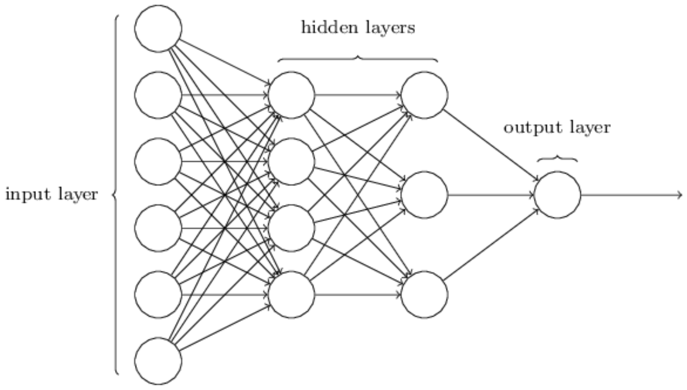


Figure 2.9: Neural Network Layers

Input layer is the layer in which the attributes of the sample that is wanted to be learned is input as input to a network. In the input layer, there should be as many input neurons as the number of features of the examples to be taught.

Hidden layers are the layers between the input layer and the output layer. Forward calculations and backward error propagation are made in these layers. The high number of layers causes computational complexity and increased computation time. In general, the number of layers and the number of neurons in the layers are high for problem solving in complex problems.

Output layers that processes information from hidden layers and produces outputs corresponding to data from the input layer of the network. The outputs produced in this layer are sent to the outside world. New weight values of the network are calculated using the output produced in this layer in feedback networks. It is the layer where the class information or label value of the samples to be learned in the artificial network is calculated as output.

2.3.2 Unsupervised Learning

The purpose of this learning path is to reveal the learning that makes the given inputs and outputs the same. In some problems, the data used for training consists of a storage vector without any target. The main target in these types of learning problems is to find similar examples from those found in the data or to determine the distribution of these data in space by focusing on the density calculation of similar samples.

Issues with Unsupervised Learning:

- It is more difficult than supervised learning.

- It is difficult to state that the results are meaningful exactly.
- External evaluation should be done.
- Internal evaluation should be done.

2.3.2.1 Clustering

One of the important techniques used in data mining is clustering analysis. Clustering focuses on the clusters and groups that will emerge, and the resulting clusters are expected to be homogeneous and heterogeneous among themselves. As mentioned above, clustering combines homogeneous objects with each other to create heterogeneous groups and units are placed in a hierarchical order. Classification allows observation results to be aggregated with little loss. In the literature, keyword extraction using clustering was done by Haggag et al.[29] The classification example is shown in Figure 2.10.

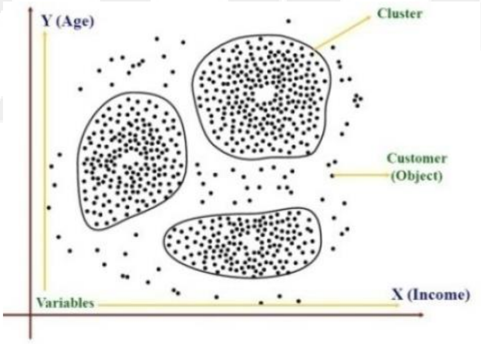


Figure 2.10: Clustering Schema[30]

The main purpose of clustering is to detect the similarities or distances between the observed individuals or objects. Similarity is explained as the strength of the relationship between two objects or two properties. This quantitative value is obtained in different ways depending on the scale or data type. Distance measures the differences, which are a measure of the contrast or incompatibility between two objects. Similarity and distance measurements allow observations to be distinguished from each other so that observations are divided into groups. Distance measurement differs depending on whether the data are quantitative or mixed data.

2.3.2.2 Association Rules

Association Rules in data mining were proposed in 1993 by Agrawal [31]. Association Rules is an unsupervised data mining method that extracts relationships between large data sets and analyzes the possibilities of objects realizing together.

Some basic concepts used in the Association Rules are given below;

- Sets consisting of one or more items are called a Set of Items.
- The incidence of product groups in the set of items is expressed as the Number of Supports.
- Support: It defines how frequent the correlation is in the data and is calculated by the ratio of the associations in which the set of items is to the total number of associations. The support is expressed as (A => B).
- The set of items whose support value is greater than or equal to the threshold is called Common Items.
- Trust: Gives the probability that a person who has bought good A will receive good B. It expresses the accuracy of the unity between items. Confidence is shown as (A => B).

To explain with an example of the concepts: If those who buy 1 pack of milk in a market also buy yogurt product, this is expressed as Milk Yogurt [support = 3%, trust = 70%]. The support and trust values here are the difference values of the association rule.

A support value of 3% indicates that 3% of all analyzed 50 purchases have both milk and yoghurt sales. The 70% confidence value reveals that 70% of those who buy 1 packet of milk also buy yoghurt. It is said that there is an association rule for objects that exceed the threshold trust value determined by the data miner. While determining association rules in large data sets, a process consisting of 2 steps is followed. The first step is to find frequent repeating objects: each of these objects are repeated at least as often as the predetermined number of threshold supports. In the second stage, strong association rules are created among frequently repeated objects. The main method used in association analysis models is the Apriori algorithm.

2.4 HYBRID APPROACH

This approach generally includes statistical approach, machine learning approach and linguistic approaches. In this approach, Bharti et al. used information such as the length of the words, the position of the words, the layout features of the words, and html tags in keyword extraction[5].

Hybrid approaches to keyword extraction combine basic methods or, in the keyword extraction task, the position of words, length, layout feature, html tags around words, etc. Uses some intuitive information such as. However, these can be applied to more than one document depending on their suitability. Since the name expression contains very important information about the text document, the name expression is defined and keywords are extracted. Keywords are selected on the basis of their linguistic features [32] and informative features such as highlighted words [33].

Query-oriented and words in summaries or titles can be part of candidate keywords. Methods such as co-occurrence [34] and machine learning [35] have been used to extract keywords from a single document. Topics are detected using keyword clustering. In addition, extracting and clustering related keywords based on query frequency history [36] is one of the methods adopted.

Wang et al. [37] proposed a hybrid method based on TF and semantic strategies for keyword extraction for more than one word in the Chinese document in their proposed method. Word segmentation, feature calculation and detailing are shown in Figure 2.11.

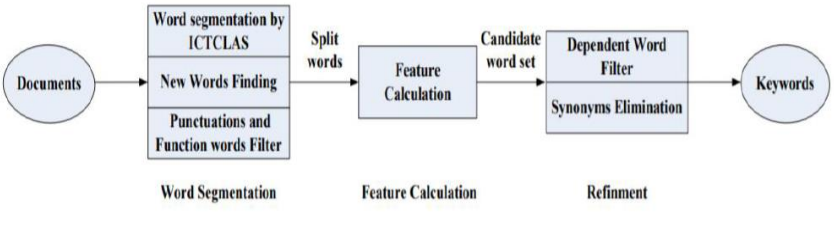


Figure 2.11: Word segmentation, feature calculation and refinement[34]

A document is represented as a graph where nodes represent terms and edges represent the occurrence of terms together. Whether a term is a keyword is determined by measuring its contribution to the graph. Wang et al. [16] presented a Neural Network based keyword extraction approach using traditional term frequency, reverse document frequency and position (binary) properties. The neural network model for keyword extraction is shown in Figure 2.12.

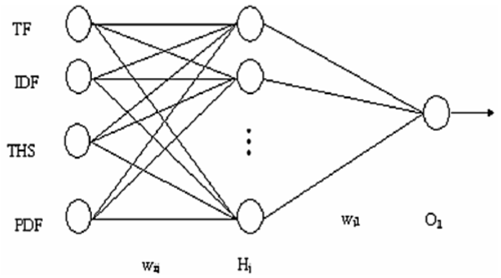


Figure 2.12: Neural network model for keyword extraction [16]

The neural network has been trained to classify a candidate phrase as a keyword or not. In [38], keyword extraction problem was seen as a supervised learning task. In this task, nine attributes are used to score a candidate sentence; some properties are the position of the expression in the document and whether the phrase is a suitable name. Keywords are extracted from candidate statements based on examination of their properties. Turney's program is called Extractor. One form of this Extractor is called GenEx, which is designed based on a set of parameterized heuristics that are fine-tuned using a genetic algorithm. Turney compares GenEX to a standard machine learning technique called Bagging, which uses a bag decision tree for keyword extraction and shows that GenEX outperforms the bagging procedure.

A keyword extraction program named Kea developed by Witten et al. [39] uses Bayesian learning for the keyword extraction task. An example is learned from training documents with sample keywords and corresponds to a specific group of training documents. Each model consists of a Naive Bayes classifier and two support files containing phrase frequencies and interrupted words. The learned model is used to describe keywords in a document. In Kea and Extractor, candidate keywords are defined by dividing the introductory text by phrase boundaries (numbers, punctuation marks, dashes, and square brackets, etc.). Finally, a sentence is defined as a sequence of one, two or three words that appear consecutively in the text. Phrases that begin or end with a stopped word are ignored. Kea and Extractor used controlled machine learning-based approaches. Two important attributes are the location of the first appearance of the expression to the document and the TF * IDF (used in the retrieval setting) for the development of Kea. Here TF corresponds to the document frequency of the statement, and the IDF is estimated by calculating the number of documents in the training number containing the statement in the statement group. Witten et al. [39] Compares Kea's performance to Turney's work and shows that his performance is comparable to GenEx proposed by Turney. Also, Witten et al. [39] claim that training the Naive Bayes learning technique is faster than GenEx training, which uses a special-purpose genetic algorithm for training. Data training and extraction processes are shown in Figure 2.13.

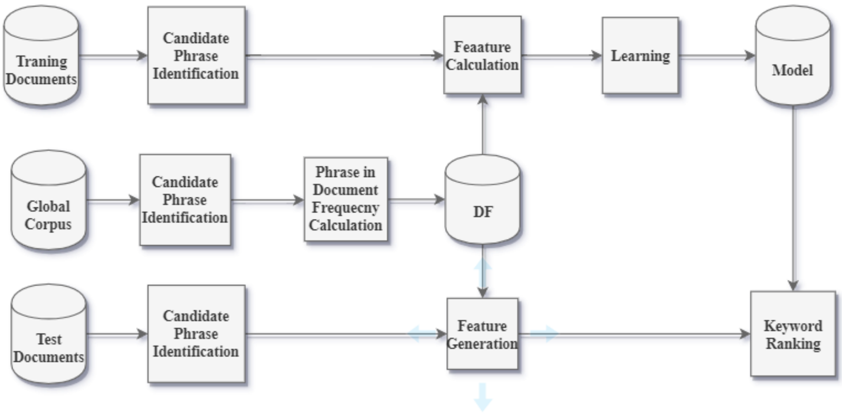

Traning Documents
Candidate Phrase Identification
Feaature Calculation
Learning
Model
Global Corpus
Candidate Phrase Identification
Phrase in Document Frequecny Calculation
DF
Test Documents
Candidate Phrase Identification
Feature Generation
Keyword Ranking

Figure 2.13: Data training and extraction processes[36]

They suggest that deriving domain-specific models would be less practical with the original long genetic algorithm approach. An effective keyword extraction system called KP-Miner developed by El-Beltagy et al. [40] uses simple features such as term (phrase) frequency (TF), inverse document frequency (IDF), and location. One of the important features used in this keyword extraction system is the boost factor used to increase the weight of the TF * IDF of multi-word expressions. This is based on the fact that the frequency of a multi-word expression in a small group is lower than a single word expression. This issue is dealt with in a slightly different way in [41]. A neural network based keyword extraction system has been presented [41]. This uses features such as phrase frequency, phrase links to other phrases, reverse document frequency, phrase position, phrase length, and word length.

Sarkar has taken a hybrid keyword extraction approach in the medical field. The approach he proposes combines domain knowledge with phrase frequency, reverse document frequency, and statement position more effectively. An effective candidate keyword identification component has also been used as the first part of the proposed keyword extraction system [42].

CHAPTER III

KEYWORD EXTRACTION WITH DEEP LEARNING

Machine learning, a branch of artificial intelligence has become a popular method used in areas such as system, object recognition, speech transformation, and mapping products to interests. It aims to learn data and calculate accuracy without the need for programming. In machine learning, they aim to develop the decision and prediction mechanism about this data by developing algorithms on the data and are trained in this direction. Decisions made and estimates made on the data give us information about how good the algorithm is. Currently, machine learning returns to us in response to voice commands. In addition, it makes predictions about the topics researching today or on a product that has been purchased before. It is an inevitable fact that machine learning will add efficiency to our lives as cost reductions are achieved and new powerful algorithms are created. Due to the characteristic of its architectural structure, it provides great help in solving artificial intelligence problems. In this way, the areas of use today are increasingly diversified.

It has significantly improved the latest technology in many different artificial intelligence tasks such as deep learning, object detection, speech recognition, machine translation. Its deep architectural nature allows it to solve more complex AI tasks in deep learning. As a result, researchers are extending deep learning to a variety of different modern fields and tasks in addition to traditional tasks such as object detection, facial recognition or language models.

Wang et al. use deep learning to allow emotional analysis from multiple modalities simultaneously. However, to push the deep learning research frontier from the ground up, it is necessary to thoroughly understand what has been tried in history and why existing models exist in existing forms [37].

Deep learning is a method of learning obtained by converting abstract data at a representative level into nonlinear models. It has the text processing feature found in large data. When processing data, it shrinks the data area, shrinks the dimensions, and helps identify features that are fit for purpose. By reducing data sizes, it deletes unnecessary features in data processing, thus increasing the accuracy of the result and reducing the time [43].

Filtering, mapping, clustering, and Fusion are also among the methods used to extract text properties. Deep learning is also not difficult to learn new data, and it is not difficult to study and draw conclusions about this data. In this way, it is quite fast to get effective results from trained data. Deep learning today has been very successful in processing audio, video, text and images. In addition, emotion analysis, subject categorization, question answering issues such as the accuracy of the results that will not be ignored. It has been a solution to many problems, especially in the field of text classification.

