

KEYWORD EXTRACTION USING CO-OCCURRENCE GRAPH BASED
APPROACH

A THESIS SUBMITTED TO
THE FACULTY OF ARCHITECTURE AND ENGINEERING
OF
EPOKA UNIVERSITY

BY

ORALD VEIZI

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

MARCH, 2022

Approval sheet of the Thesis

This is to certify that we have read this thesis entitled “**Keyword Extraction Using Co-Occurrence Graph Based Approach**” and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Dr. Arban Uka
Head of Department
Date: March, 07, 2022

Examining Committee Members:

Dr. Igli Hakrama (Civil Engineering) _____

Dr. Shkelqim Hajrulla (Civil Engineering) _____

Dr. Julian Hoxha (Civil Engineering) _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name Surname: Orald Veizi

Signature: _____

ABSTRACT

KEYWORD EXTRACION USING CO-OCCURRENCE GRAPH BASED APPROACH

Orald Veizi

M.Sc., Department of Computer Engineering

Supervisor: Dr. Julian Hoxha

The complexity to get relevant information for a user is very high due to increasing rate of text over the internet. To address these issues, more study has been conducted when information is gathered and text analytics, and it is the most popular research area in terms of extracting keyworders. There are many types of data regarding to the observations and analysis such as graphical data and others. The user may also produce data by using social media, Wikipedia, or any other resource. Most of the people generate their own data by Twitter (social media, considered as one of the most popular platforms for crawling the short text, because it contains 140 characters per tweet).

Keyword extraction is a process where a text is givento the computer and the computer return a set of keywordsthat recommended topical words and phrases from the contentof documents. Keyword extractionhelps the reader to understand the summary or at least the coreidea of the document without reading the whole document. Asa result, the prospect readers do not waste their valuable timesreading the irrelevant documents comprehensively. Generaly, by searching the keywords, users could find related posts toan event. Keyword extraction methods are being appliedto many areas especially when we extract keywords in the areaof information retrieval. This has a particular interest becausepeople retrieve significant information based on keywords. In this thesis, we have used agraph-based keyword extraction algorithm over four different datasets collected from Twitter on different terms. By the preprocessing of datasets through NLTK we will get more optimized data, and the co-occurrence graph

also generated by this dataset. Moreover, we have also shown whether the study of co-occurrences allows keeping track of the structure of each text, however, it is more tedious to handle and often leads to messy visualizations.

There are many libraries there for visualization, python is giving more reliability for plotting because it provides many built-in libraries. TextRank algorithm is a graph-based keyword extraction algorithm, it follows the Google PageRank algorithm but somehow it is different from that by the words and links. TextRank calculates the score of every relevant word and by that score, we can find more important words of the corpus, further, it also finds the precision of those relevant words. Word cloud is also enhancing its popularity by the visualization, by its different look there are many word clouds are present over the internet.

The genuine data set, crawled from Twitter, provides the data for the experimental assessment of the proposed work.

Keywords: *co-occurrence graph, information retrieval, TextRank, PageRank*

ABSTRAKT

EKSTRAKTIMI I FJALËVE KYÇE DUKE PËRDORU QASJEN E BAZUAR NË GRAFET E BASHKË-NDODHURA

Orald Veizi

Master Shkencor, Departamenti i Inxhinierisë Kompjuterike

Udhëheqësi: Dr. Julian Hoxha

Kompleksiteti për të nxjerrë informacion te vlefshëm për një përdorues është shumë i lartë për shkak të rritjes së shkëmbimit të informacioneve në internet. Për të adresuar këtë problematikë, shumë studime janë kryer në lidhje me ekstraktimin dhe analizimin e fjalëve kyçe. Ka shumë lloje të dhënash në lidhje me vëzhgimet dhe analizat, si të dhënat grafike, tableare dhe të tjera. Përdoruesi mund të prodhojë gjithashtu të dhëna duke përdorur mediat sociale, Uikipedia (Wikipedia) ose ndonjë burim tjetër. Shumica e njerëzve gjenerojnë të dhënat e tyre nga Tuitër (Twitter) (media sociale, e konsideruar si një nga platformat më të njohura për zvarritjen e tekstit të shkurtër, sepse përmban 140 karaktere për postim).

Nxjerrja e fjalëve kyçe është një proces ku një tekst i jepet kompjuterit dhe kompjuteri kthen një grup fjalësh kyç që rekomandojnë fjalë dhe fraza aktuale nga përmbajtja e dokumenteve. Nxjerrja e fjalëve kyçe ndihmon lexuesin të kuptojë përmbledhjen ose të paktën idenë thelbësore të dokumentit pa lexuar të gjithë dokumentin. Si rezultat, lexuesit e mundshëm nuk e humbin kohën e tyre të vlefshme duke lexuar në mënyrë gjithëpërfshirëse dokumentet e parëndësishme. Në përgjithësi, duke kërkuar fjalët kyçe, përdoruesit mund të gjenin postime të lidhura me një ngjarje. Metodat e nxjerrjes së fjalëve kyçe po aplikohen në shumë fusha, veçanërisht kur nxjerrim fjalë kyçe në fushën e marrjes së informacionit. Kjo ka një interes të veçantë sepse njerëzit marrin informacion të rëndësishëm bazuar në fjalë kyçe. Në këtë tezë, ne kemi përdorur algoritmin e nxjerrjes së fjalëve kyçe të bazuara në graf mbi katër grupe të dhënash të ndryshme të mbledhura nga Tuitër (Twitter) në terma të ndryshëm.

Me përpunimin paraprak të grupeve të të dhënave përmes NLTK, ne do të marrim të dhëna më të optimizuara, dhe grafi i bashkë-ndodhjes do të gjenerohet gjithashtu nga ky grup të dhënash. Për më tepër, ne kemi treguar gjithashtu nëse studimi i bashkë-ngjarjeve lejon mbajtjen e gjurmëve të strukturës së çdo teksti, megjithatë, është më i lodhshëm për t'u trajtuar dhe shpesh çon në vizualizime të çrregullta.

Ka shumë librari për vizualizim, python po jep më shumë besueshmëri për komplotimin sepse ofron shumë librari të integruara. Algoritmi “TextRank” është një algoritëm i nxjerrjes së fjalëve kyçe i bazuar në grafik, ai ndjek algoritmin e “Google PageRank”, por disi ndryshon nga ai me fjalët dhe ndërlidhura. “TextRank” llogarit rezultatin e çdo fjale përkatëse dhe me atë pikë, ne mund të gjejmë fjalë më të rëndësishme të korpusit, më tej, gjen edhe saktësinë e atyre fjalëve përkatëse. Reja e fjalëve po rrit gjithashtu popullaritetin e saj nga vizualizimi, me pamjen e saj të ndryshme ka shumë re fjalësh të pranishme në internet.

Seti i vërtetë i të dhënave, i zvarritur nga Tuitet (Twitter), ofron të dhëna për vlerësimin eksperimental të punës së propozuar.

Fjalët kyçe: grafe të bashkëndodhura, marrje informacioni, TextRank, PageRank

ACKNOWLEDGEMENTS

I would like to express my special thanks to my supervisor Dr. Julian Hoxha for his enormous guidance, motivation, encouragement and support during the writing and all the stages while I was working of my thesis. The door to Dr. Hoxha office was always open whenever I ran into a trouble spot or had a question about my research or writing. I sincerely appreciate the time and effort he has spent to improve my experience during my graduate years.

I would also like to thank Prof. Dr. Arban Uka for his teachings during my Master Studies and his great contribution of explaining the topics of Image Processing in a very interesting and engaging way.

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

Thank you to You all!

TABLE OF CONTENTS

ABSTRACT	iii
ABSTRAKT	v
ACKNOWLEDGEMENTS.....	vii
TABLE OF CONTENTS.....	viii
LIST OF FIGURES	xi
LIST OF TABLES	xii
LIST OF ABBREVIATIONS.....	xiii
CHAPTER 1	1
INTRODUCTION.....	1
1.1 Overview	1
1.2 Problem Definition	2
1.3 Scope of Proposed Work.....	3
1.4 Thesis Outline.....	3
CHAPTER 2	5
PRELIMINARIES	5
2.1 Introduction	5
CHAPTER 3.....	9
LITERATURE REVIEW.....	9

3.1	Introduction	9
CHAPTER 4		20
METHODOLOGY.....		20
4.1	Introduction	20
4.2	Why Graph-based approach?	20
4.3	Graph Co-occurrence.....	21
4.4	TextRank Algorithm.....	22
4.5	Generating Graph-based Data	23
4.5.1	Crawling Data from Twitter	23
4.5.2	Tweets Pre-processing	23
4.5.3	Construction of Co-occurrence Graph.....	23
4.5.4	Normalization of Matrix	24
4.5.5	Keyword Extraction.....	24
4.5.6	Calculating Precision	24
4.5.7	World-Cloud Constructing	24
CHAPTER 5		25
EXPERIMENTS AND RESULTS.....		25
5.1	Introduction	25
5.2	Creating Dataset	25
5.2.1	Data Collection	25
5.2.2	Natural Language Pre-processing.....	29
5.2.3	Case Folding	32
5.2.4	TF-IDF	32

5.3	Implementation of TextRank Algorithm.....	35
5.4	Precision	37
5.5	Word Cloud	38
CHAPTER 6	40
CONCLUSIONS AND FUTRE WORK	40
6.1	Conclusions	40
6.2	Future Work.....	41
REFERENCES	42

LIST OF FIGURES

Figure 1. Example of a Graph	5
Figure 2. Classification of keyword extraction method	10
Figure 3. Explaining the Precision and Recall	13
Figure 4. Text as a Graph	14
Figure 5. Classification of Keyword extraction methods	16
Figure 6. Classification of Keyword Graph Types	17
Figure 7. Classification of graph-based methods	18
Figure 8. Co-Occurrence Graph	21
Figure 9. Sample Tweets	27
Figure 10. Access Tokens and Secret keys	27
Figure 11. Crawling of Data by Python	29
Figure 12. Stemming	31
Figure 13. Pre-Processed Data	33
Figure 14. Co-occurrence of relation of words	33
Figure 15. Code for visualize of graph	34
Figure 16. Graph visualization of dataset	35
Figure 17. Score	36
Figure 17. Score	38
Figure 19. Word Count of Dataset	39

LIST OF TABLES

Table 1. Overview of Test Statistics	7
Table 2. Tweets related to various events	26
Table 3. Precision of various events	37

LIST OF ABBREVIATIONS

NLTK	Natural Language ToolKit
URL	Uniform Resource Locator
TF	Term Frequency
IDF	Inverse Document Frequency
API	Application Programming Interface
TP	True-Positive
FP	False-Positive
TN	True-Negative
FN	False-Negative
IG	Information Gain
MI	Mututal Information
CN	Candidate Network
IR	Information Retrieval

CHAPTER 1

INTRODUCTION

1.1 Overview

Here, I explain a short introduction about the keyword word extraction model from microblogging dataset based on graph-based. Keyword extraction is a task (also the set of techniques) for extracting required keywords from the text. The keyword extraction approach is also used to recover meaningful information from large amounts of data based on a particular query. There are various real applications of keyword extraction as follows:

- Web search.
- Text summarization.
- Trending on Twitter.
- Finding the main topic from the text.

