Project Report on

“Keyword and Key-phrase Extraction using a graph-based algorithm”

Project submitted
in partial fulfilment of the necessities for the degree of

MASTER OF COMPUTER APPLICATION

By

SUBHANKAR SAHA

Roll No: **002110503029**
Registration No: **160134** of **2021-22**
Examination Roll No: **MCA2340047**

Under the supervision of

Prof. Kamal Sarkar

Department of Computer Science and Engineering
Faculty of Engineering and Technology
Jadavpur University
Kolkata – 700032, India

Jadavpur University
Faculty of Engineering and Technology
Department of Computer Science and Engineering

Certificate of Recommendation

This is to certify that **SUBHANKAR SAHA** (Reg. No.: 160134 of 2021-2022, Roll No: 002110503029) is a student in Master of Computer Application course and the project entitled "**Keyword and Key-phrase Extraction using a graph-based algorithm"** is a bonafide record of work carried out by him, is accepted in partial fulfilment of the requirement for the degree of **Master of Computer Application** from the **Department of Computer Science and Engineering, Jadavpur University** during the academic year **2022-2023**. He has been able to follow all the instructions in a calm and responsible way and successfully carried out his research work. Wish him all the best for his future endeavours.

Prof. Kamal Sarkar (Project Supervisor)
Professor, Dept. of Comp. Science & Engineering

Jadavpur University, Kolkata-700032

Dr. Nandini Mukhopadhyay

Head of the Department, Dept. of Comp. Science & Engineering

Jadavpur University, Kolkata-700032

Prof. Ardhendu Ghoshal

Dean, Faculty Council of Engineering & Technology

Jadavpur University, Kolkata-700032

Jadavpur University
Faculty of Engineering and Technology
Department of Computer Science and Engineering

CERTIFICATE OF APPROVAL

This is to clarify that the project entitled "**Keyword and Key-phrase Extraction using a graph-based algorithm**" has been completed by **SUBHANKAR SAHA.** This work is applied under the supervision of **Prof. Kamal Sarkar** in partial fulfilment for the award of the degree of **Master of Computer Applications** of the **Department of Computer Science and Engineering**, **Jadavpur University**, during the academic year **2022-2023**. The project report has been approved because it satisfies the tutorial requirements in respect of project work prescribed for the said degree.

.....

Signature of Examiner 1

Date:

.....

Signature of Examiner 2

Date:

Jadavpur University
Faculty of Engineering and Technology
Department of Computer Science and Engineering

Declaration of Originality and Compliance of Academic Ethics

I hereby declare that this project contains original work by the undersigned candidate, as a part of his Master of Computer Applications (MCA) studies.

All information during this document is obtained and presented in accordance with academic rules and ethical conduct.

I declare that, as required by these rules and conduct, I have got fully cited and referenced all material results that don't seem to be original to the current work.

I also declare that this can be a real copy of my thesis, including any final revisions, which this thesis has not been submitted for higher degree to the other University or Institution.

Name: **Subhankar Saha**

Registration No: **160134** of **2021-22**

Class Roll No: **002110503029**

Examination Roll No: **MCA2340047**

Project Title: **"Keyword and Key-phrase Extraction using a graph-based algorithm"**

_____

(Signature of the Candidate)

Jadavpur University
Faculty of Engineering and Technology
Department of Computer Science and Engineering

Acknowledgement

With my most sincere and gratitude, I would like to thank **Prof. Kamal Sarkar**, **Department of Computer Science & Engineering**, my supervisor, for his overwhelming support throughout the duration of the project. His motivation always gave me the required inputs and momentum to continue with my work, without which the project work would not have taken its current shape. His valuable suggestion and numerous discussions have always inspired new ways of thinking. I feel deeply honoured that I got this opportunity to work under him.

I would also prefer to thank the experts involved within the validation survey for this project, **Dr. Nandini Mukhopadhyay**, Head of the Department of Computer Science and Engineering and every one the university faculty members of the **Department of Computer Science and Engineering**.

Finally, I need to express my profound gratitude to my parents and my friends cum classmates for providing me with unfailing support and continuous encouragement throughout my years of study and throughout the method of my project work and writing this project report. This accomplishment wouldn't be possible without them.

Thank you.