Deep learning has shown very promising results for a variety of tasks in natural language understanding [44], in particular subject classification, sentiment analysis, question answering and language translation [45]. Its deep architectural nature allows deep learning to solve much more complex AI tasks.

Deep learning is a method of learning obtained by converting abstract data at a representative level into nonlinear models. It has the text processing feature found in large data. When processing data, it shrinks the data area, shrinks the dimensions, and helps identify features that are fit for purpose. By reducing data sizes, it deletes unnecessary features in data processing, thus increasing the accuracy of the result and reducing the time. Hinton et al. It was an unsupervised learning classroom in 2006 [46]. Its concept comes from artificial neural network studies. For a multi-layered perceptron, it can be said to have a deep learning structure. Deep learning is the exploration of the distributed feature representation of data by combining it to create more abstract, higher level representing feature classifications or properties. Deep learning, as opposed to a surface learning, now many learning methods are surface structure algorithms, and there are some limitations, such as limited instances of complex function capability, limited ability to generalize for complex classification problem [47].

Figure 3.1 gives examples of the elements of a shape, object area, functions represented by a calculation graph that each node takes in the allowed calculation set.

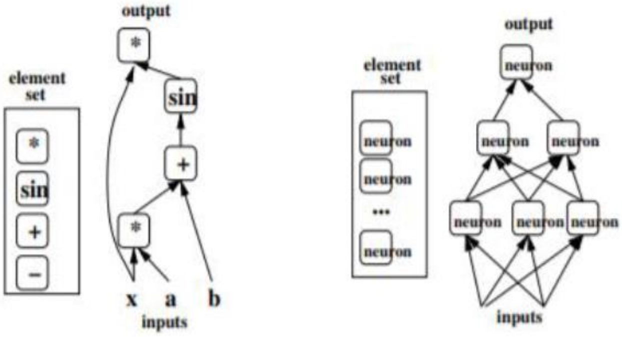


Figure 3.1: An example of functions represented by a measure chart that each node takes in its allowed set of calculations

Frequencies in the dimensions of the text tell us the property of the text. And can be created keywords based on this text feature. Calculating the vector of the text is performed using the weight of these keywords. The proximity distance relationship of these inferred vectors to each other contains important information about how can get the result.

When looking at the difference between deep learning and others, deep learning is the automatic learning feature of big data. In other words, leather learning does not require manual operation as in other learning methods.

Lexical, orthographic (ie. Capitalization, character n-gram), syntactic, semantic and local context (ie. N-gram, adjacent words), discourse level characteristics (ie. Sentence length, chapter headings, writing patterns) and dependency tree-based properties have also been investigated.

Until today, many different methods have been used in the deep learning method and a multi-layer neural network. It shown below in Figure 3.2.

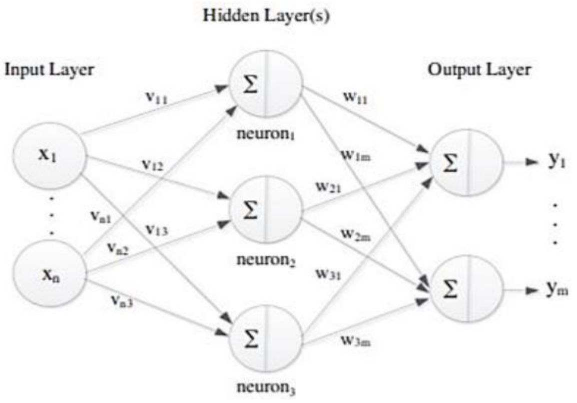


Figure 3.2: Multilayer Neural Network

3.1 Recurrent Neural Network

It is an artificial neural network used to learn time series and sequential series. It is similar to feed forward neural networks. Feedforward networks traditionally map from fixed-size inputs to fixed-size outputs. In RNN, on the other hand, since the input sequences are of variable length, the outputs match variable length lists. It is the result of sharing various parameters and conversions. It is shown below in the Figure 3.3.

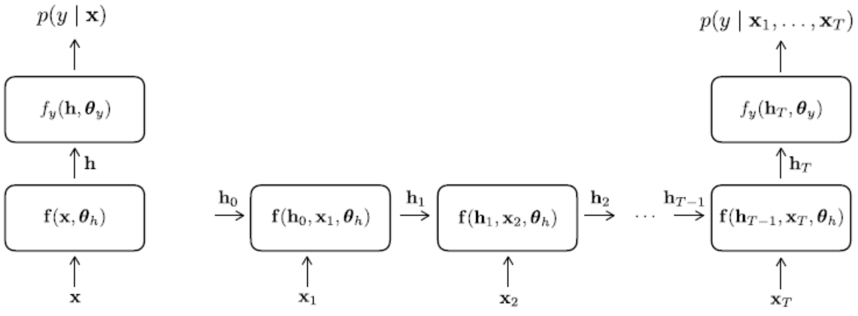

$p(y \mid \mathbf{x})$
$f_y(\mathbf{h}, \boldsymbol{\theta}_y)$
$\mathbf{h}$
$\mathbf{f}(\mathbf{x}, \boldsymbol{\theta}_h)$
$\mathbf{x}$
$\mathbf{h}_0$
$\mathbf{f}(\mathbf{h}_0, \mathbf{x}_1, \boldsymbol{\theta}_h)$
$\mathbf{x}_1$
$\mathbf{h}_1$
$\mathbf{f}(\mathbf{h}_1, \mathbf{x}_2, \boldsymbol{\theta}_h)$
$\mathbf{x}_2$
$\mathbf{h}_2$
$\cdots$
$\mathbf{h}_{T-1}$
$\mathbf{f}(\mathbf{h}_{T-1}, \mathbf{x}_T, \boldsymbol{\theta}_h)$
$\mathbf{x}_T$
$\mathbf{h}_T$
$f_y(\mathbf{h}_T, \boldsymbol{\theta}_y)$
$p(y \mid \mathbf{x}_1, \ldots, \mathbf{x}_T)$

Figure 3.3: Leftside–Feedforward Network / Rightside Recurrent Neural Network

F: function

Θh: parameters both of them shared.

Recurrent structures are separated from feedforward structures because they use their output as input in the next process. That is, recurrent networks have a memory. The reason for adding memory to a network is that the input set that comes in a certain order has a meaning for the output. Feedforward networks are not enough for these kinds of data sets.

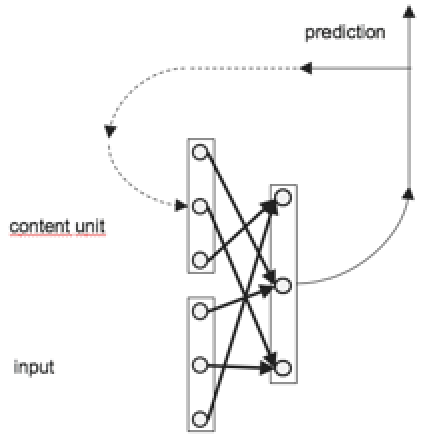


Figure 3.4: RNN Process

As seen in the Figure 3.4, the result from the hidden layer in the RNN operation loop produces both output and written to the content units. In this way, each new input is processed together with the content units produced by processing previous inputs. If there is a correlation between data stored at different times, it is called a "long term" dependency. RNN is a network that can calculate the relationship between these long-term dependencies.

Looking at the example of feedforward neural networks,

$$\mathbf{h} = \tanh(\mathbf{W}_{xh}\mathbf{x}+\mathbf{b}_{h}) \qquad (3.1)$$

$$p(y \mid \mathbf{x}) = \sigma(\mathbf{W}_{hy}\mathbf{h}+ b_{y}) \qquad (3.2)$$

h: result of the hidden layer at time T.

p(y | x): probability of input

x: input

W: weight

b: bias

tanh: hyperbolic tangent function

σ: sigmoid function

Linear transformation used in latent state calculation to depend not only on our x input, but also on information from the past carried in hidden state:

$$\mathbf{h}_0 = \mathbf{0} \quad (3.3)$$

$$\mathbf{h}_1 = \tanh(\mathbf{W}_{hh}\mathbf{h}_0 + \mathbf{W}_{xh}\mathbf{x}_1 + \mathbf{b}_h), \quad (3.4)$$

$$p(y_1 \mid \mathbf{x}_1) = \sigma(\mathbf{W}_{hy}\mathbf{h}_1 + b_y). \quad (3.5)$$

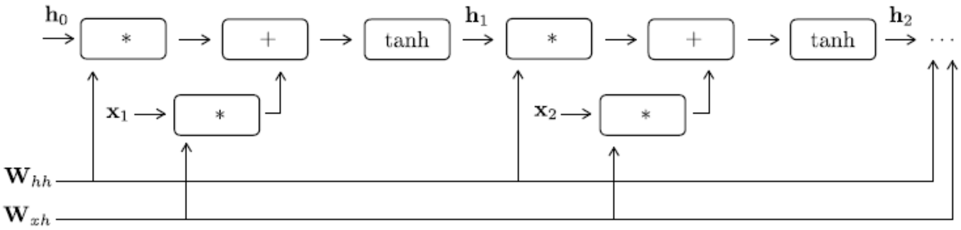


Figure 3.5: A computation graph corresponding to a simple RNN.

*:matrix multiplication

$$\mathrm{h}_2 = \tanh(\mathrm{W}_{hh}\mathrm{h}_1 + \mathrm{W}_{xh}\mathrm{x}_2 + \mathrm{b}_h) \quad (3.6)$$

. . .

. . .

$$\mathrm{h}_T = \tanh(\mathrm{W}_{hh}\mathrm{h}_{T-1} + \mathrm{W}_{xh}\mathrm{x}_T + \mathrm{b}_h) \quad (3.7)$$

$$p(y_T \mid \mathrm{x}_1, \ldots, \mathrm{x}_T) = \sigma(\mathrm{W}_{hy}\mathrm{h}_T + b_y). \quad (3.8)$$

This RNN is capable of processing sequences of any length. In equation 3.3 and 3.4, these are represent one of the first RNN variants. In general formula is,

$$\mathrm{h}_t = \tanh(\mathrm{W}_{hh}\mathrm{h}_{t-1} + \mathrm{W}_{xh}\mathrm{x}_t + \mathrm{b}_h) \quad (3.9)$$

Assume $\mathbf{h}_0 = \mathbf{0}$. Also shown in Figure 3.5.

Simple RNNs map from hidden state to hidden state through in equation 3.9.

Backpropagation Trough Time (BPTT) is purpose of recurrent networks is to classify sequential inputs correctly. In order to do this, backprop of the error and gradient descent are used. Backprop is done by distributing the error in the probe output back to the weights of the error in feedforward networks. By using this derivative, the learning coefficient and gradient descent are arranged to reduce the error.

The method used for RNN is a backprop application for all of the time-sequential sequence calculations known as BPTT. Artificial networks use a number of functions nested f(h(g(x))). When the time dependent variable is added here, derivative operation can be resolved with the chain rule.

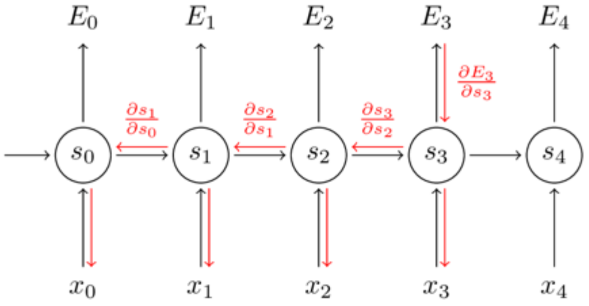

E_0
E_1
E_2
E_3
E_4
$\frac{\partial E_3}{\partial s_3}$
$\frac{\partial s_1}{\partial s_0}$
$\frac{\partial s_2}{\partial s_1}$
$\frac{\partial s_3}{\partial s_2}$
s_0
s_1
s_2
s_3
s_4
x_0
x_1
x_2
x_3
x_4

Figure 3.6: RNN Backpropagation Processing

A recurrent network structure with 5 sequential entries is shown Figure 3.6.

E represents the error that occurred here. For example, when backpropagating for E3, the derivative of W weight is used. Another method is the LSTM method, which is designed to solve this problem.

3.1.1 Long Short Term Mermory (LSTM)

Sepp Hochreiter and Juergen Schmidhuber developed LSTM in 1997 to solve the vanishing gradient problem. LSTM, which was later organized and popularized with the contribution of many people, now has a wide range of uses. Wang et al. Took this method and worked on keyword extraction [49].

LSTM helps to protect the error value coming from different time and layers in the backprop. By providing a more stable error value, it enables the learning steps of recurrent networks to continue. It does this by opening a new channel between cause and effect.

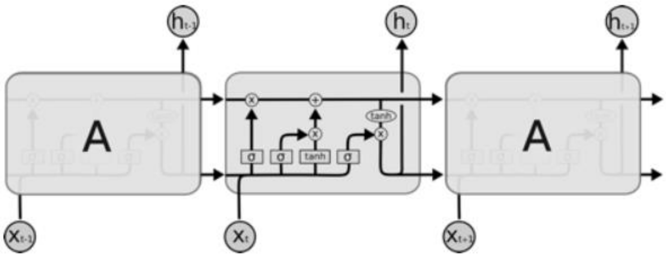


Figure 3.7: LSTM Archticture

The difference of the repeating module in the LSTM structure is that instead of a single neural network layer, there are 4 layers connected in a special way. It shown in Figure 3.7. These layers are also called doors. It is a structure that receives information outside of the normal flow. This information can be stored, written to the cell, and read.

The cell decides what to store, when to read, write or delete, through gates. These gates have a network structure and activation function. Just like neurons, it passes or stops the incoming information according to its weight. These weights are calculated during the learning of the recurrent network. With this structure, the cell learns whether to take or leave or delete data.

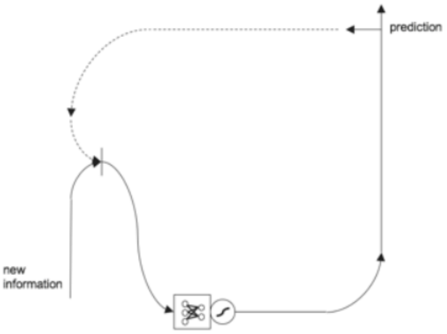


Figure 3.8: RNN

The Figure 3.8 shows the normal RNN operating system. In the Figure 3.9, unlike the first, memory is added to the cell. By doing this, the decision is made using data from a long time ago.