Because of the fast rising Internet usage, there are webpages and programs with an abundance of user-generated information being produced, and more are being added on a regular basis. Twitter, Quora, and StackOverflow are examples of brief text websites and applications. User-generated material is becoming increasingly diverse and brief. It has become more important for the application service provider to handle a significant volume of brief client material. Its management is focused on data acquisition and consistency. There are different techniques of information retrieval, keyword extraction is one of the most identified technique around the globe in terms of the research area, nowadays. Keyword extraction is important in many domains, including text retrieval, text clustering, text summarization, and several data processing applications. By the keyword extraction, we can go through the document whether it is relevant or irrelevant, many documents might contain more than enough pages and processing through them will also take much time. In recent time many researchers

analyzed keywords extraction from the documents containing short text, we can also call it Micro-blogs.

Micro blogsites have increasingly gained those who want to promote themselves and communicate with others. There exists a big number of micro blogsites but Twitter is the most used. I consider my work of analysis for Twitter social network. Twitter allows users to tweet (maximum length message of 140 characters), and tweets can include text, photos, and videos. The post is accessible to all Twitter users without restriction. Twitter users can follow people without having to verify their following. This has resulted in Twitter being a popular social media platform for spreading news. It has around 336 million active users (as of 2019). Smart-phones and other online apps are simple to use.; anybody with a basic understanding of how to operate a smartphone may utilize social networking services. This was like an explosion of data in social-networks. The challenging part now is determining how to efficiently use this data and extract meaningful information from it.

1.2 Problem Definition

The data on the internet is unstructured data, the way to organize the data in a structured form we use graph. Depends on the type of the information to be analyzed, the graph can be directed/undirected. As with many complex information, such as a virtual communities, an usually denoted by a collection of characteristics, and various interactions exist among an instance pair.

By the touch in recent research of text analytics area, there have been lots of work done and this amount is steadily growing in the future. The problem of my thesis is about concentrating on the most critical keywords from the text, and that text is graph-based text. Provided a collection of Twitter articles all of which are linked by a similar search term (i.e., a topic), the data is crawled from Twitter (social network) on the basis of some countable tweets from four different events. The tweets are preprocessed by NLTK (Natural Language Tool Kit), in which stemming, punctuation, stop words removal, and URL removal processed. With only consideration of the text, hashtags, and timing of tweets, all the other unnecessary things removed by NLTK. To generate a set of keyword that is relevant to the document and finds out the most

relevant keywords by ranking them. Before calculating the score of keywords through the TextRank algorithm, I create an incident matrix for normalization of words. Finally, the top-k keywords are being selected to evaluate the context of the document.

1.3 Scope of Proposed Work

In research, a lot of documents, journals, and papers need to be processed to find out key information. In sentimental analysis fields, keywords define the core information of the documents. Data analysis requires a huge amount of processing in order to identify relative information. At present, it is almost impossible to keep track of the similar type of documents, journals or research papers altogether. Moreover, detecting relevant articles would be slow and time-consuming. By extracting keywords from the documents would be a sufficient approach to keep track of the similar type of journals, articles or documents. Many algorithms or approach has been introduced to extract a keyword from documents. Term frequency, POS-tagging, semantic relation, Support vector machine, C4.5 decision trees, Conditional random fields, etc. are already introduced in this field and all approaches have shown better results in extracting keywords but to increase the precision and recall, the relation between words and sentences needs to be captured carefully. Even they didn't show sufficient result for microblogging data. In this thesis, we utilized a graph-based approach to extract the keywords that will help us to understand the context of the microblogging dataset.

1.4 Thesis Outline

The other part of the chapters are structured as follow:

Chapter2 the preliminaries here it defines the concepts that are used in the thesis and in my work, which will give a brief to the reader to understand and read the rest of chapters comfortably.

Chapter 3 is the literature review, this chapter gives an introduction to the related works done on the same topic.

Chapter 4 is proposed work, this chapter explains the methods that are used in experiments for Keyword extraction using a co-occurrent graph-based approach.

Chapter 5 is the experimental result and evaluation, this chapter explains step by step the experiments over four datasets.

Chapter 6 is about the results and future research of the suggested work.

CHAPTER 2

PRELIMINARIES

2.1 Introduction

Here, I am introducing some terms that I am using in the rest of the thesis. It helps for understanding those terms which are used in further chapters.

Definition 2.1. The *Keyword Extraction* is the task (set of techniques) for extracting interesting keywords from the text.

Definition 2.2. The *Supervised method* is the method in which data is trained with the labels. The input data is labeled for training. Classification comes under supervised learning.

Definition 2.3. The *Unsupervised method* is the method in which data is not trained with the labels. The process of attempting to discover hidden information in dataset.

Definition 2.4. A *Graph* 'G' is a set of vertex 'V' called Nodes that are connected by edges 'E' called Links. Mathematically we can say $G = (V, E)$. The graph is a way to formally represent a network or collection of interconnected objects. Graph is a powerful tool for modeling database objects and their relationships among data items in various application domains.

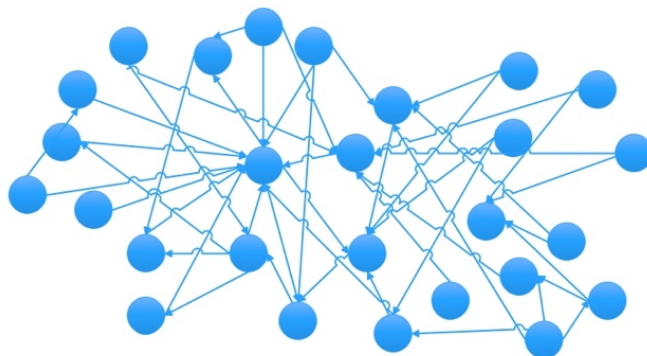


Figure 1. Example of a Graph

Definition 2.5 The *Co-occurrence graph* is the relationship between neighbor nodes. It is also called a ‘neighborhood’ relation. It always takes the next word as a node and creates a relationship between them.

Definition 2.6. The *Similarity matrix* allows you to understand how similar or far apart each pair of items is from the participants’ perspective. It is also called a distance matrix. It is a matrix of scores that represents the similarity between the pairs of data points by 0 and 1.

Definition 2.7. The TF (*Term Frequency*) Count the number of times a phrase appears in a document. We calculated as.

$$TF = \frac{\text{Number of term } t \text{ appears in a document}}{\text{Total number of terms in the document}}$$

Definition 2.8. The *IDF (Inverse Document Frequency)* determines the significance of a phrase We may compute it as follows:

$$idf = \log \frac{\text{Total number of documents}}{\text{Number of documents with term } t \text{ in it}}$$

Definition 2.9. A true positive (Tp)test answer is one which finds the issue when it exists.

Definition 2.10. A *true negative (Tn)* test answer is one which does not find the issue when it doesn’t exist.

Definition 2.11. A *false positive (Fp)* test answer is one which finds the issue when it is not present.

Definition 2.12. A *false negative (Fn)* test answer is one which does not find the issue when it is present.

Table 1. Overview of Test Statistics

		-Condition-	
		-Present-	-Absent-
-TEST-	-Positive-	-True- Positive-	-False- Positive-
	-Negative-	-False- Negative-	-True- Negative-

Definition 2.13. A *Precision(P)* is the proportion of necessary documents among those recovered.

$$\text{Precision} = \frac{Tp}{Tp+Fp}$$

Definition 2.14. A *Recall(R)* is the proportion of necessary documents that are found.

$$\text{Recall} = \frac{Tp}{Tp+Fn}$$

Definition 2.15. The *F-measure* a single metric that compromises precision for recall the weighted harmonic mean of accuracy and recall is used.

$$\text{F-measure} = 2 \times \frac{\text{Precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

Definition 2.16. A *PageRank algorithm* a Google search technique used to rank web sites in search rankings. It calculates the important pages over the internet and retrieves based on popularity.

Definition 2.17. The *Text summarization* is the process of shortening a text document with software, in order to create a summary with major points of the original document.

Definition 2.18. The *TextRank algorithm* a graph-based keyword extraction algorithm which uses a Google PageRank. The assumption of this algorithm is when a word appears frequently.

Definition 2.19. An *Application Programming Interface (API)* is the part of the server that receives the request and sends responses. It is a software intermediary that allows two applications to talk to each other.

Definition 2.20. The *NLTK* is a natural language processing toolkit that allows to pre-process of the data such as stop words removal, hashtag removal, URLs removal and stemming, etc. It is provided by natural language processing.

Definition 2.21. The process of collecting useful data from a set of data is referred to as information extraction.

CHAPTER 3

LITERATURE REVIEW

3.1 Introduction

This part is related to the extraction of knowledge through different proposed work.

A more and more web-based community add the data explosion in the internet. The growing number of unstructured and massive data on the internet provides individuals with many benefits, but it makes information processing and extraction harder. The keyword is the most concise description of the text that can also give us the impact of relevant or irrelevant document, actually by the keyword we can summarize the document too. Keyword extraction is a trending topic in text analytics field and there are many ways of the keyword extraction. The search engine operates on the premise of keyword extraction, which means that when we input a word into the search bar, it displays recommendations further down in the search bar, and when we search, Google returns numerous links linked to that term. This process is totally based on the keyword extraction and analyzing the process. Here I considered my micro-blog dataset with corresponding to the graph based.