_____

(Signature of the Candidate)

ABSTRACT

The technique of keyword and key-phrase extraction involves feeding a text to a computer, which then recommends words and phrases that are relevant to the text's content of documents. The use of keyword and key-phrase extraction techniques is widespread, especially when it comes to information retrieval. This is particularly interesting since individuals employ keywords and key-phrases to access important information.

Numerous methods have been created for English texts. However, very few attempts have been made to extract Bengali keywords and to understand context. Four distinct datasets that were gathered from news portals are used in this study to demonstrate a Graph-based algorithm for keyword and key-phrase extraction.

Additionally, we have demonstrated if studying co-occurrences permits documenting the creation of each Bengali text. However, manipulating it takes more time and frequently results in messy visualisations.

Index Terms: Word Clouds, Context Learning and Bengali Keyword Extraction

TABLE OF CONTENTS

Chapter 1
INTRODUCTION

Keywords are precise words that sum up the key points of a text, document, or piece of information. They act as crucial descriptors or tags that make it easier to recognise the major themes or subjects included in the content. Search engine optimisation (SEO), information retrieval, content classification, and text analysis are just a few areas where keywords are vital.

Keywords are essential in the context of SEO in order to optimise site content and increase its exposure in search engine results. Website owners want to increase organic search traffic by deliberately adding pertinent keywords into the text, meta tags, headers, and other components of their sites. Understanding user search intent, analysing competition, and picking phrases that are consistent with the content and goal are all necessary for choosing the proper keywords.

Key-phrase, sometimes referred to as a multi-word or multi-token phrase, is a group of words that express a particular idea or meaning. Key-phrases are frequently more illuminating and indicative of the content than single words. They effectively summarise text segments and may be used for content analysis, information retrieval, and summarising activities. Key-phrases can range from straightforward noun phrases (such as "machine learning algorithms") to more intricate expressions including several speech sounds and grammatical elements (such as "natural language processing techniques for sentiment analysis").

Various methods, such as statistical techniques, graph-based methods, and machine learning models, are frequently used to extract key-phrases from text. The objective is to determine the key words and phrases that best express the essential points or subjects covered in the text.

With the widespread use of the Internet as a source of information, keywords (or key-phrase) have emerged as a crucial tool for readers to rapidly comprehend the content of a document and understand its context [1]. Finding the words (or phrases) in any text that most accurately express the fundamental concept of that document or topic is done by keyword extraction (or key-phrase extraction), also known as automated keyword extraction (or automated key-phrase extraction). The terms selected for keyword extraction (or key-phrase extraction) are those that are specifically referenced in the document. Without reading the entire document, keyword extraction (or key-phrase extraction) enables the reader to comprehend the summary or at the very least the main concept. As a consequence, the potential readers do not spend their time by carefully reading the useless materials. Users may typically find similar articles to an event quickly by Googling the keywords (or key-phrase). Users must unavoidably navigate through a great volume of news and superfluous information

in order to comprehend the context of the event. Given the need, keyword extraction (or key-phrase extraction) techniques have exploded in popularity and have helped to make a human's laborious job easier. Ping-I and Shi-Jen claim that there are two main categories of keyword extraction techniques: statistical approaches and machine learning approaches [2]. In this paper, we primarily concentrated on the graph-based extraction method for Bengali keywords and Bengali key-phrases. Although Bengali is among the seventh most spoken languages in the world, relatively little study has been done on Bengali keyword extraction (or key-phrase extraction) [3]. Because automated keyword extraction (or automated key-phrase extraction) makes it simple for readers to determine the category or genre of the documents, there is a huge market for Bengali keywords and key-phrases. As a result, it takes a lot of time and effort for the reader to comprehend the teachings in the paper. There have been several studies on word extraction using image processing, but few studies on keyword extraction (or key-phrase extraction) from documents for the Bengali language.

An effective and precise keyword extraction (or key-phrase extraction) algorithm speeds up the distribution and circulation of information in the network while simultaneously lowering the cost of screening and filtering useful information.

A keyword extraction algorithm is graph-based algorithm for keyword and key-phrase extraction. The preferred keyword extraction algorithm is to calculate the score of words (or phrases) on the basis of Graph. In specific, each document is represented as a graph where a node corresponds to a word (or phrase) and the weight of an edge is the Cosine Similarity between two vectors of the corresponds words (or phrases). The vectors of each word (or phrase) are obtained from the fastText.

FastText is a library for learning of word embeddings and text classification created by Facebook's AI Research (FAIR) lab.