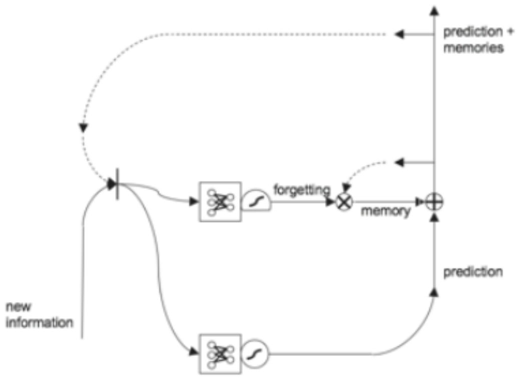


Figure 3.9: RNN Archtitecture via Memory

The + sign seen in the diagram shows the addition process on an element basis. The x sign is an element-based multiplication operation. With the multiplication process performed on an element basis, how much of the data is used or not is calculated by multiplying the data in the memory by the weights. Then, a prediction is produced by summing the coming from the memory and the probability with the + operation.

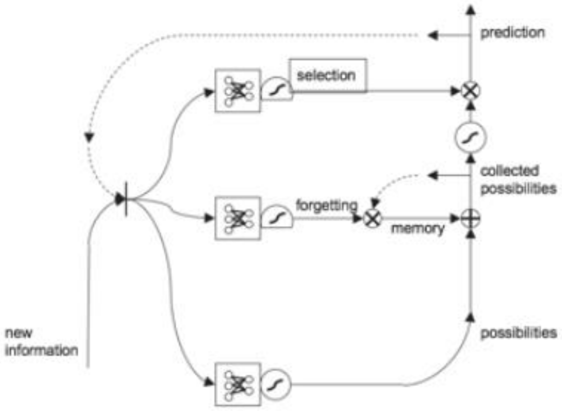


Figure 3.10: RNN Archtitecture via Selection

In Figure 3.10, a new door has been added. After this gate collection process is done, a filter is a section to choose which one to use and how much. In this section, the process of keeping data in memory and separating it from estimation is performed. Again, this door has a unique neural network.

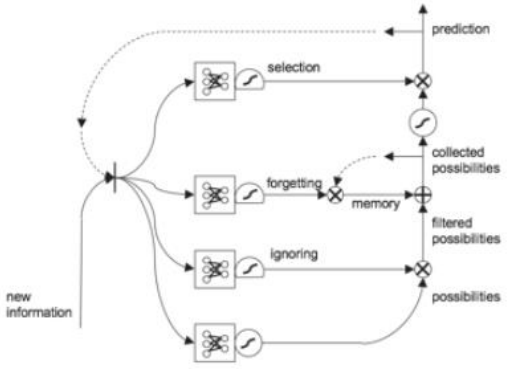
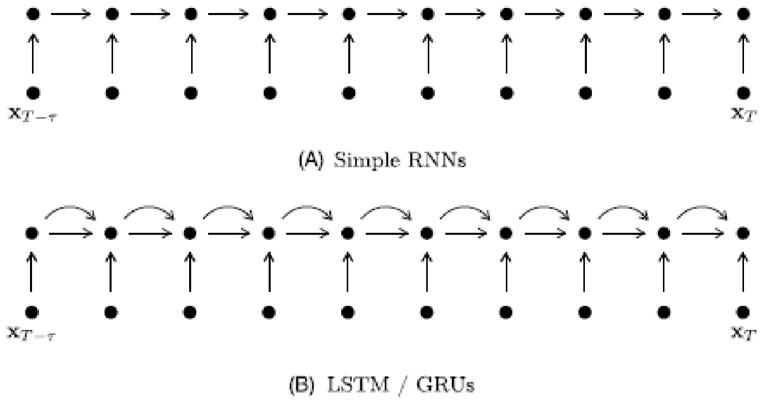


Figure 3.11: RNN Archtitecture via Ignoring

Another port is used for filtering before collecting the first incoming probabilities with the ones coming from memory. Again, there is a network structure here. The incoming results are multiplied on an element basis and pass to the next process. Thus, data that do not need to go to memory are filtered.

LSTM has different models according to different needs. These are provided by the above-mentioned doors taking their inputs from different places or sending their outputs to different places.



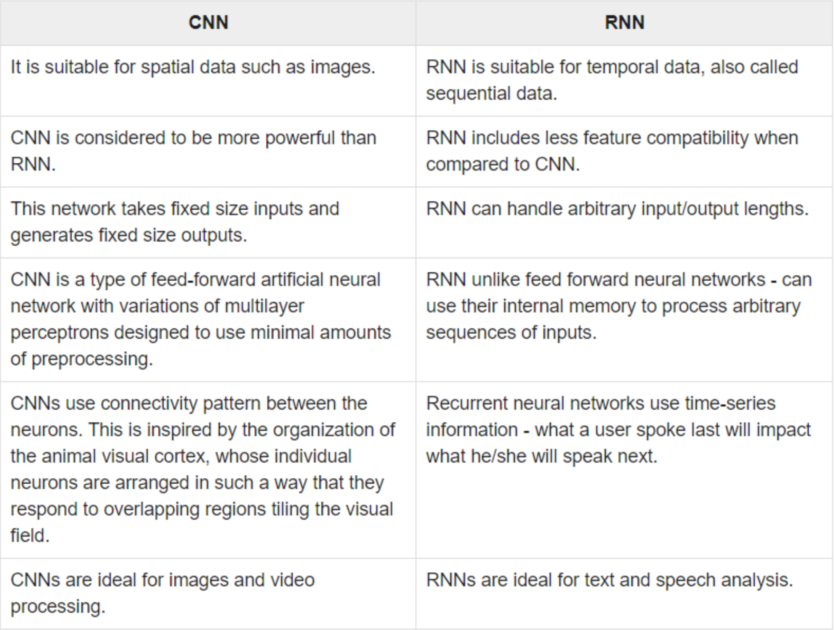
(A) Simple RNNs

(B) LSTM / GRUs

Figure 3.12: Differences between RNN and LSTM

Simple RNNs yield exactly one path between times T and T − τ , with each step inhibited by linear transformations and nonlinearities. LSTM and GRUs give exponentially many paths between times T and T − τ , with one path inhibited by neither linear transformations nor nonlinearities. In Figure 3.12, it shown above. To briefly summarize the difference between RNN and CNN, refer to the Table 3.1.

Table 3.1: Compare of CNN and RNN



CNN	RNN
It is suitable for spatial data such as images.	RNN is suitable for temporal data, also called sequential data.
CNN is considered to be more powerful than RNN.	RNN includes less feature compatibility when compared to CNN.
This network takes fixed size inputs and generates fixed size outputs.	RNN can handle arbitrary input/output lengths.
CNN is a type of feed-forward artificial neural network with variations of multilayer perceptrons designed to use minimal amounts of preprocessing.	RNN unlike feed forward neural networks - can use their internal memory to process arbitrary sequences of inputs.
CNNs use connectivity pattern between the neurons. This is inspired by the organization of the animal visual cortex, whose individual neurons are arranged in such a way that they respond to overlapping regions tiling the visual field.	Recurrent neural networks use time-series information - what a user spoke last will impact what he/she will speak next.
CNNs are ideal for images and video processing.	RNNs are ideal for text and speech analysis.

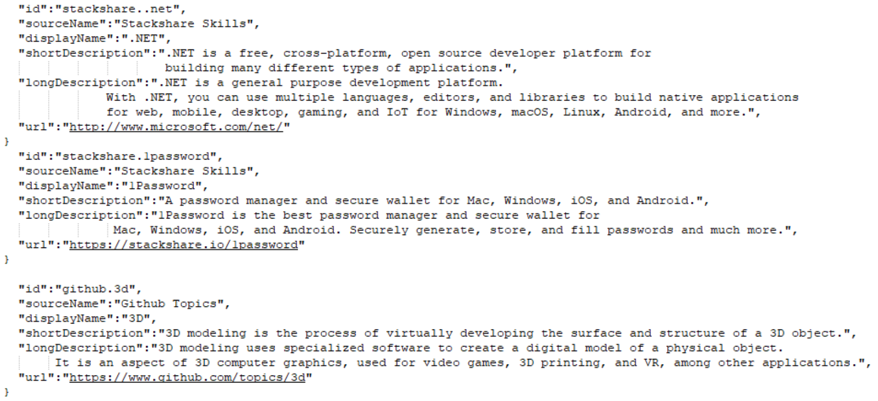
CHAPTER IV

METHODOLOGY AND MATERIALS

4.1 Dataset

The CV corpus subject to the research has been selected from the IT sector and a number of capabilities have been defined for this IT sector. If the corpus consists of only CVs, it was thought that a small dataset would be obtained, and abilities and post-tag were used to obtain a larger dataset and to get more accurate results for vector classification.

A number of talent pool and their definitions were made in the CVs and CVs of the research subject. Online open sources such as Coursera, Microsoft Academic Graph, StackShare Tools, ONET Online Hot Technology Index, ACM Classifications were used for Skills definitions. An example of how definitions are made is given in Figure 4.1.

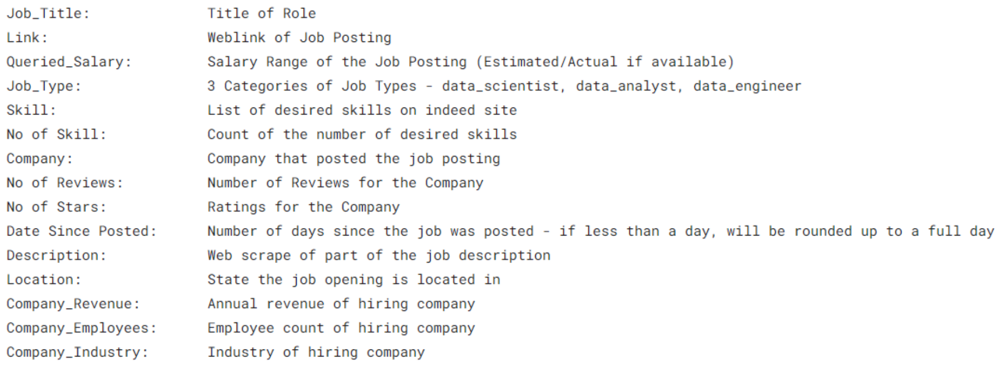


```
  "id":"stackshare..net",
  "sourceName":"Stackshare Skills",
  "displayName":".NET",
  "shortDescription":".NET is a free, cross-platform, open source developer platform for
                    building many different types of applications.",
  "longDescription":".NET is a general purpose development platform.
            With .NET, you can use multiple languages, editors, and libraries to build native applications
            for web, mobile, desktop, gaming, and IoT for Windows, macOS, Linux, Android, and more.",
  "url":"http://www.microsoft.com/net/"
}
  "id":"stackshare.1password",
  "sourceName":"Stackshare Skills",
  "displayName":"1Password",
  "shortDescription":"A password manager and secure wallet for Mac, Windows, iOS, and Android.",
  "longDescription":"1Password is the best password manager and secure wallet for
             Mac, Windows, iOS, and Android. Securely generate, store, and fill passwords and much more.",
  "url":"https://stackshare.io/1password"
}

  "id":"github.3d",
  "sourceName":"Github Topics",
  "displayName":"3D",
  "shortDescription":"3D modeling is the process of virtually developing the surface and structure of a 3D object.",
  "longDescription":"3D modeling uses specialized software to create a digital model of a physical object.
      It is an aspect of 3D computer graphics, used for video games, 3D printing, and VR, among other applications.",
  "url":"https://www.github.com/topics/3d"
}
```

Figure 4.1: Example of Skills for Job

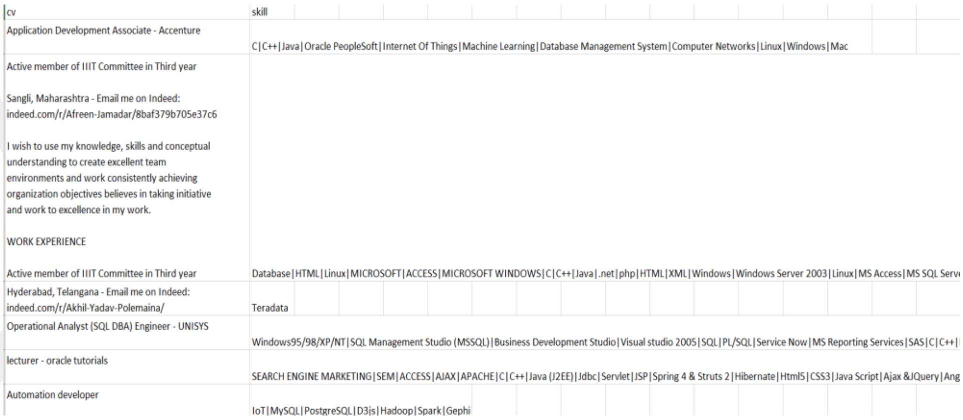
For the required dataset, an IT sector dataset on Kaggle and a number of CVs collected from various sources on the internet were used[50].

Most of the data in the dataset is shown in Figure 4.2 below. Such a dataset has been chosen to get more accurate results.