Yujun Wen et al. [1] talks about the classifications of the keyword extraction method:

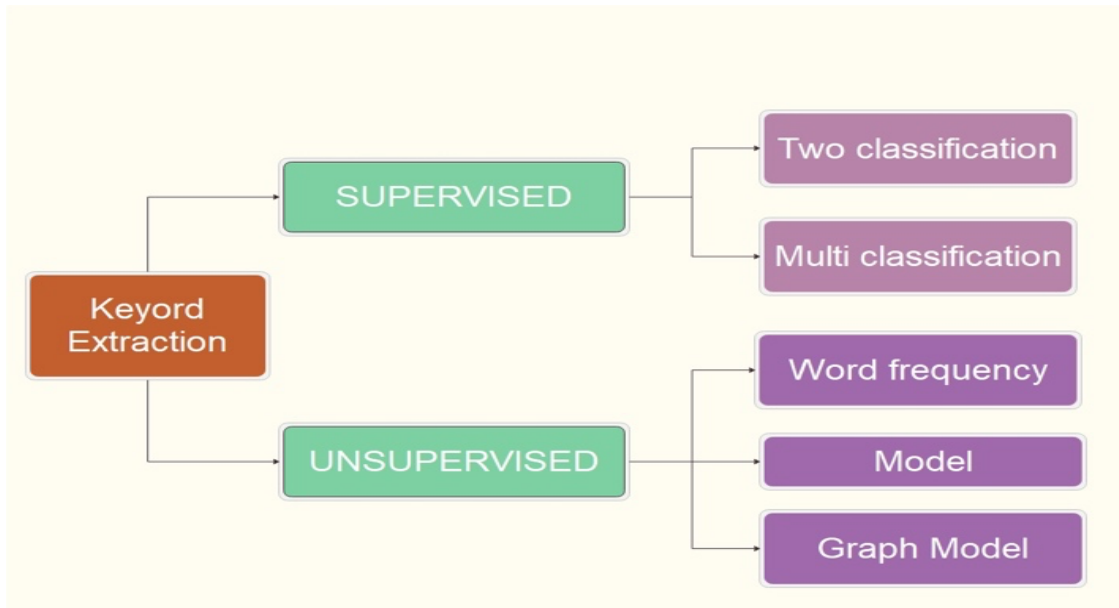


Figure 2. Classification of keyword extraction method

Keyword extraction can be divided into two categories:

- Supervised Keyword Extraction.
- Unsupervised Keyword Extraction.

Supervised methods are without human intercession that directly extracts keywords from information and way of the text with improving efficiency.

Unsupervised methods can be summarized in three kinds, keywords extraction based on the statistical characters of TF-IDF, keyword extraction model based on the theme of keywords and keywords extraction based on the word graph model.

In this paper authors performed research on the keyword extraction from news articles, they build a candidate keyword graph model based upon TextRank, by calculation of similarity between words as transition probability of nodes, then by an iterative method calculate the score of words and finally pick the top N-keywords as that final result. They also create a process of constructing the candidate graph model. Many other scholars drag this field in other areas of the graph to extend the work of keyword extraction from graph-based models. Further **Jian Cao et al.** [2] describes the way to improve graph based keyword extraction, authors suggested a method for calculating the relevance of co-occurrence words in a document and applying it in a

graph approach to discover more representative phrases ,also incorporate the degree of word co-relation in the document language network to boost performance when extracting the average number of keywords in texts. They created a graph on the basis of, word-to-word graph building, sentence-to-sentence graph building, and word-to-sentence graph building. The provided word set $W=\{w_j|1 \leq j \leq n\}$ of a document, the relation between any two words w_i and w_j can be computed using approaches such as mutual information. Mutual information between words can be followed as

$$\text{Sim}(w_i, w_j) = \log N \times p(w_i, w_j) / p(w_i) \times p(w_j)$$

Similarity is taken because of the saving space and time to compute the co-relation between words.

The word-to-word graph can be build on the basis of:

$$W_{ij} = \begin{cases} \text{Sim}(W_i, W_j) \times \text{Weight}(W_i, W_j) & \text{if } i \neq j \\ 0 & \text{otherwise} \end{cases}$$

For the sentence-to-sentence, given the sentence set $S=\{S_j | 1 \leq j \leq m\}$ of a document, each sentence is represented as a node, the sentence collection is represented as an undirected graph by generating an edge between each sentence and the weight of an edge is their relation determined by their content. Here graph's edge weight is described by the matrix

$$U_{ij} = \begin{cases} S_i \times S_j \frac{1}{||S_i|| \times ||S_j||} & \text{if } i \neq j \\ 0 & \text{otherwise} \end{cases}$$

For the sentence to word graph building, they assume that given the set of words $W=\{w_j|1 \leq j \leq n\}$ and set of sentences $S=\{S_j | 1 \leq j \leq m\}$ of a document. We can build a weighted graph from S and W and defined their relation in the following way

$$\text{Weight}(W_i, S_j) = S_{j_i} / S_j$$

For the co-occurrence, they choose TF-IDF of words in a document to get the score of words. Further **Md Rafiqul Islam et al.** [3] proposed a new improved method for keyword extraction using random walk model by considering position of terms within the document and information gain (IG) of terms corresponds to the whole set

of document, they also incorporate mutual information (MI) of terms with help of a random walk model to extract keywords from documents. TextRank established a random walk model before this algorithm. Several forms of random walks have been created before this algorithm, if we consider the TextRank than we know how previously it founded successfully in a number of applications, including web link analysis, social networks, citation analysis, and more recently in several text processing applications. As we know the term weighting is the one of the most consideration of keyword extraction, the motive of the term weighting method acting is to assort terms by allocating them weights corresponding to how well they amend both precision and recall. They calculate the results on the basis of term frequency (TF) and inverse document frequency (IDF) and random walk weights. For the keyword extraction method using a graph-based random walk, they followed the process as first they identified the text than calculate the TF-IDF of the pre-processed document or text and then vertices are stored on their final score. Further, there are many ways to analyze the text for keyword extraction, some people use the ranking of the words whereas some people are using precision and recall, and others are using indexing, it means there are so many aspects to find keyword extraction. Their work of post processing phase was similar to the Mihalcea [4]. **Jing Zhou et al.** [5] describes the work for ranking the result of keywords over the structured data by the proposed approach. Their work is based upon the schema graph-based approach to keyword search which comprises of a candidate network (CN) generation and its evaluation phase. By this idea, the ranking process can also be done to the keywords which result in optimized words from the document. Precision and recall are the most common parameters to find the accuracy of the words in a corpus. They are calculated in terms of true positive (TP), true negative (TN), false positive (FP) and false negative (FN).

True positive is an outcome where the text anticipates the positive class, Similarly True negative is an outcome where the text anticipates the negative class. False positive is an outcome where the text incorrectly anticipates positive class, Similarly False negative is an outcome where the text incorrectly anticipates negative class.

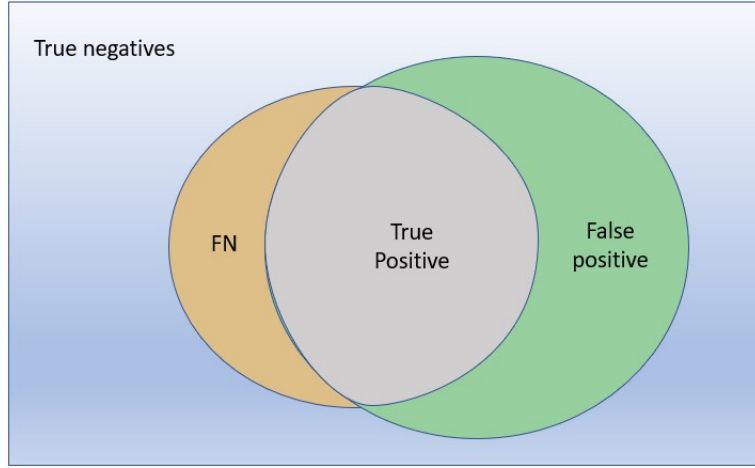


Figure 3. Explaining the Precision and Recall

Moreover, lots of research have been done in the part of keyword extraction and many authors processed data in the form of structure and unstructured, such as graph form, directed graph, undirected graph, weighted graph, attributed graph, etc. **RadaMihalcea et al.** [6] describe the type of data extracted by them and they converted the text in the form of the graph as structured data. They introduce the TextRank graph-based ranking models for graphs extracted from natural language text, they evaluate and investigate the applications of TextRank to two language processing tasks containing unsupervised keywords and sentence extraction and shows the results obtained with TextRank are competitive with state-of-the-art systems developed in these areas. There is also work done in the area of a directed graph, for the directed graph we need recursive algorithms to compute the graph because nodes contain self-loops in the possible case and those algorithms can be also applied to the undirected graph. In that scenario the out-degree is equal to the in the degree of a vertex and if the graph is loosely connected then several edges proportional with the number of vertices. If we want to convert the graph into weighted graph, then it can be computed as there will be an edge if the two vertices V_i and V_j have a connection between them and W_{ij} added to their edge and it will be determined as

$$W S(V_i) = (1-d) + d * \sum_{V_j \in In(V_i)} \frac{W_{ji}}{\sum_{V_k \in Out(V_j)} W_{jk}} WS(V_j)$$

Text as a graph can be build which acts like text and interconnects words or other text entities with meaningful relations. Text units of various size and characters can be added as vertices in the graph. There are many types of the graph can be

generated such as co-occurrence graph in which a node will generate the connection to the next of the node. After the creation of graph author find the keywords from that graph and applied a TextRank algorithm, which is an extractive summarization method build upon PageRank algorithm. TextRank algorithm will extract the most important word in last and achieves the highest precision and F-measure across all the systems, although recall is not high as in supervised methods. The assumption of this proposed work is ranking model of graphs and shows how it can be successfully used for natural language applications and shows the accuracy achieved by TextRank in these types of applications. An important aspect of TextRank is that it does not require deep linguistic knowledge, nor domain or language-specific annotated corpora which makes it highly portable to other domains, genres or languages.

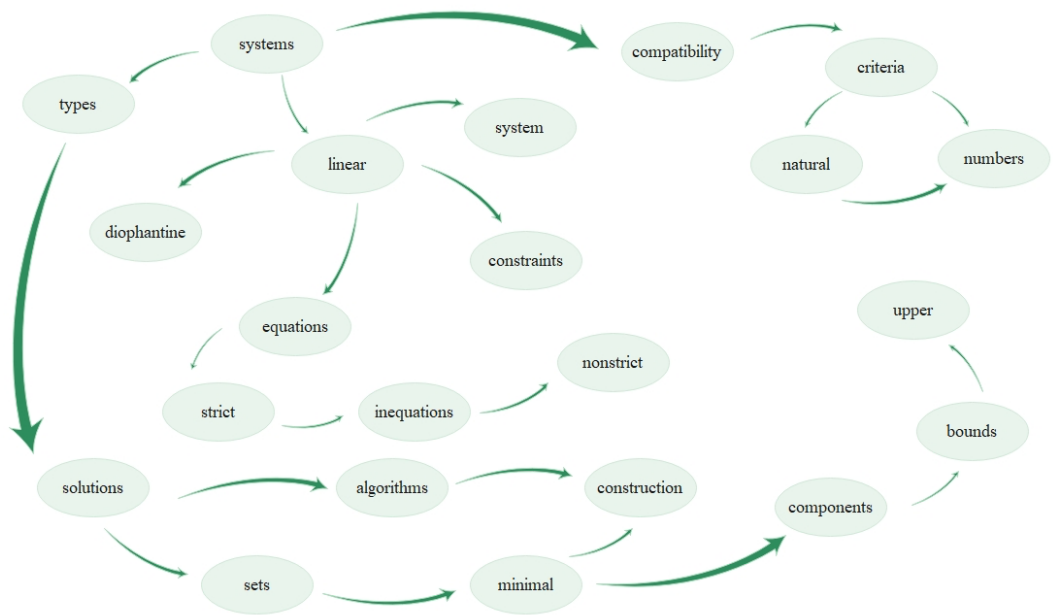


Figure 4. Text as a Graph

Larry Page et al. [7] The Google Page Rank is described to find out the web page's popularity score. The Page Rank formula presented in front of the world in Brisbane at seventh World Wide Web conference (WWW98) by Sergey Brin and Larry Page (founders of Google in 1998). There is an algorithm for calculating the score of the webpage and it is an iterative process. Page Rank also used in the graph field to calculate the popular nodes (having a greater number of in-degrees). Webster starts a

random webpage, whenever he visits a web page, the randomly hyperlink on that page chosen by him. The Web pages with hyperlinks between them are viewed as a directed graph called hyperlink graph. If there are some web pages P_i and P_j and there is a hyperlink that points from P_i to P_j called Outlink and hyperlinks that points from P_j to P_i called Inlink. For the representation, a matrix will be created, called Hyperlink matrix (aka Adjacency Matrix) of the graph. For the popularity score, a link from a more valuable page to the user's page is more important and link from a page having more outlines to user's page is less valuable. To calculate the popularity score there are some steps followed by the algorithm:

1. The Hyperlink Matrix.
2. The Stochastic Matrix.
3. The Google Matrix.
4. First Iteration.
5. More Iteration.

Sergey Brin et al. [8] they present Google a paradigm for a search engine (large scale) that makes the heavy use of structure in hypertext, the assumption of their proposed work is designed to crawl and indexing to the web optimized and generate much more satisfying search results. They worked in the search engines for the most important and frequent results on search engine and scaling the web pages with the web. They started work from the improved search quality (quality of web search engine) and academic search engine research (firstly the academic domain was commercial but up till now it has gone to companies with little publications of technical details). Further for the system features they bring the order of text in the form of PageRank; through this rank algorithm they assign a rank on every web page but there was an issue of dangling links. Dangling links are those links that point to any web page with no outgoing links. They are affecting the model because it is not clearly visible where their weight should be disturbed and there are huge numbers of those pages on the internet. Dangling pages don't affect the ranking of another page directly, there is a solution of those pages that we simply remove them from the system until all the webpages are calculated. After calculation, they can be added back without affecting things significantly. **Wengen Li et al.** [9] Authors proposed work for the combined algorithms PageRank and TextRank for better precision and recall. Firstly, they create a matrix (keyword matrix) and then use traditional TextRank for keyword

extraction. Traditional TextRank is graph-based algorithm in which words are equal to the nodes in the graph and the connection between words is equal to the edge of the graph and the connection between words is represented by the occurrence number of the word in the fix-sized sliding window. This algorithm works like the PageRank algorithm, it considers that if a word connects to another word through an edge, then the word cast a vote for latter whereas PageRank considers nodes with Inlinks and Outlinks. **Slobodan Beliga et al.** [10] describes an overview of graph-based keyword extraction methods and approaches, in which many graphs can be considered for the keyword extraction analyze and compare. They suggest future work and boost the development of new graph-based approach for keyword extraction.

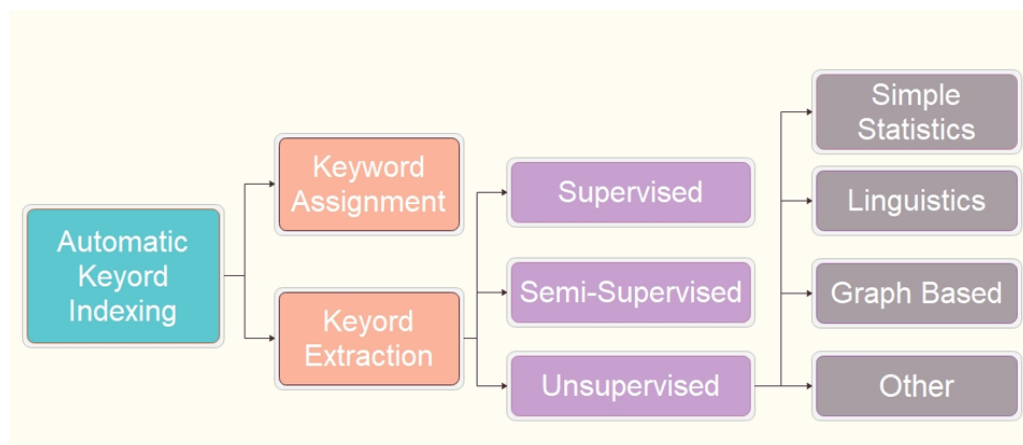


Figure 5. *Classification of Keyword extraction methods*

According to [9] keyword extraction roughly divided into different approaches such as

- Statistical Approaches.
- Machine Learning Approaches.
- Linguistic Approaches.
- Other Approaches.

Statistical Approaches: Incorporate simple methods which do not demand the training data. The statistic of a document can be used to identify keywords n-gram statistics, TF-IDF model, word frequency, co-occurrences words, etc. The

disadvantage of this approach is that the most important keywords will appear only once, such as from the medical domain.

Linguistic Approaches: It uses the linguistic properties in the document for each sentence and keyword. The most examined properties used in this approach are Lexical, semantic and syntactic.

Machine Learning Approaches: It depends on supervised and unsupervised learning, but the keyword extraction process is more reliable on supervised learning. In supervised learning, there is given a set of trained keywords but unfortunately, authors usually assign keywords to their documents only when they are compelled to do so. The model can be induced using one of the machine learning algorithms: SVM (Support Vector Machine), Naïve Bayes, C4.5, etc.

Other Approaches: Combine all the methods above in general, sometimes they incorporate heuristic knowledge such as position, length and text formatting, etc. for the fusion.

Further authors discuss the keyword extraction on graph-based data, moreover, they classified the graph types.

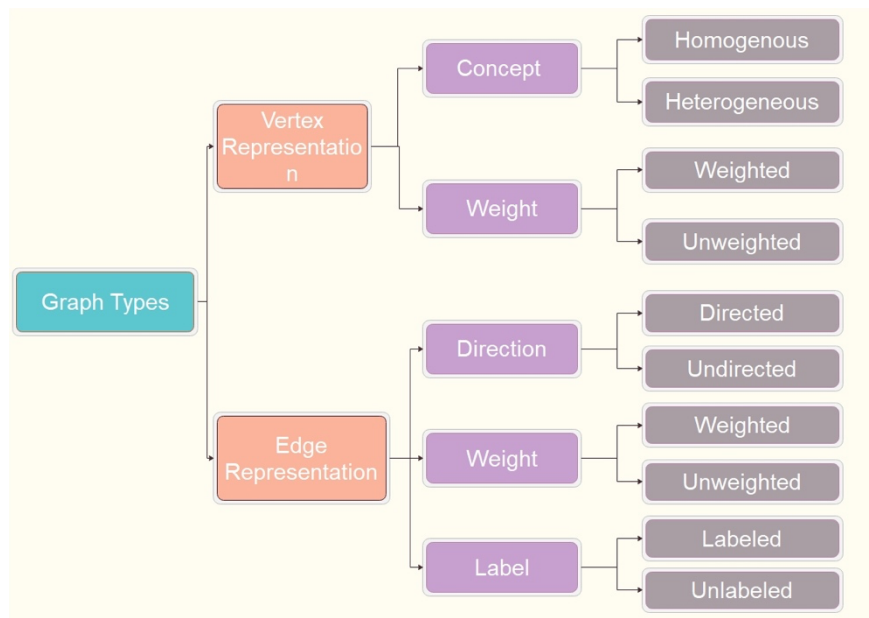


Figure 6. Classification of Keyword Graph Types

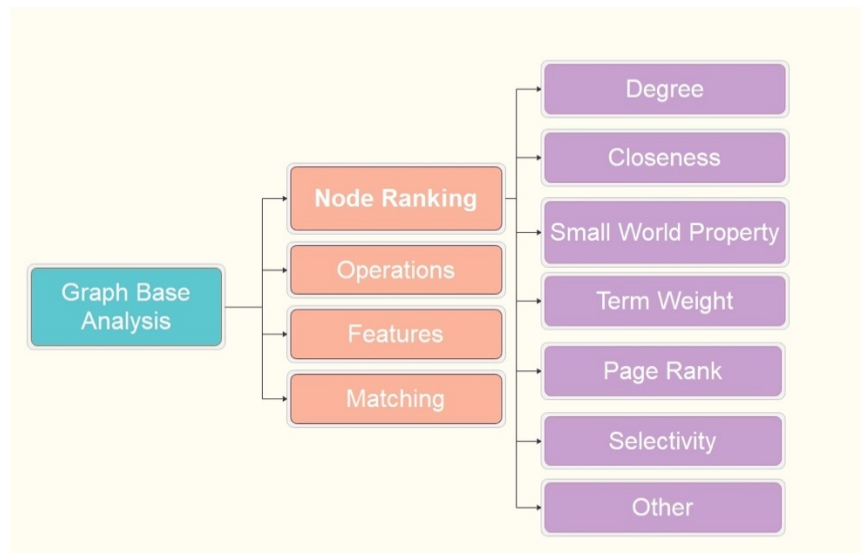


Figure 7. Classification of graph-based methods

For the analysis of the graph, there are many techniques which are used so far. The co-occurrence graph is used for most of the study since it is easy to execute and create also with two or more words at the same time. **Ohsawa et al.** [11] KeyGraph is a suggested technique for automated indexing using co-occurrence networks built from metaphoric. This approach is based on segmenting a graph that represents the co-occurrence of phrases in a text. KeyGraph proven to be a content-aware, context referencing mechanism. **Xialong Wang et al.** [12] describes the sentiment analysis in Twitter, our data is crawled from Twitter too. Twitter is a popular platform used over the globe where massive messages with their real feelings posted everyday freely. We can download all the messages with the term using the hashtag (#) symbol ahead of keyword or keyphrase, most people use this hashtag for coarse-grained issues. In this research, they discussed the hashtag-level subjectivity categorization, and this related task tries to instantly assess the overall sentiment orientation for a given hashtag during a specified time frame. They created a hashtag graph model with the use of some relevant hashtags and created a graph with the nodes (hashtags) and the relation between them (those hashtags which are linked with other). With the kind of some algorithm such as Iterative classification algorithm, Enhanced boosting loopy propagation and others they reached the result of positive and negative hashtags related to a particular hashtag. **Jinghua Wang et al.** [13] worked on keyword extraction using PageRank. They used two algorithms WordNet and PageRank to propose their work,