Job_Title:	Title of Role
Link:	Weblink of Job Posting
Queried_Salary:	Salary Range of the Job Posting (Estimated/Actual if available)
Job_Type:	3 Categories of Job Types - data_scientist, data_analyst, data_engineer
Skill:	List of desired skills on indeed site
No of Skill:	Count of the number of desired skills
Company:	Company that posted the job posting
No of Reviews:	Number of Reviews for the Company
No of Stars:	Ratings for the Company
Date Since Posted:	Number of days since the job was posted - if less than a day, will be rounded up to a full day
Description:	Web scrape of part of the job description
Location:	State the job opening is located in
Company_Revenue:	Annual revenue of hiring company
Company_Employees:	Employee count of hiring company
Company_Industry:	Industry of hiring company

Figure 4.2: Dataset Description[50]



<table>
<tr><th>cv</th><th>skill</th></tr>
<tr><td>Application Development Associate - Accenture</td><td>C|C++|Java|Oracle PeopleSoft|Internet Of Things|Machine Learning|Database Management System|Computer Networks|Linux|Windows|Mac</td></tr>
<tr><td>Active member of IIIT Committee in Third year
Sangli, Maharashtra - Email me on Indeed: indeed.com/r/Afreen-Jamadar/8baf379b705e37c6
I wish to use my knowledge, skills and conceptual understanding to create excellent team environments and work consistently achieving organization objectives believes in taking initiative and work to excellence in my work.
WORK EXPERIENCE
Active member of IIIT Committee in Third year</td><td>Database|HTML|Linux|MICROSOFT|ACCESS|MICROSOFT WINDOWS|C|C++|Java|.net|php|HTML|XML|Windows|Windows Server 2003|Linux|MS Access|MS SQL Serv</td></tr>
<tr><td>Hyderabad, Telangana - Email me on Indeed: indeed.com/r/Akhil-Yadav-Polemaina/</td><td>Teradata</td></tr>
<tr><td>Operational Analyst (SQL DBA) Engineer - UNISYS</td><td>Windows95/98/XP/NT|SQL Management Studio (MSSQL)|Business Development Studio|Visual studio 2005|SQL|PL/SQL|Service Now|MS Reporting Services|SAS|C|C++|</td></tr>
<tr><td>lecturer - oracle tutorials</td><td>SEARCH ENGINE MARKETING|SEM|ACCESS|AJAX|APACHE|C|C++|Java (J2EE)|Jdbc|Servlet|JSP|Spring 4 &amp; Struts 2|Hibernate|Html5|CSS3|Java Script|Ajax &amp;JQuery|Ang</td></tr>
<tr><td>Automation developer</td><td>IoT|MySQL|PostgreSQL|D3js|Hadoop|Spark|Gephi</td></tr>
</table>

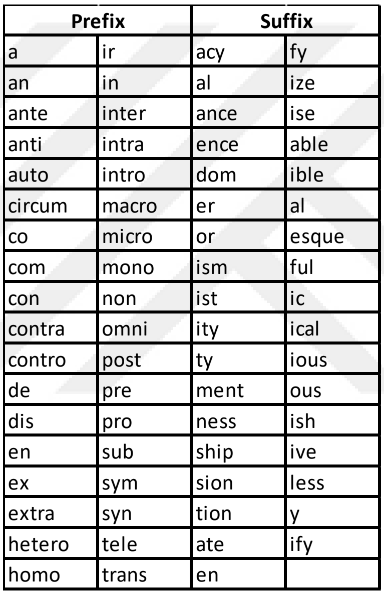
Figure 4.3: Dataset

A section from the formed corpus is shown in Figure 4.3.

4.2 Dataset Pre-Processing Steps

In this section, the steps followed in the study are mentioned. Firstly, started the work by creating the corpus and then cleaned the corpus created with pre-processing. During pre-processing; there are more common but meaningless words such as "and", "the", "or", "only". These are conjunctions or words that are abundant in documents and do not make much sense on their own by the reader. Often such words are removed during data preprocessing. Otherwise, these high-frequency words will not be able to provide the information that is actually intended to be stated in the document in question.

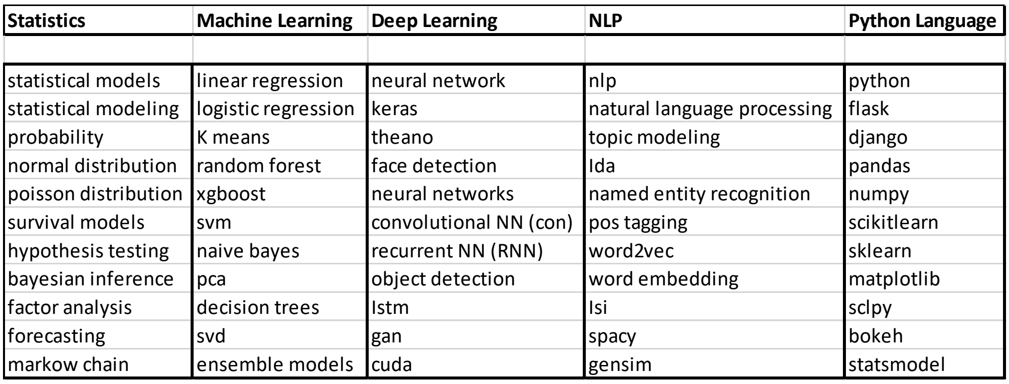
Text preprocessing is a laborious phase of Natural Language Processing, which affects the success of the study, reduces the computational burden in finding the class of the text, and increases the success of the text's class. Conjunctions, exclamations, prepositions, letters and words that do not have meaning should be removed. These words, generally called stop words, should be filtered in the first step in text preprocessing. Later, the punctuation marks in the corpus were removed from the page numbers at the pre-processing stage of the text. In addition, in order to increase the success rate, capitalization in the corpus has been eliminated. Examples of prefix and suffix are shown in the Figure 4.4.



Prefix		Suffix	
a	ir	acy	fy
an	in	al	ize
ante	inter	ance	ise
anti	intra	ence	able
auto	intro	dom	ible
circum	macro	er	al
co	micro	or	esque
com	mono	ism	ful
con	non	ist	ic
contra	omni	ity	ical
contro	post	ty	ious
de	pre	ment	ous
dis	pro	ness	ish
en	sub	ship	ive
ex	sym	sion	less
extra	syn	tion	y
hetero	tele	ate	ify
homo	trans	en	

Figure 4.4: Extracted Prefix and Suffix

If the skills in the resumes are vectorized in this study, they can be matched with much more successful positions and their credibility can be increased. In this way, CVs can be categorized with suitable positions. Therefore, some specific abilities that are generally required for positions are also defined. These are listed in Figure 4.5.



Statistics	Machine Learning	Deep Learning	NLP	Python Language
statistical models	linear regression	neural network	nlp	python
statistical modeling	logistic regression	keras	natural language processing	flask
probability	K means	theano	topic modeling	django
normal distribution	random forest	face detection	lda	pandas
poisson distribution	xgboost	neural networks	named entity recognition	numpy
survival models	svm	convolutional NN (con)	pos tagging	scikitlearn
hypothesis testing	naive bayes	recurrent NN (RNN)	word2vec	sklearn
bayesian inference	pca	object detection	word embedding	matplotlib
factor analysis	decision trees	lstm	lsi	sclpy
forecasting	svd	gan	spacy	bokeh
markow chain	ensemble models	cuda	gensim	statsmodel

Figure 4.5: Abilities and Skills for Position

4.3 Methods

4.2.1 Natural Language Processing

One of the major problems is extracting some structured information from an unstructured text. In general, skills are included in CVs as noun phrases. As long as the skills are mainly present in noun clauses, the extraction process is done by entity recognition performed with the built-in NLTK library methods. The NLTK library is used for Natural language processing.

Natural language processing aims at analyzing and understanding the canonical structure of languages. Natural language processing aims to enable systems to understand spoken language and to speak that language comfortably like a human. Today, the most work on studies are natural language processing studies, especially in parallel with artificial intelligence and machine learning studies.

It is difficult to teach the visual, situational, and auditory experiences people have acquired throughout their lifetime to computers with just writing. Some of these difficulties are; Natural languages have many more words than machine languages. Although there are general rules of natural languages, these rules are often exceeded. It is also very difficult to symbolize words as Vectorals.

With NLP, the relationship between noun clauses and other elements of the sentence can be represented. Thanks to the NLTK library, it can be displayed with representative trees as shown Figure 4.6

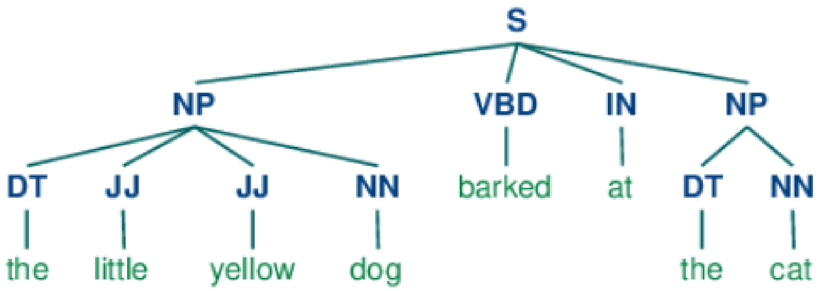


Figure 4.6: An example of NLP

DT: Determiner

JJ: Adjectives

NN: Noun

S: Subject

NP: Noun Phrase

VBD: Verb, non-3rd person singular present

IN: Preposition or subordinating conjunction

Thanks to the NLTK library, a sentence can be defined as a series of adjectives and nouns. Or, NLTK can be taught how to label a string of sentences. In this way, it can be seen whether the correct target skills are available for expression.

4.2.2 Word2Vec

Word2Vec is a two-layer neural network that processes text by vectorizing words. Its input is a collection of text and its output is a set of vectors. The purpose of Word2Vec is to group vectors of similar words together in vector field.

Two different algorithms for creating word vectors in Word2Vec are available. These are Skip-Gram and Continious Bag of Words(CBOW).

4.2.2.1 Skip Gram

Loops on the existing corpus for the words in each sentence or uses the words in the middle to guess the words next to it. This method is called Skip-Gram. Uses all the sentences available for the current word predict. This method is called Continious Bag of Words (CBOW). The word count limit in each context is determined by a parameter called "window size".

After training a simple neural network with a single hidden layer to perform a specific task, it is actually the weight of the hidden layer used. In the meaning of these weights, our aim is to find the vectors. If one of the words to the right or left of the word is selected in a sentence, the model can show the probability that the selected word is a close word by looking at the existing corpus. To do this, the neural network is trained by feeding word pairs found in training documents.

It can be shown with an example using the small window size of two in the Figure 4.7. The blue is input word.

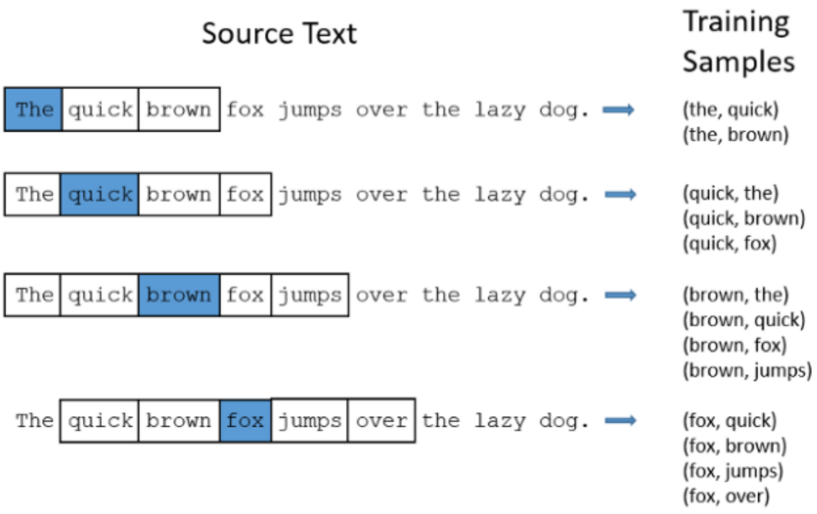


Figure 4.7: An example of Skip-Gram

To explain with an another example; Let's choose "ants" as the input word. There will be one component for each word in our vocabulary. It is continued by writing 1 in the position corresponding to the "ants", that is, the input word. The output of the network is a single vector for each word in our vocabulary containing the probability that a randomly chosen word is that word. It shown in Figure 4.8.

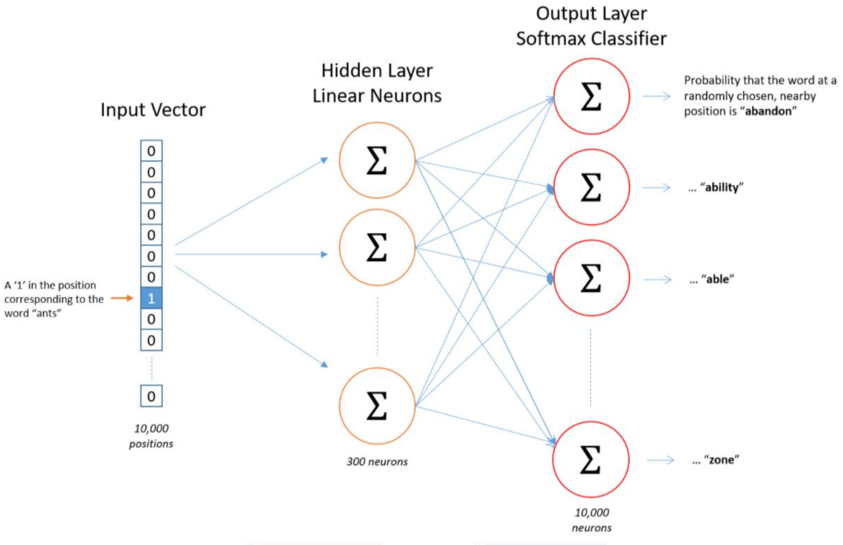

Input Vector
0
0
0
0
0
0
0
1
0
0
0
10,000 positions
A '1' in the position corresponding to the word "ants"
Hidden Layer Linear Neurons
Σ
Σ
Σ
300 neurons
Output Layer Softmax Classifier
Σ
Σ
Σ
Σ
10,000 neurons
Probability that the word at a randomly chosen, nearby position is "abandon"
… "ability"
… "able"
… "zone"

Figure 4.8: Architecture of Neural Network

Therefore, the hidden layer will be represented by a weight matrix containing 10,000 rows (one for every word in our dictionary) and 300 columns (one for each hidden neuron). The number of features is a "hyper parameter" that you should set in your application. The best valuation efforts for Features are still ongoing today. The weight matrix will actually be our word vectors. An example is shown in the Figure 4.9.

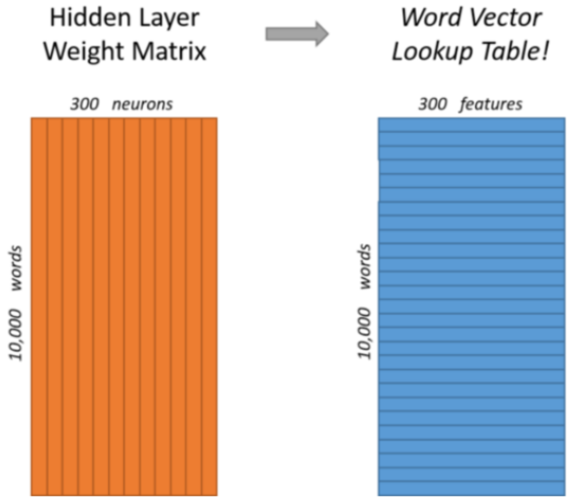


Figure 4.9: Weight Matrix and Word Vector

As it was said at the beginning, hidden layer weight matrix is what you want to learn. A 1x300 word vector is created for "ants". There is a weight vector corresponding to the word vector in the hidden layer and exp (x) is applied to this vector. For the sum to be 1, It is divided into the vocabulary owned. The stronger the links between the two words, the more the model gives similar results for these two words. This conclusion follows vectors will be similar because the word. The Skip-Gram algorithm can be explained as follows:

- Generate word vector, x.
- Word vectors for the text $\vartheta_c = \vartheta_x$
- Generate 2m score vectors, $\vartheta_{c-m} + \vartheta_{c-m+1} + \cdots + \vartheta_{c+m}$ via u= $u = U\vartheta_c$
- Transforming the result into probability: y = softmax(u)
- Generated vector matching true probabilities $y^{(c-m)}, \dots.. y^{(c-1)}, y^{(c+1)}, \dots y^{(c+m)}$ fort he actual output vectors.

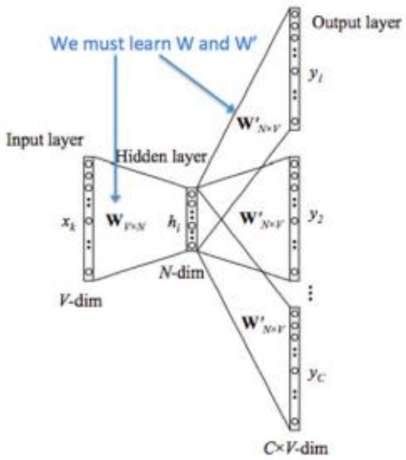


Figure 4.10: Skip-Gram Architecture

Skip-Gram algorithm architecture is shown in the Figure 4.10 above.

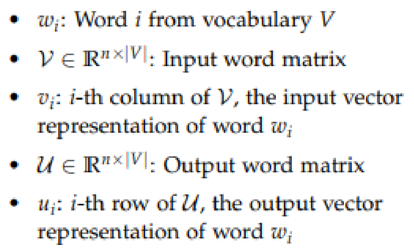
4.2.2.2 Continious Bag of Words

In the CBOW model, in order to guess the middle word for the sentence the words to the right or left are reconciled. It predicts a center word from the surrounding context. The CBOW algorithm can be explained as follows:

- Generate word vectors $(x^{(c-m)}, \dots.. x^{(c-1)}, x^{(c+1)}, \dots x^{(c+m)})$ for the input context of size m.

- Word vectors for the text ($\vartheta_{c-m} = \vartheta_{x^{(c-m)}}, \vartheta_{c-m+1} = \vartheta_{x^{(c-m+1)}}, \ldots\ \vartheta_{c+m} = \vartheta_{x^{(c+m)}}$)
- Avarage of vectors = $\vartheta = x = \frac{\vartheta_{c-m} + \vartheta_{c-m+1} + \cdots + \vartheta_{c+m}}{2m}$
- Find score vector = $z = \upsilon\vartheta$
- Transforming the result into probability: y = softmax(z) where y is vector of actual word.

Where it represents;



- w_i: Word i from vocabulary V
- $\mathcal{V} \in \mathbb{R}^{n\times|V|}$: Input word matrix
- v_i: i-th column of $\mathcal{V}$, the input vector representation of word w_i
- $\mathcal{U} \in \mathbb{R}^{n\times|V|}$: Output word matrix
- u_i: i-th row of $\mathcal{U}$, the output vector representation of word w_i

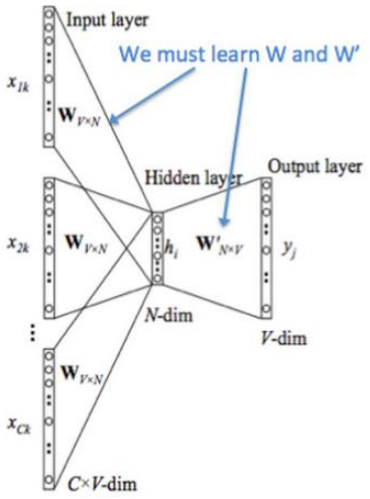


Figure 4.11 CBOW Archtitecture

CBOW algorithm architecture is shown in the Figure 4.11 above.

When in the skip-gram model, the distributed representation of the input word is used to predict context. Comparison of Cbow and Skip-Gram is in the Figure 4.12.

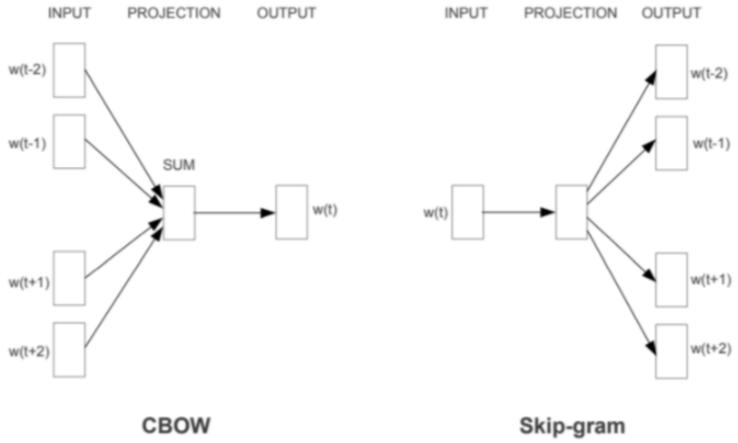


Figure 4.12: CBOW and Skip-Gram Architecture

4.2.3 N-Gram

N-Gram is a method used to find the repetition rate in a given sequence. Its name is a combination of the words n and gram. Here n is the value by which repetition is checked. Gram is used to express the weight of this repeat within the sequence. Items can be syllables, letters, words or base pairs, depending on the application. The n-gram is called "unigram"; size two is a "bigram". Size three is a "trigram" also called.

For example, “statistics” is a unigram (n = 1), “machine learning” is a bigram (n = 2), “natural language processing” is a trigram (n = 3), and so on.

Given Sentence : “i have a dream [END]”.

Estimation: P(i)

P(have | i)

P(a | i have)

P(dream | i have a)

P([END] | i have a dream)

The probability of each word is independent of any previous word. Training the model is nothing more than calculating these fractions for all unigrams in the training text.

$$P_{\text{train}}(\text{dream} \mid \cancel{\text{i have a}}) = P_{\text{train}}(\text{dream}) = \frac{n_{\text{train}}(\text{dream})}{\underset{\substack{\uparrow \\ \text{total number of words in} \\ \text{training text}}}{N_{\text{train}}}}$$

After evaluating all the probabilities of the unigrams, These estimates can be applied to calculate the probability of each sentence in the evaluated text: the probability of each sentence is the product of the word probabilities.

$$\begin{aligned} P_{\text{eval}}(\text{i have a dream [END]}) &= P_{\text{train}}(\text{i})\, P_{\text{train}}(\text{have} \mid \text{i})\, P_{\text{train}}(\text{a} \mid \text{i have})\, P_{\text{train}}(\text{dream} \mid \text{i have a})\, P_{\text{train}}(\text{[END]} \mid \text{i have a dream}) \\ &= P_{\text{train}}(\text{i})\, P_{\text{train}}(\text{have})\, P_{\text{train}}(\text{a})\, P_{\text{train}}(\text{dream})\, P_{\text{train}}(\text{[END]}) \end{aligned}$$

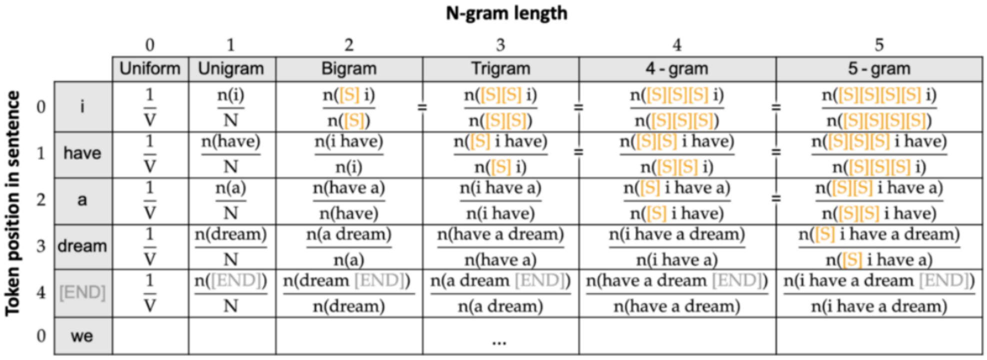
Products of all n-gram probabilities:

$$\begin{aligned} \text{Evaluation text:} &\quad \text{"I have a dream. We are free at last."} \\ P_{\text{eval}}(\text{text}) &= P_{\text{eval}}(\text{i have a dream [END]})\, P_{\text{eval}}(\text{we are free at last [END]}) \\ P_{\text{eval}}(\text{i have a dream [END]}) &= P_{\text{train}}(\text{i} \mid \text{[S][S]})\, P_{\text{train}}(\text{have} \mid \text{[S] i})\, P_{\text{train}}(\text{a} \mid \text{i have})\, P_{\text{train}}(\text{dream} \mid \text{have a})\, P_{\text{train}}(\text{[END]} \mid \text{a dream}) \\ P_{\text{eval}}(\text{we are free at last [END]}) &= P_{\text{train}}(\text{we} \mid \text{[S][S]})\, P_{\text{train}}(\text{are} \mid \text{[S] we})\, P_{\text{train}}(\text{free} \mid \text{we are})\, P_{\text{train}}(\text{at} \mid \text{are free})\, P_{\text{train}}(\text{last} \mid \text{free at})\, P_{\text{train}}(\text{[END]} \mid \text{at last}) \\ \hline \Longrightarrow P_{\text{eval}}(\text{text}) &= P_{\text{train}}(\text{i} \mid \text{[S][S]})\, P_{\text{train}}(\text{have} \mid \text{[S] i})\, P_{\text{train}}(\text{a} \mid \text{i have})\, P_{\text{train}}(\text{dream} \mid \text{have a})\, P_{\text{train}}(\text{[END]} \mid \text{a dream}) \\ &\quad P_{\text{train}}(\text{we} \mid \text{[S][S]})\, P_{\text{train}}(\text{are} \mid \text{[S] we})\, P_{\text{train}}(\text{free} \mid \text{we are})\, P_{\text{train}}(\text{at} \mid \text{are free})\, P_{\text{train}}(\text{last} \mid \text{free at})\, P_{\text{train}}(\text{[END]} \mid \text{at last}) \end{aligned}$$

Use the average log likelihood as the evaluation metric for the n-gram model. The better our n-gram model is, the probability that it assigns to each word in the evaluation text will be higher on average.

$$\begin{aligned} P_{\text{eval}}(\text{text}) &= \prod_{\text{word}} P_{\text{train}}(\text{word}) \\ \log(P_{\text{eval}}(\text{text})) &= \sum_{\text{word}} \log(P_{\text{train}}(\text{word})) \\ \text{Average log likelihood}_{\text{eval}} &= \frac{\sum_{\text{word}} \log(P_{\text{train}}(\text{word}))}{\underset{\substack{\uparrow \\ \text{total number of words in} \\ \text{evaluation text}}}{N_{\text{eval}}}} \end{aligned}$$

Example of N-Grams shown in the Figure 4.13.



N-gram length

Token position in sentence		0	1	2	3	4	5
		Uniform	Unigram	Bigram	Trigram	4 - gram	5 - gram
0	i	$\frac{1}{V}$	$\frac{\text{n(i)}}{\text{N}}$	$\frac{\text{n([S] i)}}{\text{n([S])}}$ =	$\frac{\text{n([S][S] i)}}{\text{n([S][S])}}$ =	$\frac{\text{n([S][S][S] i)}}{\text{n([S][S][S])}}$ =	$\frac{\text{n([S][S][S][S] i)}}{\text{n([S][S][S][S])}}$
1	have	$\frac{1}{V}$	$\frac{\text{n(have)}}{\text{N}}$	$\frac{\text{n(i have)}}{\text{n(i)}}$	$\frac{\text{n([S] i have)}}{\text{n([S] i)}}$ =	$\frac{\text{n([S][S] i have)}}{\text{n([S][S] i)}}$ =	$\frac{\text{n([S][S][S] i have)}}{\text{n([S][S][S] i)}}$
2	a	$\frac{1}{V}$	$\frac{\text{n(a)}}{\text{N}}$	$\frac{\text{n(have a)}}{\text{n(have)}}$	$\frac{\text{n(i have a)}}{\text{n(i have)}}$	$\frac{\text{n([S] i have a)}}{\text{n([S] i have)}}$ =	$\frac{\text{n([S][S] i have a)}}{\text{n([S][S] i have)}}$
3	dream	$\frac{1}{V}$	$\frac{\text{n(dream)}}{\text{N}}$	$\frac{\text{n(a dream)}}{\text{n(a)}}$	$\frac{\text{n(have a dream)}}{\text{n(have a)}}$	$\frac{\text{n(i have a dream)}}{\text{n(i have a)}}$	$\frac{\text{n([S] i have a dream)}}{\text{n([S] i have a)}}$
4	[END]	$\frac{1}{V}$	$\frac{\text{n([END])}}{\text{N}}$	$\frac{\text{n(dream [END])}}{\text{n(dream)}}$	$\frac{\text{n(a dream [END])}}{\text{n(a dream)}}$	$\frac{\text{n(have a dream [END])}}{\text{n(have a dream)}}$	$\frac{\text{n(i have a dream [END])}}{\text{n(i have a dream)}}$
0	we				...		