with the help of WordNet they represented a rough, undirected, weighted and semantic graph. They weighted the graph with the relationship of synsets, then they apply the PageRank algorithm on that rough graph to prune the graph and again applied the PageRank algorithm to a pruned graph for the keyword extraction. The WordNet defines synset as vertices and relation of synset as edges and weights edges based on the genetic similarity of a linked sentence. The Pagerank algorithm is prescribed in the aimed graph, so by the help of WordNet, they are applying this algorithm on the undirected graph. **Marina Litvak et al.** [14] proposed work for graph-based keyword extraction, firstly they compared for issue that will affect, two innovative techniques, supervised and unsupervised techniques, and then they train the classification algorithms on a reduced group of documents. They execute the HITS algorithm on document graphs under the assumptions that the top-ranked nodes should represent the document keywords. They also find the accuracy, F-measure during the simple degree-based ranking. Additionally, they suggest that it is sufficient to perform only the first iteration of HITS rather than running it to its convergence. For both the approaches they applied a graph-based representation of text documents, and those representations may vary from syntactic like words connected, simple and co-occurrence relation to more complex like semantic relation. For the current representation during their evolution, they used simple graph representation, all the pre-processing of data done during the crating of a simple graph. For the keyword extraction, they used both approaches supervised and unsupervised. In supervised they trained the extraction according to the approach, each link in the document graphs falls into one of two categories: YES, if the agreed term is contained in the document extraction conclusion, and therefore not otherwise. Another one is unsupervised; these algorithms recursively assign a weight to every element for determining how the page is. HITS algorithm applied to graph document to examine its performance and produce two types of scores: an 'authority' rating and a 'hub' rating. The rationale underlying this finding is that the overview should include as many words as possible that are closely relevant with other words in the text (excluding sentences). Running HITS, it is superfluous to its integration because it does not enhance the original outcomes of the qualification evaluation.

CHAPTER 4

METHODOLOGY

4.1 Introduction

The aim of extracting keywords is to automatically identify a group of phrases that best represent the content of the document. Keywords exist independently of any corpus and can be used to information retrieval (IR) system. In the recent years keyword extraction has caught the interest in the field of research. Although it commonly works in a single document it can also be used for more complex tasks such as for the integral website or for automatic web summarization and for the whole collection etc. There are so many applications of keyword extraction in the research area like as

- Topic detection.
- Website categorization or clustering.
- Indexing and automatic summarization.
- Document management.
- Constructing domain-specific dictionaries.

It sums up that keyword extraction is an important task in the field of text mining. Several methods exist regarding the graph such as attributed, multi-valued, directed, undirected etc. I chose the co-occurrence graph because it was not taken into consideration yet by other researchers.

4.2 Why Graph-based approach?

Rate of increase of data over the internet suffers user to maintain, visualize and analyze data, a graph is the best way to represent which data is relevant to another set of data. Graphs represent a statistical model, that allows for the exploration of connection and structural information in a very effective way. The graph represents

the content where terms identify the vertices, and their connections identify the edges. Words relationship edges could be built by utilizing diverse scopes of information or interaction among terms for graph generation, such as co-occurrence relation, syntax relation, semantic relation, and other possible relations.

4.3 Graph Co-occurrence

Graph Co-occurrence is a collective interconnection of terms depends on their paired presence within a specific unit of text. The generation and visualization of the co-occurrence graph have become practical by the coming of electronically stored text amenable to text mining. It is created by the pair of terms called neighbors such as A, B and C called co-occur if the relationship will be A to B and B to C. The working applications of co-occurrence graph are PubGene (addresses the interests of the biomedical community) and NameBase (the website shows the personal names in the newspaper and other text).

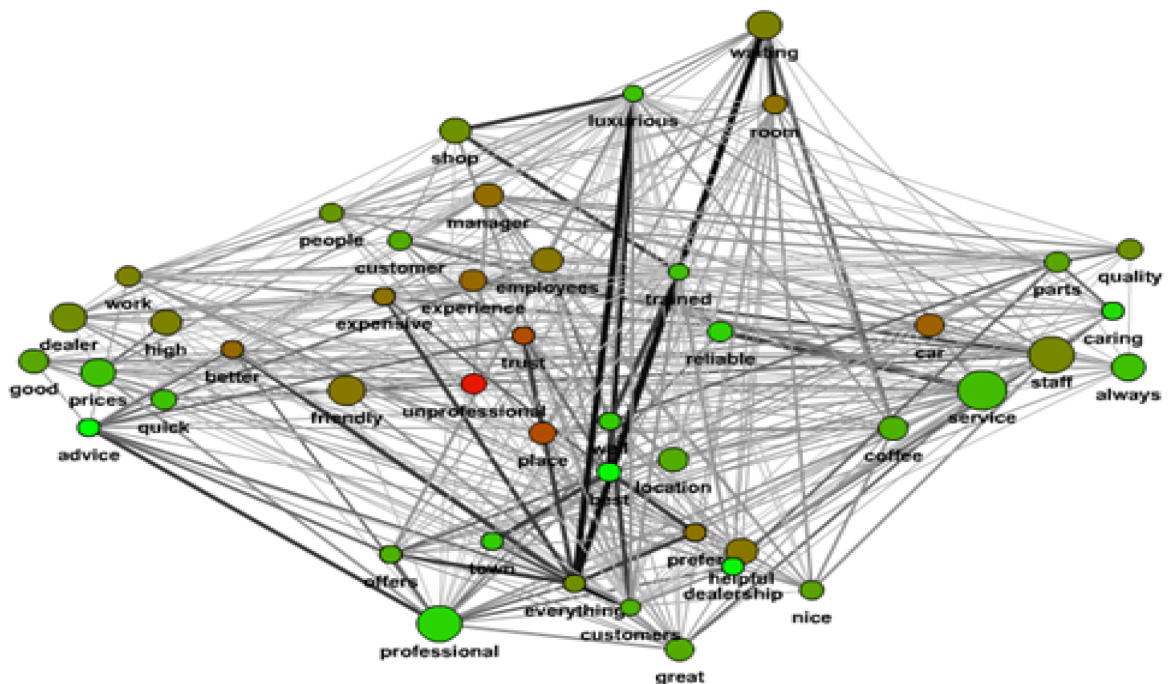


Figure 8. Co-Occurrence Graph

4.4 TextRank Algorithm

TextRank is an extractive and unsupervised text summarization technique, it assigns a value to the phrases that are relevant for the content based on the structure of the material. It is graph –based keyword extraction algorithm which uses Google PageRank algorithm. In this algorithm, the document is considered as graph and keywords are considered as vertices of the graph, the edge between vertices is indicated by the defined number of instances of items inside the dimension window. Finally, the PageRank algorithm is utilized to compute the global weight for each vertex in the graph. The fundamental concept of using TextRank is that if a word co-occur repeatedly with different important words, can be a significative word. The words having low TextRank score is treated as unimportant. It is completely unstructured and depends solely on the input text to generate an extracting description, representing a summarizing methodology more like with what people do while creating an abstraction for a given material. It separates the text into sentences based in train model. It builds a sparse matrix of words and counts it appears in each sentence and normalize each word with tf-idf. After it constructs the similarity matrix between sentences. It is basically derived from the PageRank algorithm, so it also follows the working flow of the PageRank algorithm, with this algorithm it calculates the score of each word in a data. Textrank gets the best accuracy and F-measure throughout all systems, even though recall is low, presumably due to constraints placed by our technique on the number of terms chosen. The significant element of Textrank would be that it ranks most of the words in the sentence, which implies that it may be simply modified to retrieving extremely short texts. Textrank is successful in finding the most essential meaning of a text based only on data provided from the content itself. Finally, another vintage of this algorithm is the fact that it doesn't need any training corpora, as a result of this it is easy adaptive to new situations like language or domain.

$$S(V_i) = (1 - d) + d * \sum_{j \in In(V_i)} \frac{1}{|Out(V_j)|} S(V_j)$$

Therefore, d is indeed the attenuation ratio, which may be chosen among 0 and 1, and serves as an integration component in the system. The attenuation ratio is often set at 0.85 (Brin and Page, 1998), which is exactly the figure we use in our method.

4.5 Generating Graph-based Data

For generating the graphical data, I have crawled the data from Twitter. Twitter gives us the shortest text and for the keyword extraction, we need short text that's why I opted Twitter and it is also used worldwide. I have crawled the data for different hashtags, and I want to convert that tweets data into a graph. Here is the methodology of generating data then convert that data into a graph and finally to apply the TextRank algorithm to determine the value of every caption in a text.

4.5.1 Crawling Data from Twitter

Data from Twitter may be accessed in two ways: Typical search API (a), Corporate search API (b). The regular search API is completely free for using. A normal search API delivers a list of relevant messages that fit a specified inquiry. It retrieves messages from the past 7 days. Using a common search API to gather comments regarding multiple activities.

4.5.2 Tweets Pre-processing.

Due to the obvious existence of extraneous stuff like sentences, commas, and URLs, the tweets must be processed. Because hashtags could be used to build relationships among elements, they must be divided.

4.5.3 Construction of Co-occurrence Graph

After pre-processing of tweets, I have saved the tweets into a text file individually. In the text file from the first tweet to the last tweet every word will make a co-occur graph by considering his next word as a neighbor. It exploits a basic neighbouring relationship occur when 2 words in a phrase are nearby. With this technique, we are converting our text into a graph, and we will get graphical data as our input.

4.5.4 Normalization of Matrix

After constructing the graph, now I am done with the graph-based approach, and I must extract the keywords for the calculate accuracy of extracted keywords. I need to solve the graph data first and I created the matrix because the matrix is the solution of the graph. After the creation of matrix, I normalized the matrix to make stochastic matrix (divide the sum of the row to the individual elements of a row in a matrix).

4.5.5 Keyword Extraction

Using that stochastic matrix, I applied a TextRank algorithm on the matrix to extract keywords and after the extraction it calculates the score of every word that helps to find the most valuable words in a corpus. With the help of those keywords, we will find the accuracy of words in the corpus, such as which are relevant, and which are irrelevant to the heading.

4.5.6 Calculating Precision

For the accuracy, TextRank gives the best precision as compared to the other algorithms. I chose counted words in every data set for the three times in increasing order and calculated the different results according to keywords.

4.5.7 World-Cloud Constructing

Word-cloud is the description of the extracted keywords with good precision. It is generally a graphical visualization of topmost extracted keyword.

CHAPTER 5

EXPERIMENTS AND RESULTS

5.1 Introduction

To experience that how we get the desirable results from our above-proposed methods and to assess the effectiveness of the planned methods, I applied the methodology for the keyword extraction on graph-based approach on the real-world dataset. I used many algorithms to match the efficacy of the approach with others and chose best algorithm for a suitable result.

5.2 Creating Dataset

It is very difficult to get a graphical dataset on the internet, but there are many datasets for which we must request the authors and owners to get that dataset. So, want to overcome this problem I have created my own dataset because I did experiment on real tweet dataset.