Figure 4.13: N-Gram Arthitecture

V: number of unique unigrams in training text.

N: total number of tokens in training text.

Equal signs: identical starting probabilities.

CHAPTER V

ANALYSIS

The main purpose here is to distinguish skills on CVs and make a smoother classification. In order to reach the result, the sentences in the training set should be formed accordingly. In order to create a labeled training set and to understand whether it has skills or not, a corpus has been created in this way. Our main goal is to reveal skills that are not visible or overlooked on CVs.

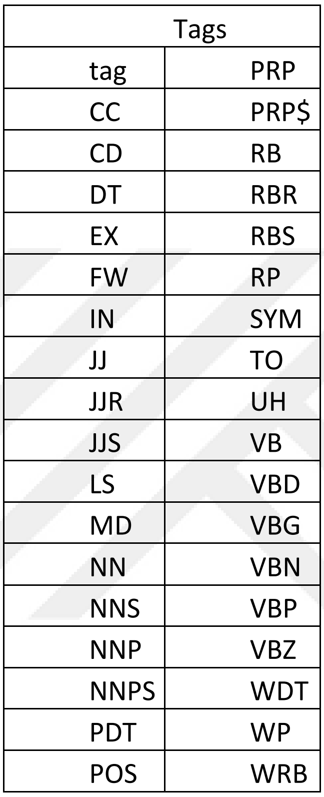
For the vector of each word; When defining skills in CVs, it may not contain only letters or only numbers. All can be written in large or small. For example, while a developer creates his own skill in his CV, the basic skills he will write: "Machine Learning - Python3". This example can also be considered as "SQL" as a skill for database administrator. It is also an important detail whether the words in the training set are in the English vocabulary or not.

Vector operations for data can be summarized as follows;

- Preprocesses string data(phrases, context and/or skills) into vectors for neural network input.
- Transforms words into vectors (Word vectorising based on representing presence or absence of multiple binary features) and then into matrix.
- Returns concatenated vector of maximal and minimal features for given phrase vectors and context vectors.
- Main logic that for phrases, context and/or skills returns its vector values.

Grammatical rules are not followed when preparing a CV. One can use inverted sentences to emphasize one's own abilities. It can begin a sentence with a predicate, not a subject. This is actually against grammatical structure. Because many names and words have specific meanings, they can distort the structure, which in turn disrupts semantically integrity and grammatical structure. The NLTK library actually makes some mistakes in this regard. Thanks to NLTK Part of Speech tagger, this process can be done, but it can make mistakes while tagging CV sentences.

For this reason, many Part of Speech TAGs have been introduced to the system as a result of research on the internet for dataset. It can be seen examples in the Figure 5.1 below.



Tags	
tag	PRP
CC	PRP$
CD	RB
DT	RBR
EX	RBS
FW	RP
IN	SYM
JJ	TO
JJR	UH
JJS	VB
LS	VBD
MD	VBG
NN	VBN
NNS	VBP
NNP	VBZ
NNPS	WDT
PDT	WP
POS	WRB

Figure 5.1: Pos Tags

Classification is accomplished by a Keras neural network with three input, each designed to receive specific data classes. The first input takes a variable-length vector in which defined various properties of sentences containing various numbers of words and can take these properties. This created vector is processed with the LSTM layer.

The vector of the second variable contains the structure and content of the sentence. For the candidate sentence, the window size value should be selected, and for the candidate sentence, the words for this value are selected both to the right and to the left of the candidate sentence. The statement in bold represents the candidate statement. All of them express the content in Figure 5.2.

Sentence:	Responsible	For	**Natural**	**Language**	**Processing**	X	X
	Words Embedding						

Figure 5.2:Word Embedding

The third variable vector, general information about candidate word processing as about content structure and content itself. In addition, it compares the min and max values of vectors between sentences and their contexts. Represent the presence or absence of many binary features in the whole phrase.

Model consists of three input:

- First (LSTM) takes variable length vector of arbitrary number of words (phrase).
- Second (LSTM) takes context of a phrase. Variable length vector of a phrase and n-words to the right and left of a phrase.
- Third (DENSE) takes fixed size vector representing presence or absence of binary features.

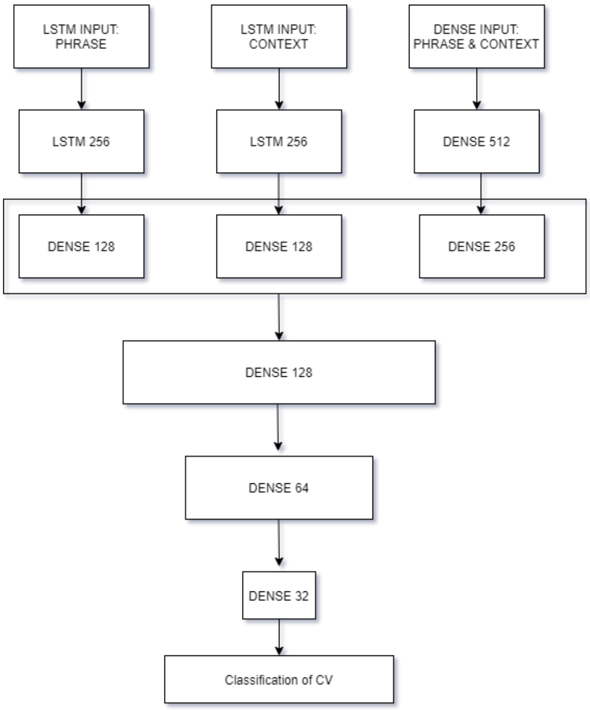
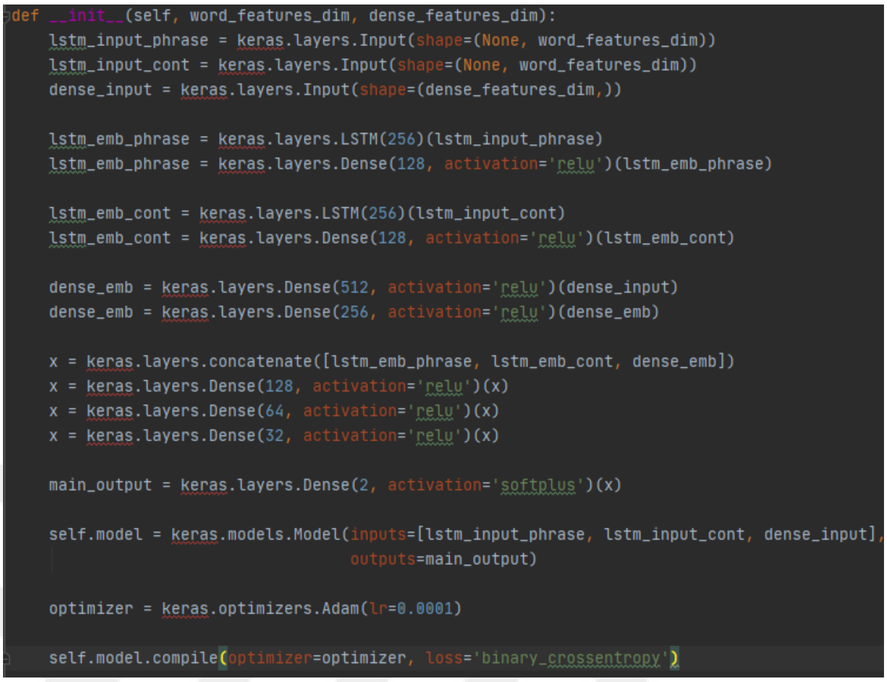


Figure 5.3: Neural Network Archtitecture

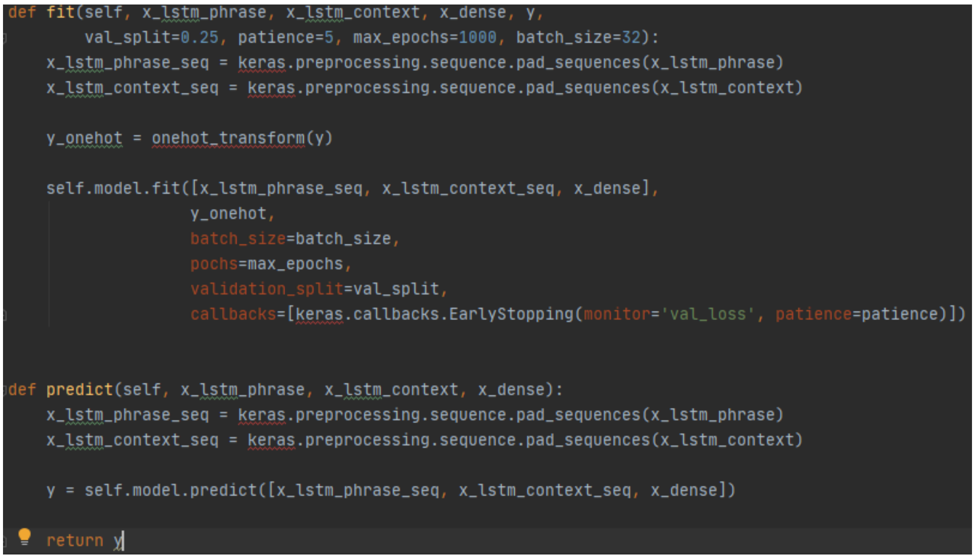
How the system architecture described in Figure 5.3 is implemented is shown in Figure 5.4 below.



```
def __init__(self, word_features_dim, dense_features_dim):
    lstm_input_phrase = keras.layers.Input(shape=(None, word_features_dim))
    lstm_input_cont = keras.layers.Input(shape=(None, word_features_dim))
    dense_input = keras.layers.Input(shape=(dense_features_dim,))

    lstm_emb_phrase = keras.layers.LSTM(256)(lstm_input_phrase)
    lstm_emb_phrase = keras.layers.Dense(128, activation='relu')(lstm_emb_phrase)

    lstm_emb_cont = keras.layers.LSTM(256)(lstm_input_cont)
    lstm_emb_cont = keras.layers.Dense(128, activation='relu')(lstm_emb_cont)

    dense_emb = keras.layers.Dense(512, activation='relu')(dense_input)
    dense_emb = keras.layers.Dense(256, activation='relu')(dense_emb)

    x = keras.layers.concatenate([lstm_emb_phrase, lstm_emb_cont, dense_emb])
    x = keras.layers.Dense(128, activation='relu')(x)
    x = keras.layers.Dense(64, activation='relu')(x)
    x = keras.layers.Dense(32, activation='relu')(x)

    main_output = keras.layers.Dense(2, activation='softplus')(x)

    self.model = keras.models.Model(inputs=[lstm_input_phrase, lstm_input_cont, dense_input],
                                    outputs=main_output)

    optimizer = keras.optimizers.Adam(lr=0.0001)

    self.model.compile(optimizer=optimizer, loss='binary_crossentropy')
```

Figure 5.4: Implementation of Extraction for Resume

Various optimizer trials have been made to avoid overfitting during model training. The created vectors are classified as [1,0] or [0,1], and the loss function used belongs to the keras library. Fit method has been applied to be more useful. The aim is to construct the feature vector for the candidate expression. Automatic stopping is aimed with cross validation and prediction methods. It shown in Figure 5.5 below. The arrangement required for input in the LSTM layers has been brought to the equal width 2D array format.



```
def fit(self, x_lstm_phrase, x_lstm_context, x_dense, y,
        val_split=0.25, patience=5, max_epochs=1000, batch_size=32):
    x_lstm_phrase_seq = keras.preprocessing.sequence.pad_sequences(x_lstm_phrase)
    x_lstm_context_seq = keras.preprocessing.sequence.pad_sequences(x_lstm_context)

    y_onehot = onehot_transform(y)

    self.model.fit([x_lstm_phrase_seq, x_lstm_context_seq, x_dense],
                   y_onehot,
                   batch_size=batch_size,
                   pochs=max_epochs,
                   validation_split=val_split,
                   callbacks=[keras.callbacks.EarlyStopping(monitor='val_loss', patience=patience)])


def predict(self, x_lstm_phrase, x_lstm_context, x_dense):
    x_lstm_phrase_seq = keras.preprocessing.sequence.pad_sequences(x_lstm_phrase)
    x_lstm_context_seq = keras.preprocessing.sequence.pad_sequences(x_lstm_context)

    y = self.model.predict([x_lstm_phrase_seq, x_lstm_context_seq, x_dense])

    return y
```

Figure 5.5: Implementation of Prediction Method

Since the number of words in the corpus given as input is not known, defining a fixed length vector system may cause errors. Therefore, an artificial neural network structure that can process vectors used in various lengths has been established. With the LSTM method, higher accuracy results were obtained.

Several architectures have been tested with different combinations of dense layers with LSTM ones. The resulting architecture configuration (the size and number of the layers) showed the best results on cross-validation test which corresponds to the optimal usage of training data. Enlarging the data set will always increase the accuracy on the model, so the layer size and number can be tested differently.

The basic model of the system is described in the Figure 5.6 below.

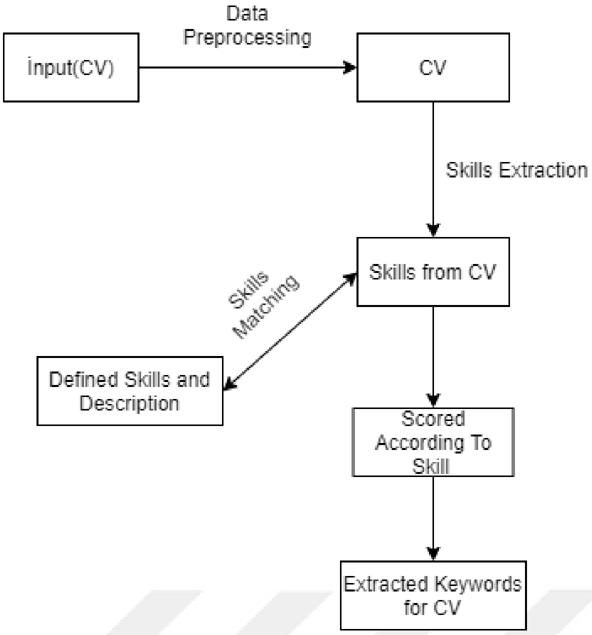

Data Preprocessing
İnput(CV)
CV
Skills Extraction
Skills from CV
Skills Matching
Defined Skills and Description
Scored According To Skill
Extracted Keywords for CV

Figure 5.6: System Architecture

For the keywords that the end user searches in the CV, the CV is entered into the system. Thanks to the model for Input CV, Keyword extraction is aimed. Thanks to the skills obtained from the CV and the skills and description previously defined in the system, the skills in the CV were scored and the skills with the best scores were brought to the fore. A keyword has been suggested for the skill with the highest score.