5.2.1 Data Collection

The data was gathered from social twitter network, Twitter is microblogging service, which permits its user to post tweets, a status message which can have maximum 140 characters which usually use to carry personal views, news information, events information and information related to the different topics. Because it's just so usual for internet users to enter the statement in not proper structure and they do not give importance to the grammar to write the proper sentence.the most important to them is to spread their views rather thanwritingGrammarly correct sentence and correct typing, and social network users are now adopted in reading and understanding of such unstructured sentence and Grammarly corrected sentence in social networks. So, because of this, all the time we get tweets from the twitter's app, it will contain punctuations, stop words, hashtags, abbreviations, and slang language. This creates a

bottleneck for the processing of such data. So, for this reason, we consider the natural language preprocessing for our tweet dataset to remove those unnecessary part of tweets which either can affect our result and degrade the performance or does not help the performance improvement and add more to improve the feature extraction. After this phase, we got preprocessed tweet dataset. hereafter we called it corpus which contains preprocessed tweets without having an unnecessary part as like original tweets, and then we did different feature extraction from the corpus which after we use for the generation of the social graph which has the form of an attributed graph. after these processes, we use to visualize the result of each step, as almost all our results are in the form of a graph. Hence, we use networks python library to plot the graph for our resulted edges pair. Below, it explained in detail for each step:

Table 2. *Tweets related to various events*

Events	No.of tweets
#AmericanIdol	100
#NotreDame	100
#GoodFriday	100
#RedSkins	100
Total Tweets	400

As I collected 400 tweets from different events now, I want to preprocess tweets with NLP techniques.

There are four of them: the client key (API key), the client secret (API secret), token access, and the secret token access required for access the twitter account for crawling data. I used these keys in the python code and crawl all the relevant tweets of one week regarding the topic. The advantages of crawling data are you get to gather data you want, and the disadvantage of crawling is your traffic may be identified as abusive or suspicious and blocked and you may be constrained by your limits in bandwidth, processing or storage. The API used in this code will give you the crawling data of past one week, the past data more than one week we can't crawl by this code for crawling of two weeks we have to change the API. Python is a very easy way to crawl the data from social media because there are so many libraries are defined and it is widely used for data science (data manage, access and retrieve, etc.).

I used the algorithm of data crawling from twitter in python to crawl the tweets of different events.

Algorithm 5.1: Crawling the tweets from Twitter in python.

Require: crawl the tweets using personal keys of account for different terms related to the hashtag.

1. START
2. Construct and enter the keys.
3. **Contact** by authentication.
4. **Match** the given hashtag over Twitter.
5. **Create** a file (.CSV).
6. **If** the file doesn't exist.
7. **Retrieve** the tweets of one week by given hashtag.
8. **Write** it into the excel file.
9. **Close** the file.
10. END

Here it is the algorithm of crawling data.

```

import tweepy
import csv
import os

okey = "lTt7FQv7Vd3mDug2zKEFFa18M"
osecret = "7nbEpVT3VLptDw2Kt5oVb4HgFP4B0REAbXvQxmuGacYb1818K"
atoken = "2217386142-N92g6ySIWdL2m2Mf8B0if9A320wP9jOQZ24gF"
asecret = "oqg1HCyJazFKbGdWLoY5WqVuvWYq77F404cz11Dqqa"

OAUTH_KEYS = {'consumer_key':okey, 'consumer_secret':osecret,
              'access_token_key':atoken, 'access_token_secret':asecret}
auth = tweepy.OAuthHandler(OAUTH_KEYS['consumer_key'], OAUTH_KEYS['consumer_secret'])
api = tweepy.API(auth)

file_exists = os.path.isfile('0dataset\americanidol.csv')
csvFile = open('americanidol.csv', 'ab')
fields = ('Tweet_Id', 'Tweet_Text', 'Tweet_authorscreen_name', 'Tweet_author_id', 'Tweet_created_at', 'Tweet_coordinates', 'Tweet_source', 'Tweet_user_verified', 'Tweet_retwe')
csvWriter = csv.DictWriter(csvFile, fieldnames=fields)
if not file_exists:
    csvWriter.writeheader()

c = tweepy.Cursor(api.search, q="$americanidol", since="2019-04-14", until="2019-04-19", lang="en", tweet_mode="extended").items()

count=0
while True:
    tweet = c.next()
    for tweet in tweepy.Cursor(api.search, q="$pol", since="2019-04-14", until="2019-04-19", tweet_mode="extended").items():
        print (tweet.id_str, (tweet.full_text.encode('utf-8').replace('\n', '').replace('\r', ' ').decode('unicode_escape').encode('ascii','ignore').strip()), tweet)
        csvWriter.writerow({'Tweet_Id': tweet.id_str, 'Tweet_Text': (tweet.full_text.encode('utf-8').replace('\n', '').replace('\r', ' ').decode('unicode_escape')).
        count +=1

    except tweepy.TweepError:
        print("Whoops, could not fetch more! just wait for 15 minutes :)")
        time.sleep(900)
        continue
    except StopIteration:
        break
csvFile.close()
print(count)

```

Figure 11. Crawling of Data by Python

5.2.2 Natural Language Pre-processing

Content can take several forms, such as a range of specific words, phrases, or many sentences containing special characters. As a result, the content is semi-structured or unprocessed information that does not live in a specific field or entry, thus we must change the text into some pattern that a program can consume, which would be a difficult task. There have been 4 distinct sections that assist us in transforming the data into a format that the method can handle:

- Cleanup entails deleting less important sections of the text by eliminating stop - word, considering texts with capitals and symbols, and other aspects when utilizing it in the software.
- The implementation of a plan and design to texts is referred to as annotations. Structured markup and portion of speech labeling are examples of annotations.
- Standardization entails the translation (mapping) of method concepts or language decreases via Stemming, Lemmatization, as well as other types of normalization.
- The decision making in relation analytically exploring, modifying, and extrapolating from the data in order to do feature extraction.

Here for tweets dataset, we did text data cleaning process which is consist of the following process:

1. URLs removal
2. Punctuation removal
3. Stopword removal
4. Stemming

5.2.2.1 URLs Removal

In order to remove the URLs (unified resource locators) or web address. Where it does not carry any information to help our clustering process improvement. So, in closing first we remove the available URLs in tweet dataset.

5.2.2.2 Punctuation Removal

As like all language writing use punctuation, in English language writing we also use punctuation which help the readers to understand the text as like question mark(?) use for interrogative sentences ,exclamation mark(!),dots(.) use at the end of paragraph or sentence and comma which use to separate various part in a single sentence. Her feature extraction from tweets we do not consider this punctuation. So used to remove the punctuations.

5.2.2.3 Stopword Removal

The bulk of terms in a particular text are linking pieces of a phrase instead of displaying key topics, arguments, or purpose. Words as “the”, “and” other English stopping words which have the connecting job in a sentence could be eliminated by matching the content to a block set of words. We used Python programming language for our implementation which has NLTK (Natural Language Toolkit) library for text data pre-processing and NLTK library by default has the list of stop words for the Englishlanguage, where we used the list of default stop words, and we also added some stop words into list which were not available after we noted in our tweet dataset.

5.2.2.4 Stemming

This approach is used to determine a word's root/stem. The phrases connect, connected, connecting, and connections, for instance, can all be derived from the term "connect." This technique's goal is to eliminate different suffixes, minimize the number of words, have exactly matching roots, and save time and memory storage and it helps to have a single root for different variants of a term. stemming can improve the features extraction, specifically the TF-IDF a feature that we consider in our implementation. We use the ported stemmer algorithm in our implementation.

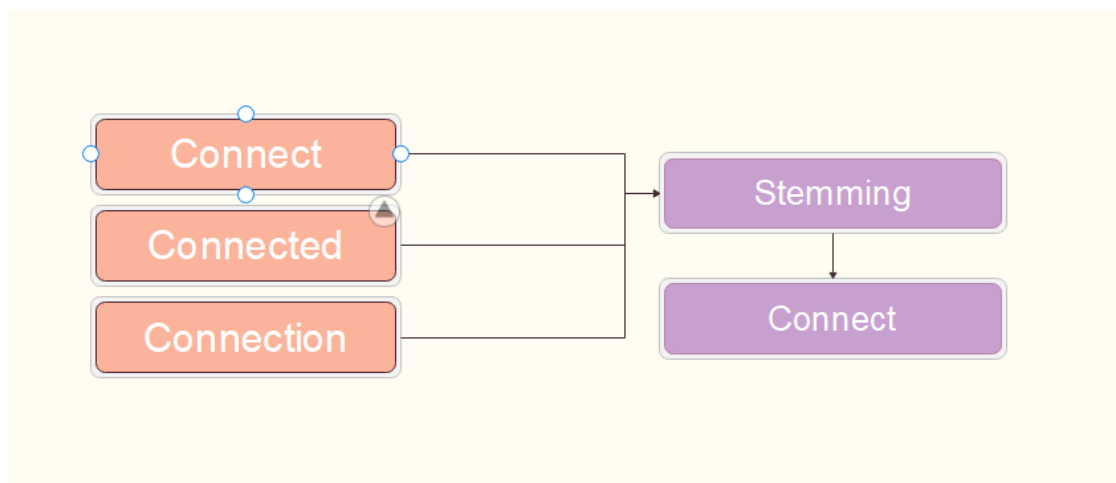


Figure 12. Stemming

Stemmer Porters Porters filtering method [11] [12] is a common stemming algorithm that was introduced in 1980. Many changes and improvements have indeed been performed and recommended to the core algorithm. It assumes that the English suffixes (about 1200) are generally made up of a collection of smaller and manageable suffixes. It consists of five phases, and rules are implemented within each step until one of them meets the criteria. If a condition is granted, the term is deleted, and the following methodology is carried out. The resulting stem is given at the end of the 5th phase.

5.2.3 Case Folding

Users do not bother about how they type the words like the small letter, capital letter or title letter. Same here in our dataset some users type the hashtag in capital letter, or they differentiate the two words that write together by the capitalization of the first character of each word.

i.e., #AmericanIdol, #NotreDame. Therefore, our comparisons are case-insensitive.

So, we normalized the text for comparison, for this purpose we did case folding, where we transform all terms to lowercase and then we did case insensitive comparison.