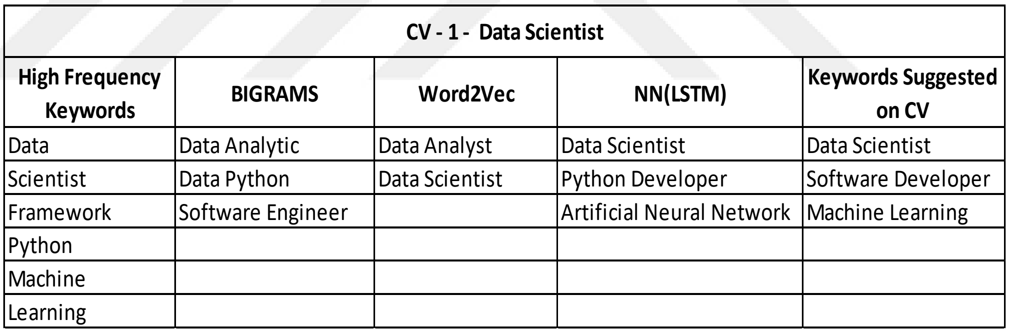
CHAPTER VI

RESULT

In order to indicate the results of the studies conducted, the results of keyword samples suggested on the CV are given. For each CV processed in the given tables, it includes the keywords given by the CV author and the high frequency words founded as a result of study. The keywords resulting from the comparison with Word2Vec, Bi-Grams and NN(LSTM) are compared.

The suggested keywords are given by working on the CV titled Data Scientist in Table 6.1.

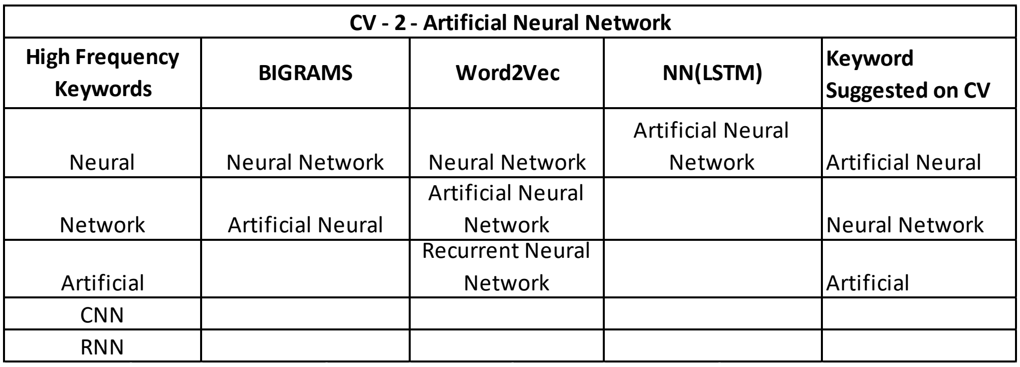
Table 6.1: Extracted Keywords for Data Scientist CV.



CV - 1 - Data Scientist				
High Frequency Keywords	**BIGRAMS**	**Word2Vec**	**NN(LSTM)**	**Keywords Suggested on CV**
Data	Data Analytic	Data Analyst	Data Scientist	Data Scientist
Scientist	Data Python	Data Scientist	Python Developer	Software Developer
Framework	Software Engineer		Artificial Neural Network	Machine Learning
Python				
Machine				
Learning				

The suggested keywords are given by working on the CV titled Artificial Neural Network in Table 6.2.

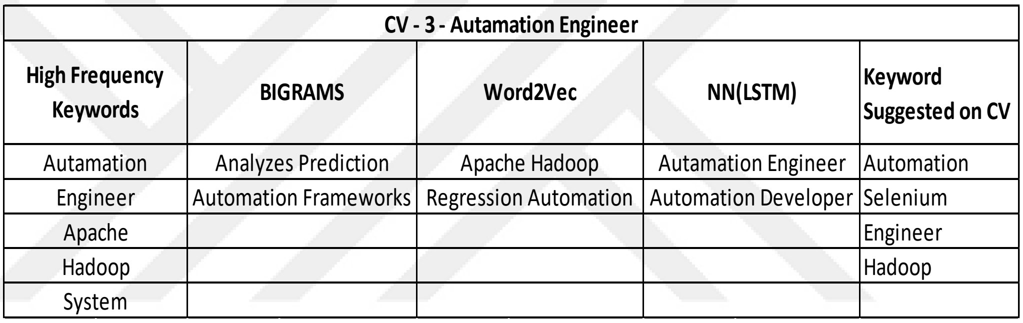
Table 6.2: Extracted Keywords for ANN CV.



CV - 2 - Artificial Neural Network				
High Frequency Keywords	**BIGRAMS**	**Word2Vec**	**NN(LSTM)**	**Keyword Suggested on CV**
Neural	Neural Network	Neural Network	Artificial Neural Network	Artificial Neural
Network	Artificial Neural	Artificial Neural Network		Neural Network
Artificial		Recurrent Neural Network		Artificial
CNN				
RNN				

The suggested keywords are given by working on the CV titled Automation Engineer in Table 6.3.

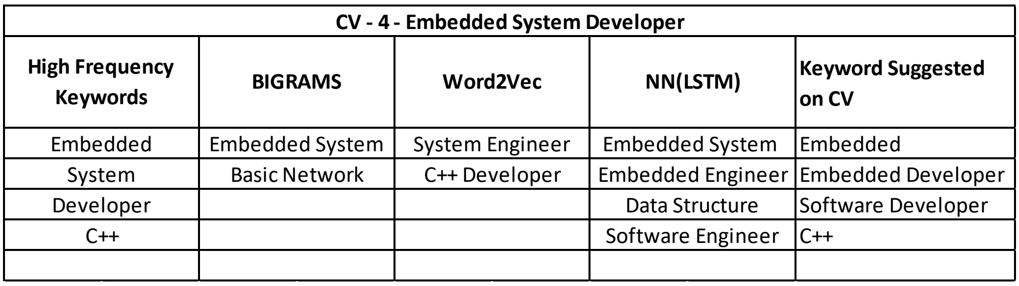
Table 6.3: Extracted Keywords for Automation Engineer CV.



CV - 3 - Autamation Engineer				
High Frequency Keywords	**BIGRAMS**	**Word2Vec**	**NN(LSTM)**	**Keyword Suggested on CV**
Autamation	Analyzes Prediction	Apache Hadoop	Autamation Engineer	Automation
Engineer	Automation Frameworks	Regression Automation	Automation Developer	Selenium
Apache				Engineer
Hadoop				Hadoop
System				

The suggested keywords are given by working on the CV titled System Developer in Table 6.4.

Table 6.4: Extracted Keywords for Embedded System Developer CV.



CV - 4 - Embedded System Developer				
High Frequency Keywords	**BIGRAMS**	**Word2Vec**	**NN(LSTM)**	**Keyword Suggested on CV**
Embedded	Embedded System	System Engineer	Embedded System	Embedded
System	Basic Network	C++ Developer	Embedded Engineer	Embedded Developer
Developer			Data Structure	Software Developer
C++			Software Engineer	C++

The suggested keywords are given by working on the CV titled Frontend Developer in Table 6.5.

Table 6.5: Extracted Keywords for Frontend Developer CV.

CV - 5 - FrontEnd Developer				
High Frequency Keywords	**BIGRAMS**	**Word2Vec**	**NN(LSTM)**	**Keyword Suggested on CV**
Frontend	Javascript development	React and Angular	FrontEnd Developer	Frontend
Developer	Development and	Javascript Solution	Javascript and CSS	Javascript
Javascript	Traning learning		Restful API	CSS
CSS				
Services				

The suggested keywords are given by working on the CV titled Test Engineer in Table 6.6.

Table 6.6: Extracted Keywords for Test Engineer CV.

CV - 6- Test Engineer				
High Frequency Keywords	**BIGRAMS**	**Word2Vec**	**NN(LSTM)**	**Keyword Suggested on CV**
Test	Experienced manuel	Web Application	Test Engineering	Testing
Oracle	Writing Test	Test Engineer	Automation Testing	Developer
Analyst			Functional Testing	Automation
Testing				Selenium

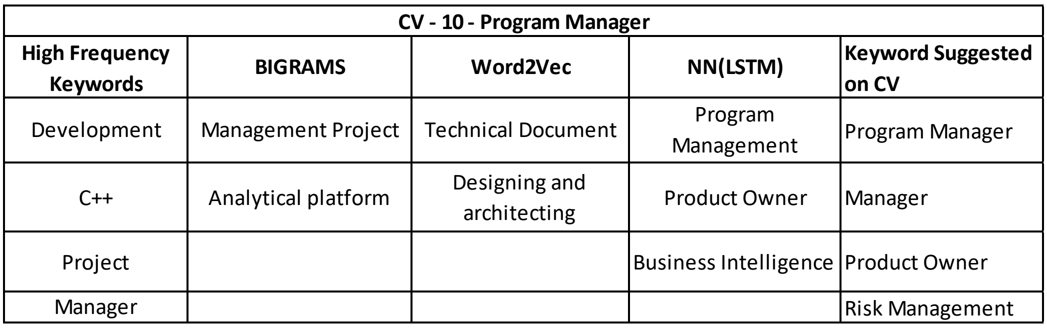
The suggested keywords are given by working on the CV titled Technical Architecture in Table 6.7.

Table 6.7: Extracted Keywords for Technical Architecture CV.

CV - 7 - Technical Architecture				
High Frequency Keywords	**BIGRAMS**	**Word2Vec**	**NN(LSTM)**	**Keyword Suggested on CV**
Technical	Working Microsoft	Technical Document	Software Architecture	
Architecture	Design and	Designing and architecting	Design and Develop	
Design	And Architecting		Technical Architecture	
System				
Microsoft				

The suggested keywords are given by working on the CV titled Program Manager in Table 6.8.

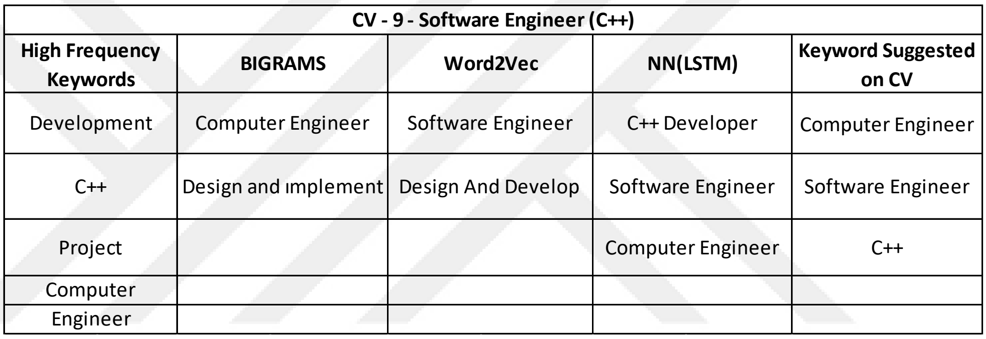
Table 6.8: Extracted Keywords for Program Manager CV.



CV - 10 - Program Manager				
High Frequency Keywords	**BIGRAMS**	**Word2Vec**	**NN(LSTM)**	**Keyword Suggested on CV**
Development	Management Project	Technical Document	Program Management	Program Manager
C++	Analytical platform	Designing and architecting	Product Owner	Manager
Project			Business Intelligence	Product Owner
Manager				Risk Management

The suggested keywords are given by working on the CV titled Software Engineer in Table 6.9.

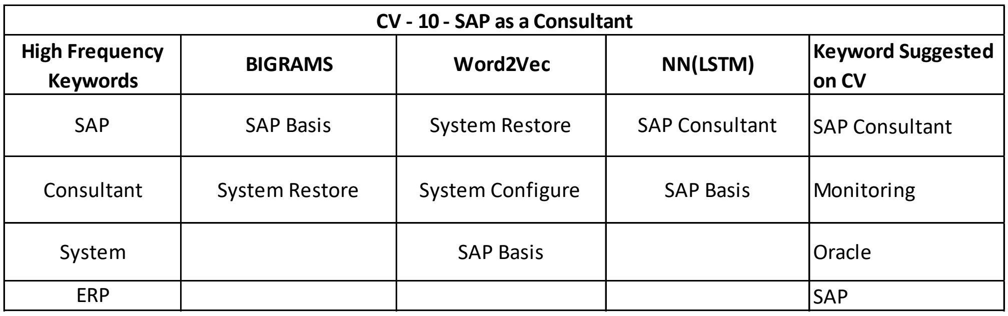
Table 6.9: Extracted Keywords for Software Engineer CV.



CV - 9 - Software Engineer (C++)				
High Frequency Keywords	**BIGRAMS**	**Word2Vec**	**NN(LSTM)**	**Keyword Suggested on CV**
Development	Computer Engineer	Software Engineer	C++ Developer	Computer Engineer
C++	Design and ımplement	Design And Develop	Software Engineer	Software Engineer
Project			Computer Engineer	C++
Computer				
Engineer				

The suggested keywords are given by working on the CV titled SAP Consultant in Table 6.10

Table 6.10: Extracted Keywords for SAP Consultant CV.