5.2.4 TF-IDF

TF is an abbreviation for term frequency, whereas IDF is an abbreviation for inverse document frequency. TF is the frequency of a term in a corpus and IDF is the ratio of $\log N/n$. N is the total number of documents in the corpus and small n is the total number of documents have that term. TF-IDF is the cross product of TF and IDF and its numerical statistic use to determine the importance of a term in the corpus. mostly use a weighting factor in information retrieval, text mining techniques, and user modeling. The TF-IDF scoring for term increases when the number of documents increase that have term i . The formula is as follow:

$$Tf - Idf_i = Tf_i x \log\left(\frac{N}{n}\right)$$

After the NLTK process our data looks like in the below picture.

```

1 notre dame fire criminal fire started according lci french tv centuries single fire notre dame https co eqayulz
2 dame emma thompson joins extinction rebellion climate change protests flying london los angeles https co szt cv cg news https co bwlqg uv
3 dame edna dropped barry humphries comedy awards fears masquerading comedian contravenes comedy rules
4 california getting wild week time ticktock wrestlingpw supportindywrestling dame wildchild feminineisthenewmasculine damesarmy glamlife rainbownicorn https co gc
5 damn time dame lillard getting credit deserved underrated damianlillard dame lillard nba nbaplayoffs trailblazers blazers
6 dame el comit olimpico internacional donar medio milln de euros para la restauracin de la catedral de notre dame https co fojss
7 random am telling dame dame lillard top favorite players time rd damn shame dame mostunderrated game period
8 notredamenotre dame le maroc annonce une contribution financire pour la reconstruction de notre damemerci notre roi tjrs prt pour aider les franais sont nos amis
9 hi am graphic designer am providing special image background removal image free sample https co wframard firevrr amazon ebay ebayeseller dame ajax baramufc pemilu
10 uploaded episode notre dame fire means speaker dame fire macron notre notredame notredamefire paris parisfire https co ddiugwyh
11 priceless art relics saved notre dame firethe crown thornstunic saint louisrose windowgreat organ bellsbronze statues twelve apostlesmays notre dame tebeq tuesbe
12 reconstruction de notre dame les ingalits sociales en france expose en un clic millions millions ils les font pour obtenir des allgements fiscaux cest une honte ;
13 literally retweet share world idk care reach understand lyric dame talentshow rappers thursdaythoughts inspiration https co gw ctheqie
14 heard notredame burning gotten day crying couldn video happier times https co teoeortbz dame
15 google warns employees measles exposure silicon valley headquarters report usa rosslynch themagicians jamesharden dame ripcity jam almurray travel instagood folk
16 notre dame fire terrible reason burn bitcoin instablockchain instavenezuela btc ripplenews cryptocurrencymarket decentralized trading ltc usd cny qrl retweethttps
17 cathedral notre dame represent cathedral notre dame meaning name english mother reference mary mother jesus visit https co ses lmmvsg notre dame othernews fire ct
18 ho caricato un nuovo episodio la bellezza delle cattedrali gotiche su speaker arte cattedrale dame notre parigi https co gbjaipm
19 beautifulself photoshoot am momthatshouldmodel breastcancersurvivor americasnexttopmodel oneboob mastectomy https co bxr uvjy dame pinupmodel https co vhw eha
20 une jeune dame se construit dans atelier de lor dame jeune demoiselle sculpture portrait chapeau workingprogress wire fildefeer artiste artwork lineart atelierlor t
21 drone footage extent damage notre dame cathedral fire dpreview https co iudfgxptv
22 ore efficient decision social policy eus support notre dame reconstructionhttps co yhaez ct
23 uploaded episode radio itvt interview lindsay stewart co founder ceo stringr speaker content dame licensing news notre ott stringr videographers https co ygbgpyc
24 dame lillard fuckin beast nba proud af trailblazers fan own ripcity lillardtime dame rosecity loyalty oak pdx blazers mvp steamdeserving
25 love minute russ mighty disrespectful damedollar saying wasnt allstar caliber hey youre field youre field love em dog dame ripcity https co dhmvl hry
26 looking forward seeing army thisisoutbreak saturday autismawareness outbreakwrestling dame wildchild feminineisthenewmasculine damesarmy glamlife https co gljicvt
27 worlds largest bitcoin exchange help rebuild notre dame rebuild notre dame https co fgbu lhq https co cugbulzq
28 wonderful washington usa westbrook lillard dame ripcity jamalmurray travel instagood follow fashion photography uk canada instagram art california https co dluuw
29 bitcoin faithful ignore crypto donation drive rebuild notre dame rebuild notre dame https co ywfjzlc https co uomgqocj
30 pakistan stands france solidarity france notre dame brutal incident represent pakistani minorities solidarity programme marry colmar france https co tunrzkmaa
31 moving forward game life afpw provrestler dame wildchild feminineisthenewmasculine damesarmy glamlife rainbow unicorn warrior https co bohndvrapp
32 pictures visual history notre dame dame notre history https co baukyjl https co grzmqm nu

```

Figure 13. Pre-Processed Data

After pre-processing of data, I must convert the dataset into graph because I want to represent my data as graph based. There are many kinds of graphs are available, I want to construct a co-occurrence graph because it is not becoming the state-of-art for this approach. It is very easy to construct because it is neighbor relation of the text, word will make the relation to the next neighbor word in the corpus.

```

1 notre dame
2 dame fire
3 fire criminal
4 criminal fire
5 fire started
6 started according
7 according lci
8 lci french
9 french tv
10 tv centuries
11 centuries single
12 single fire
13 fire notre
14 notre dame
15 dame emma
16 emma thompson
17 lci french
18 joins`extinction

```

Figure 14. Co-occurrence of relation of words

Algorithm 5.2: Creating a co-occurrence graph

Require: to make a pair of words and construct a graph of neighbor words simultaneously in the corpus.

1. START
2. Read the line of tweet
3. **Split** the line in strings
4. **If indeed the** length of the string is bigger than 1.
5. **Concatenate** neighbor words to previous.
6. **Until** the last word of text.
7. **Create** a text file.
8. **Append** all the pairs in the file
9. **Close** the file.
10. END

With the help of the algorithm, I created graph-based data and visualized the data by python tool because it is very easy to visualize graph in python. There are so many tools are available to visualize the graph, by any of them we can do it.

```
1 import pandas as pd
2 import numpy as np
3
4 import matplotlib.pyplot as plt
5 import networkx as nx
6
7 def loadData_and_process(InputFile):
8     df = pd.read_csv(inputFile, sep=' ', header=-1)
9     df.drop_duplicates(inplace=True)
10
11     tuples = [tuple(x) for x in df.values]
12     return tuples
13
14 def draw_graph(graph):
15     G=nx.Graph()
16
17     for edge in graph:
18         G.add_edge(edge[0], edge[1])
19
20     graph_pos = nx.shell_layout(G)
21
22     nx.draw_networkx_nodes(G, graph_pos)
23     nx.draw_networkx_edges(G, graph_pos)
24     nx.draw_networkx_labels(G, graph_pos)
25
26     plt.show()
27
28 InputFile = 'G:\\testing\\nodes.txt'
29
30 graph = loadData_and_process(InputFile)
31 draw_graph(graph)
```

Figure 15. Code for visualize of graph

The graph will be constructed like dense graph because there are so many words in a corpus so here are the graphs of datasets.

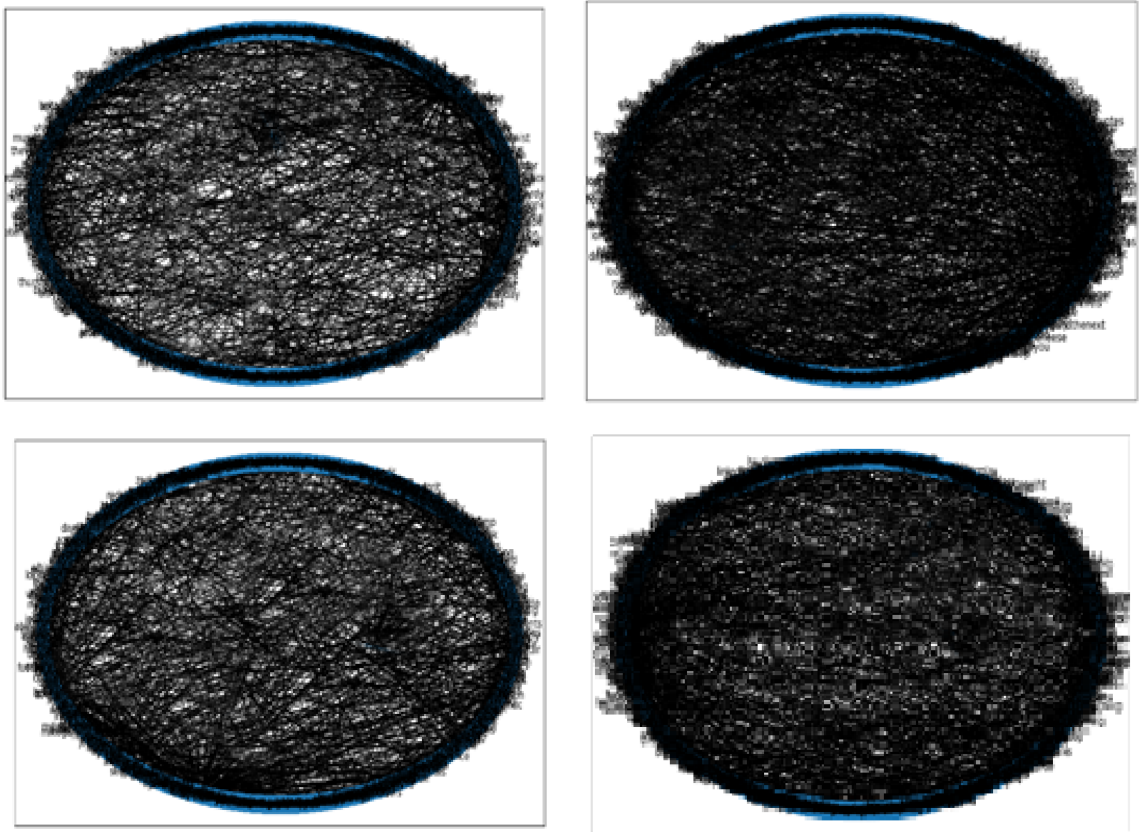


Figure 16. Graph visualization of dataset

After the creation of data in the graph, I will extract the keywords for the extraction process. For doing this I have to apply the TextRank algorithm.

5.3 Implementation of TextRank Algorithm

As I discussed the TextRank algorithm previously, here I will apply this algorithm on my dataset (graph-based data). First, it will create a matrix of the data because it works on the similarity. It will create an incident matrix of the keywords and after creating it will normalize the matrix, because the value is more than one in the matrix, and it will create much confusion while calculating. So, it will do

normalization to make a matrix in the form of a stochastic matrix (in which the sum of the row will be divided by every element present in a row).