CV - 10 - SAP as a Consultant				
High Frequency Keywords	**BIGRAMS**	**Word2Vec**	**NN(LSTM)**	**Keyword Suggested on CV**
SAP	SAP Basis	System Restore	SAP Consultant	SAP Consultant
Consultant	System Restore	System Configure	SAP Basis	Monitoring
System		SAP Basis		Oracle
ERP				SAP

With the K-Fold Cross Validation method, training and test set rates were determined to be 80% and 20%. This method has been used to prevent overfitting problem. Using the listed features, the LSTM model 72.2% accurate results on the assets test set. In our study with data preprocessing on this set, the addition of prefix and suffix increased model performance by 76.7% correct results. As a feature of the model, parts of speech were introduced using the NLTK library, and this ratio increased to 83.6%.

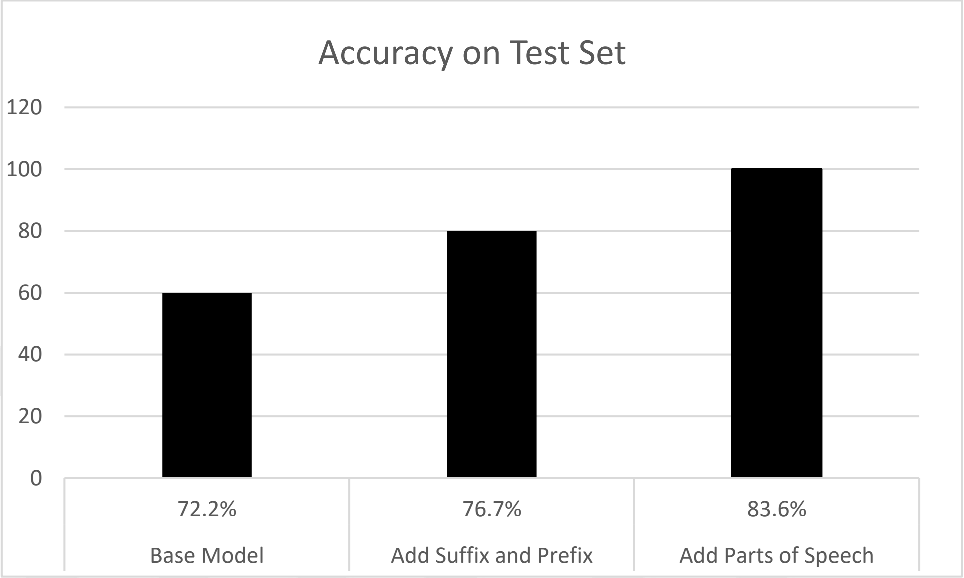


Figure 6.1: Accuracy Rate on Test Set

CHAPTER VII

CONCLUSION

This thesis focuses on deep learning, which has been widely used recently. Word counting, bigrams, Word2Vec and Neural Network (LSTM) were used to identify keyword groups in the study. The study was tested on 200 CV. In order to discover the skills in CVs and CVs, some skill definitions have been made about the jobs and the description has been added.

Different keywords have been tried to be suggested for each CV. The word groups for keyword extraction are selected in the CV. Then, with bigrams, the highest number of word groups are taken in binary groups and if the repetition rate is more than two of the binary groups, this word group is marked as a candidate for keyword. Using the Skip-Gram model for Word2Vec, suggested keywords were extracted. Of course, based on the work done during these periods, the accuracy values were increased within the datapreprocessing operations based on the base model for the created dataset. Better results were obtained by adding prefix-suffix and part of speech. While our accuracy value for the base model was 72.2%, this value reached 76.7% after the prefix and suffix were added. In the last part, the highest value has been reached in the part of speech. This value is found as 83.6%.

All CVs used in the model were selected from the informatics sector. Accordingly, job definitions and skills definitions have been made for this sector. Although a few CV skills extractions belonging to other sectors were attempted in the model for trial purposes, the success rate was lower than that of the IT sector. Model efficiency will decrease to be able to infer from CVs that have different structural and skill definitions. Talent definitions can also create differences for different people in the same industry. One of the biggest challenges in the model and the estimation that reduces the success rate; Since the company name includes concepts such as PYTHON - SQL - JS - Microsoft, the predict estimate of whether it is a company name or skills is reduced. There are many similarities in company names and defined in our skills library. This is what causes low performance. Thanks to the extracted keyword extraction, it has been ensured that companies or individuals save both time and manual work, and it is aimed to provide a lot of convenience with a success rate of 83.6% for very large dataset.

REFERENCES

[1] C. Zhang, H. Wang, Y. Liu, D. Wu, Y. Liao, and B. Wang, “Automatic keyword extraction from documents using conditional random fields,” *J. Comput. Inf. Syst.*, 2008.

[2] I. Mani, D. House, G. Klein, L. Hirschman, T. Firmin, and B. Sundheim, “The TIPSTER SUMMAC Text Summarization Evaluation,” 1999, doi: 10.3115/977035.977047.

[3] H. P. Luhn, “A Statistical Approach to Mechanized Encoding and Searching of Literary Information,” *IBM J. Res. Dev.*, 2010, doi: 10.1147/rd.14.0309.

[4] K. S. Jones, “A statistical interpretation of term specificity and its application in retrieval,”*Journal of Documentation*. 1972, doi: 10.1108/eb026526.

[5] “Effective Approaches For Extraction Of Keywords,” *Int. J. Comput. Sci. Issues*, 2010.

[6] N. Kumar and K. Srinathan, “Automatic keyphrase extraction from scientific documents using N-gram filtration technique,” in *DocEng’08 - Proceedings of the 8th ACM Symposium on Document Engineering*, 2008, doi: 10.1145/1410140.1410180.

[7] P. D. Turney, “Coherent Keyphrase Extraction via Web Mining,” in *IJCAI International Joint Conference on Artificial Intelligence*, 2003.

[8] P.D. Turney, “Learning to Extract Keyphrases from Text,” 1999.

[9] M. Chen, J. T. Sun, H. J. Zeng, and K. Y. Lam, “A practical system of keyphrase extraction for Web pages,” in *International Conference on Information and Knowledge Management, Proceedings*, 2005, doi: 10.1145/1099554.1099625.

[10] D. Bourigault, “Surface grammatical analysis for the extraction of terminological noun phrases,” 1992, doi: 10.3115/993079.993111.

[11] J. Ferreira, “A Local Maxima method and a Fair Dispersion Normalization for extracting multi-word units from corpora,” *Sixth Meet. Math. Lang.*, 1999.

[12] K. T. Frantzi and S. Ananiadou, “The C-value / NC-value domain independent method for multiword for multiword keyphrase extraction,” *J. Nat. Lang. Process.*, 1999.

[13] Z. Sui, Y. Chen, and Z. Wei, “Automatic recognition of Chinese scientific and technological terms using integrated linguistic knowledge,” in *NLP-KE 2003 - 2003 International Conference on Natural Language Processing and Knowledge Engineering, Proceedings*, 2003, doi: 10.1109/NLPKE.2003.1275948.

[14] O. Medelyan and I. H. Witten, “Thesaurus based automatic keyphrase indexing,” in Proceedings of the ACM/IEEE Joint Conference on Digital Libraries, 2006, doi: 10.1145/1141753.1141819. [20] O. Medelyan and I. H. Witten, “Thesaurus-based index term extraction for agricultural documents,” Proc. 6th Agric. Ontol. Serv. Work. EFITA/WCCA, 2005.

[15] O. Medelyan and I. H. Witten, “Thesaurus-based index term extraction for agricultural documents,” Proc. 6th Agric. Ontol. Serv. Work. EFITA/WCCA, 2005.

[16] J. Wang, H. Peng, and J. S. Hu, “Automatic keyphrases extraction from document using neural network,” in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2006, doi: 10.1007/11739685_66.

[17] Z. Zhou, X. Zou, X. Lv, and J. Hu, “Research on weighted complex network based keywords extraction,” in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2013, doi: 10.1007/978-3-642- 45185-0_47.

[18] Cavnar, W. B. & Trenkle, J. M. (1994, April). N-gram-based text categorization. In Proceedings of SDAIR-94, 3rd annual symposium on document analysis and information retrieval (Vol. 161175).

[19] S. M. Katz, “Technical terminology: Some linguistic properties and an algorithm for identification in text,” Nat. Lang. Eng., 1995, doi: 10.1017/S1351324900000048.

[20] B. Hong and D. Zhen, “An Extended Keyword Extraction Method,” Phys. Procedia, 2012, doi: 10.1016/j.phpro.2012.02.167.

[21] X. Hu and B. Wu, “Automatic keyword extraction using linguistic features,” in Proceedings - IEEE International Conference on Data Mining, ICDM, 2006, doi: 10.1109/icdmw.2006.36.

[22] Parasher, Priyanka. Different Types of Machine Learning Algorithms. https://ai.plainenglish.io/different-types-of-machine-learning-algorithms-28974016e108 27.05.2020.

[23] Z. Qingguo and Z. Chengzhi, "Automatic Chinese Keyword Extraction Based on KNN for Implicit Subject Extraction," 2008 International Symposium on Knowledge Acquisition and Modeling, Wuhan, China, 2008, pp. 689-692, doi: 10.1109/KAM.2008.87.

[24] Trevor H., Robert T., JH F., *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed., Springer-Verlag, New York, 2009.

[25] Uzun, Yasin. "Keyword extraction using naive bayes." Bilkent University, Department of Computer Science, Turkey www. cs. bilkent. edu. tr/~ guvenir/courses/CS550/Workshop/Yasin_Uzun. pdf. 2005.

[26] Zhang, Kuo, et al. "Keyword extraction using support vector machine." international conference on web-age information management. Springer, Berlin, Heidelberg, 2006.

[27]Matsuo, Yutaka, and Mitsuru Ishizuka. "Keyword extraction from a single document using word co-occurrence statistical information." *International Journal on Artificial Intelligence Tools* 13.01 (2004): 157-169. [28] A. Hulth, "Improved automatic keyword extraction given more linguistic knowledge," 2003, doi: 10.3115/1119355.1119383.

[29] Haggag, H., Amal Abutabl, and Ahmed Basil. "Keyword Extraction using Clustering and Semantic Analysis." International Journal of Science and Research 3 (2014): 1128-1132.

[30] J.A. Lozano, J.M. Pena, P. LarranagaAn empirical comparison of four initialization methods for the k-means algorithm Pattern Recognition Lett., 20 (1999), pp. 1027-1040

[31] Agrawal, Srikant R. Fast algorithms for mining association rules [C] / /Proc of International Conference on Very Large Databases. 1994: 487-499

[32] D. B. Bracewell, F. Ren, and S. Kuriowa, "Multilingual single document keyword extraction for information retrieval," in *Proceedings of 2005 IEEE International Conference on Natural Language Processing and Knowledge Engineering, IEEE NLP-KE'05*, 2005, doi: 10.1109/NLPKE.2005.1598792.

[33] R. M. Alguliev and R. M. Aliguliyev, "Effective summarization method of text documents," in *Proceedings - 2005 IEEE/WIC/ACM InternationalConference on Web Intelligence, WI 2005*, 2005, doi: 10.1109/WI.2005.57.

[34] Y. Ohsawa, N. E. Benson, and M. Yachida, "KeyGraph: Automatic indexing by cooccurrence graph based on building construction metaphor," in *Proceedings of the Forum on Research and Technology Advances in Digital Libraries, ADL*, 1998, doi: 10.1109/adl.1998.670375.

[35] A. Hulth, "Improved automatic keyword extraction given more linguistic knowledge," 2003, doi: 10.3115/1119355.1119383.

[36] T. Onoda, T. Yumoto, and K. Sumiya, "Extracting and clustering related keywords based on history of query frequency," in *Proceedings of the 2nd International Symposium on Universal Communication, ISUC 2008*, 2008, doi: 10.1109/ISUC.2008.22.

[37] S. Wang, M. Y. Wang, J. Zheng, and K. Zheng, "A hybrid keyword extraction method based on tf and semantic strategies for chinese document," in *Applied Mechanics and Materials*, 2014, doi: 10.4028/www.scientific.net/AMM.635-637.1476.

[38] P. D. Turney, “Learning algorithms for keyphrase extraction,” *Inf. Retr. Boston.*, 2000, doi: 10.1023/A:1009976227802.

[39] I. H. Witten, G. W. Paynter, E. Frank, C. Gutwin, and C. G. Nevill-Manning, “KEA: Practical Automatic Keyphrase Extraction.”

[40] S. R. El-Beltagy and A. Rafea, “KP-Miner: A keyphrase extraction system for English and Arabic documents,” *Inf. Syst.*, vol. 34, no. 1, pp. 132–144, 2009, doi:10.1016/j.is.2008.05.002.

[41] K. Sarkar, “Automatic Keyphrase Extraction from Medical Documents,” *Springer-Verlag Berlin Heidelb.*, pp. 273–278, 2009.

[42] K. Sarkar, “A Hybrid Approach to Extract Keyphrases from Medical Documents,” *Int. J. Comput. Appl.*, 2013, doi: 10.5120/10565-5528.

[43] Ø. D. Trier, A. K. Jain, and T. Taxt, “Feature extraction methods for character recognition - A survey,” Pattern Recognit., 1996, doi: 10.1016/0031-3203(95)00118-2.

[44] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, “Natural language processing (almost) from scratch,” *J. Mach. Learn. Res.*, 2011.

[45] S. Jean, K. Cho, R. Memisevic, and Y. Bengio, “On using very large target vocabulary for neural machine translation,” in *ACL-IJCNLP 2015 - 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, Proceedings of the Conference*, 2015.

[46] G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *Science (80-.).*, 2006, doi: 10.1126/science.1127647.

[47] Y. Bengio, “Learning deep architectures for AI,” *Found. Trends Mach. Learn.*, 2009, doi: 10.1561/2200000006.

[48] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput. 1997 Nov;9(8) 1735-1780. doi:10.1162/neco.1997.9.8.1735. PMID: 9377276.

[49] Wang, Y., and J. Zhang. "Keyword extraction from online product reviews based on bi-directional LSTM recurrent neural network." 2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM). IEEE, 2017.

[50] https://www.kaggle.com/elroyggj/indeed-dataset-data-scientistanalystengineer?select=indeed_job_dataset.csv