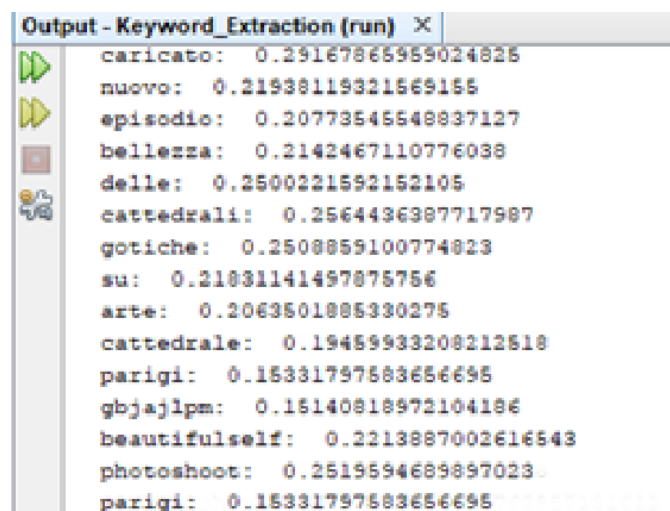
It is comparable to the PageRank method although there is a slight variation between them, PageRank is about the webpages on the internet and TextRank is about the text in a corpus.

Algorithm 5.3: TextRank algorithm

Require: to extract the words and compute the result of every word in a corpus with the threshold difference.

1. START
2. **Read** the graph.
3. **Create** an incident matrix.
4. **Create** a stochastic matrix by normalization.
5. **Adding** the dumping factor.
6. **Perform** $TR(V_i) = \frac{1-d}{|N|} + d \times \sum_{V_j \in adj(V_i)} \frac{TR(V_j)}{L(V_j)}$
7. **Iteration** until the threshold value.
8. **Calculate** the score.
9. END

$TR(V_i)$ is the TextRank score of the keywords.



```
Output - Keyword_Extraction (run) X
caricato: 0.29167866959024825
nuovo: 0.21938119321569155
episodio: 0.20773545548837127
bellezza: 0.2142467110776038
delle: 0.2500221592152105
cattedrali: 0.2564436387717987
gotiche: 0.2508859100774823
su: 0.21831141497875756
arte: 0.2063501885330275
cattedrale: 0.19459933208212518
parigi: 0.15331797583656695
gbjajlpm: 0.15140818972104186
beautifulself: 0.2213887002616543
photoshoot: 0.2519594689897023
parigi: 0.15331797583656695
```

Figure 17. Score

5.4 Precision

After calculating the scores of result I saved the data in excel file to calculate the precision of data. Though the datasets are not predefined, so it is not possible to calculate recall and F-measure. I chose the words on different sets such as a set of 20, 40 and 60, etc. after saving them I mark the 1 to those words which are very close to dataset and 0 to those words which are not close to the dataset and through this process, I calculated the accuracy of the extracted keywords.

Precision is a process of measuring the closeness of a set of answers to the others presented by the experimenter. It is only a subset of requested documentation that are obtained, and it is also the measure of accuracy.

$$\text{Precision} = \frac{Tp}{Tp+Fp}$$

Table 3. Precision of various events

Dataset	k = 20	k = 40	k = 60	k = 80	k = 100
Dataset 1	73%	73%	71.22%	70..15%	69%
Dataset 2	68%	68%	66.11%	68%	68%
Dataset 3	73%	65.20%	66.11%	68%	66%
Dataset 4	73%	63%	63%	64.16	65%

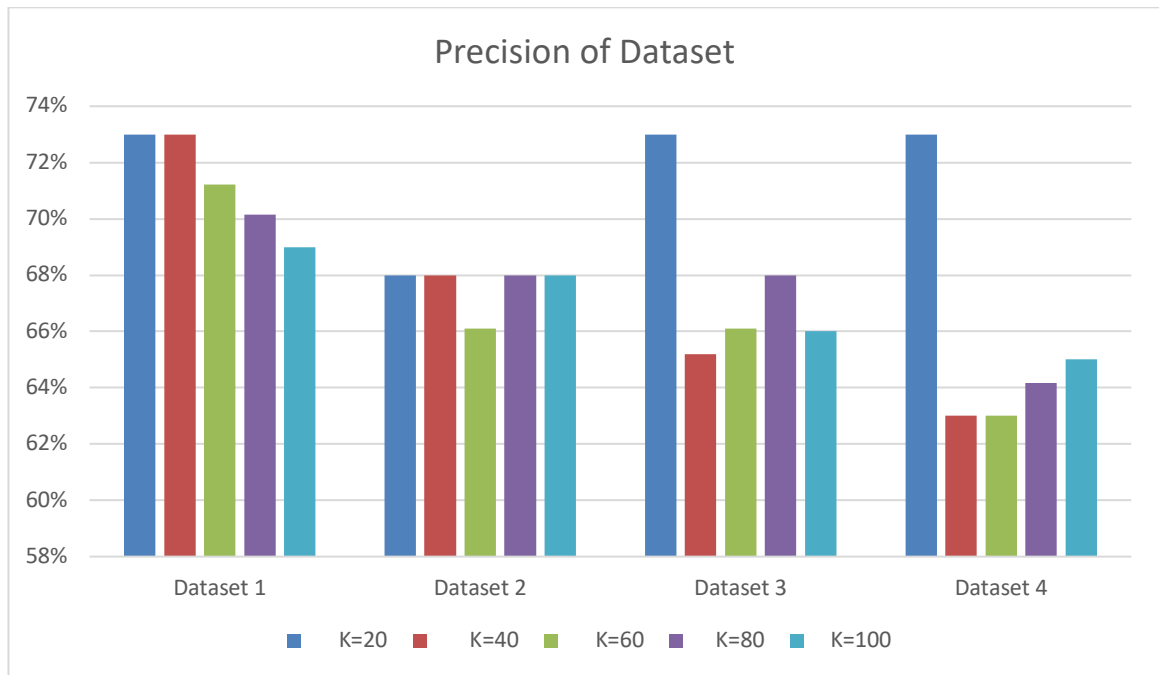


Figure 18. Score

After creating all the results here I used to create a word cloud for the extracted keywords.

5.5 Word Cloud

The prevalence of specific terms discovered in the accessible text after stop words elimination is used to generate word clouds. The most frequented words in the corpus are visualized by the different techniques, the word cloud is one of them. It is common and simple and mostly presented on the internet and websites also with explanation and without explanation. A typical complaint concerning word clouds is that they could impede comprehension since they lack evidence about the link among words. I also represented the keywords of my different dataset on the word cloud.



Figure 19. Word Count of Dataset

CHAPTER 6

CONCLUSIONS AND FUTRE WORK

6.1 Conclusions

Phrases give a detailed approximation of a document's meaning. Graph-based approaches for extracting features are fundamentally unstructured, with the basic goal of constructing a structure of words and afterwards ranking the connections. There are a lot of approaches to extract the keywords. But for microblogging dataset, only a few methods work because of the small length of the texts (tweets). The famous TF-IDF does not work over this dataset because it does not consider the grammatical relations. So, considering the grammatical coherence and cohesion, we have selected a graph-based approach. In this thesis, a detailed description of Existing techniques to information retrieval are addressed, as well as a review of the relevant work on unsupervised and supervised, with such a particular emphasis on graph-based techniques. The findings are based on existing uncontrolled approaches, notably as our method allows no language information but is generated from simple statistics and the text organization is retrieved from the net. At first, we have crawled tweets and created 4 different datasets. A preprocessing is done to make the dataset fit for our model. Furthermore, we have created a co-occurrence graph from this dataset. We applied the graph-based keyword extraction algorithm over this co-occurrence graph that will give the score for each keyword. Even though, there are many proposed algorithms are there for the graphical data such as KeyGraph algorithm, etc. TextRank algorithm works on a graph-based dataset, and it extracts the keywords from the dataset, further it calculates the score of words. Finally, we can rank the keywords according to its importance value. It will also calculate the precision of the extracted keywords. If we want to visualize those extracted keywords, then we can also visualize those important keywords from the dataset through the word cloud. Even though word cloud representation is very common nowadays, there are so many representations are available on the internet. It means to create graph-based data, extract keywords from that dataset calculate the scores of keywords from data and calculate the precision of extracted keywords are the basic assumptions of this proposed work.

6.2 Future Work

There is a vast field of text analytics, and many advance types of research are coming in front of us in today's time so we can extend this work regarding many fields. In future, we can apply our model over the dataset crawled from news portal. This will give the current trend and it will also reduce the effort. Just seeing these keywords, we will understand the overall context of the dataset.

REFERENCES

- [1] Y. Wen and Y. Hui, "Research on keyword extraction based on word2vec weighted text rank," *IEEE*, vol. 2nd IEEE International Conference on Computer and Communications (ICCC), 2016.
- [2] J. Cao, "A way to improve graph-based keyword extraction," *IEEE*, vol. 2015 IEEE International Conference on Computer and Communications (ICCC), 2015.
- [3] M. Islam and M. Islam, "An improved keyword extraction method using graph-based random walk model," *IEEE*, vol. 11th International Conference on Computer and Information Technology, 2008.
- [4] R. Mihalcea, "Graph-based ranking algorithms for sentence extraction applied to text summarization," vol. Proceedings of the ACL Interactive Poster and Demonstration Sessions. , 2014.
- [5] J. Zhou, "Ranking keyword search results with query logs," *IEEE*, no. 2014 IEEE International Congress on Big Data, 2014.
- [6] R. Mihalcea and P. Tarau, "Text rank: Bringing order into tex," vol. Proceedings of the 2004 conference on empirical methods in natural language processing, 2004.
- [7] J. Pitkethly, Introduction to PageRank, Math deliveries.
- [8] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," *Computer networks and ISDN systems* 30.1-7, 1998.
- [9] W. Li and J. Zhao, "TextRank algorithm by exploiting Wikipedia for short text keywords extraction.," *IEEE*, vol. 2016 3rd International

Conference on Information Science and Control Engineering (ICISCE), 2016.

- [10] S. Beliga, A. Meštrović and S. Ipšić, "An overview of graph-based keyword extraction methods and approaches," *Journal of information and organizational sciences*, 2015.
- [11] Y. Ohsawa, N. E. Benson and M. Yachida, "KeyGraph: Automatic indexing by co-occurrence graph based on building construction metaphor," *IEEE*, Vols. Proceedings IEEE International Forum on Research and Technology Advances in Digital Libraries-ADL'98, 1998.
- [12] X. Wang, "Topic sentiment analysis in twitter: a graph-based hashtag sentiment classification approach," *ACM*, vol. Proceedings of the 20th ACM international conference on Information and knowledge management. ACM, 2011.
- [13] J. Wang, L. Jianyi and C. Wang, "Keyword extraction based on PageRank," *Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer, Berlin, Heidelberg, 2007*, 2007.
- [14] M. Litvak and M. Last, "Graph-based keyword extraction for single-document summarization," *Proceedings of the workshop on Multi-source Multilingual Information Extraction and Summarization. Association for Computational Linguistics*, 2008.