Using this graph, we did some calculations (which will be discussed further in this paper) and the high scored words (or phrases) are identified as keywords (or key-phrases).

The remainder of the paper is structured as follows. In part II, a short overview of the relevant literature is provided. A complete explanation of the proposed model is provided in part III. In part IV a short description of the procedures and dataset that were utilised in the study is provided. The part V of the report presents the evaluation and results. Finally, part V's Conclusion and Future Works bring the paper to a close.

A keyword refers to a single word that represents a topic, while a key-phrase consists of multiple words that form a specific query or phrase.

Chapter 2
LITERATURE SURVEY

The extraction of keywords (or key-phrase) is a fundamental step in many difficult tasks, such text summarization, and is therefore important in many areas of natural language processing. This paragraph will represent a few significant literature reviews that have to do with text extraction. It should be noted that relatively few research projects on Bengali keyword extraction (or key-phrase extraction) have been carried out.

- In the realm of NLP, automatic collocation extraction from the Bengali text corpus is crucial. A collocation is the pairing or coalition of two or more words that occur more frequently together in a corpus. Collocations were discovered based on the terms' shared occurrence. When a set of terms was extracted, the combined appearances of the words had to be high. High Word Pair Occurrence (HWPO) and High Word Occurrence (HWO), two fuzzy sets, were taken into consideration for this purpose. For the purpose of determining the Fuzzy Bi-gram index (FBI), values from fuzzy membership were combined. Using this technique, FBI assigned values between 0 and 1 to describe the degree of a word pair as being bi-grammatical, which really means that the collocation would be two words long. When FBI (w1, w2) was greater than a threshold value of 0.5 in the studies, a word pair of (w1, w2) was considered to be a bi-gram [4].
- A layered technique was provided by Abulaish et al. [5] for obtaining the summary and context of the microblogging dataset. They evaluated the model using a number of graph-based techniques, including TextRank, LexRank, and graph-based algorithm for random walks, etc. Finally, they have an average accuracy of 80%.
- A strategy where automatic keyword extraction from the texts was combined with machine learning and statistical methods. After POS data from an article was identified using a Hidden Markov Model (HMM)-based POS tagger [6], the statistical technique was used to extract keywords. After that, each word's score for keyword extraction was determined by using a learning probability distribution. The article is then summarised using the chosen keywords.
- All of the articles' keywords were extracted, and after removing any redundant ones, the output file would only include the keywords. Moreover, For the purpose of extracting precise POS data from the training corpus, the HMM-based POS tagger [7] was used.
- Semi-supervised key-phrase extraction was proposed using fact-based sentiment analysis [8]. Given that most key-phrases are spoken in neutral to positive emotion, it was anticipated that the fact-based sentiment feature will increase key-phrase features in this situation.
- Additionally, a mix of supervised and unsupervised [9] techniques was suggested to benefit from the best aspects of both. It would aid in finding hidden patterns.

- Three stages made up the key-phrase extraction process: key-phrase candidate identification, key-phrase ranking, and key-phrase categorization. Following the storing of key-phrase candidates in descending order according to their relative relevance, each candidate would be chosen at random from the list's beginning and fed through a classifier until N key-phrases were chosen. Deep Belief Networks (DBN) are used in this technique because deep learning has been shown to be more successful than regular learning on a variety of learning tasks [10].
- DBN's initial node weights were pre-trained with Restricted Boltzmann Machine [11] to make them more in line with the data itself.
- A Conditional Random Field (CRF) model, which is regarded as a contemporary sequence labelling model, was another technique for retrieving keys. It is a probabilistic model that may be used to segment and label sequential data in a very effective and efficient manner [12].
- The conditional probability distribution for a given collection of characteristics is often hidden by the Conditional Random Field, an undirected graph model. Conditional Random Field (CRF) was trained using a maximum entropy learning technique. Prior to CRF model training, the documents were moved into the tagging order. Sentence fragment POS labelling was added for a new document. The characteristics were then removed [13].

Chapter 3
PROPOSED METHODOLOGY

3.1. Keyword Extraction Model

Our proposed keyword extraction using graph has several steps (1) Collecting Data, (2) Pre-processing of Data, (3) Graph Creation, (4) Ranking Algorithm, (5) Keyword Extraction, (6) Calculate Precision, Recall, F-measure. A block diagram of the Proposed method is shown in Figure1.

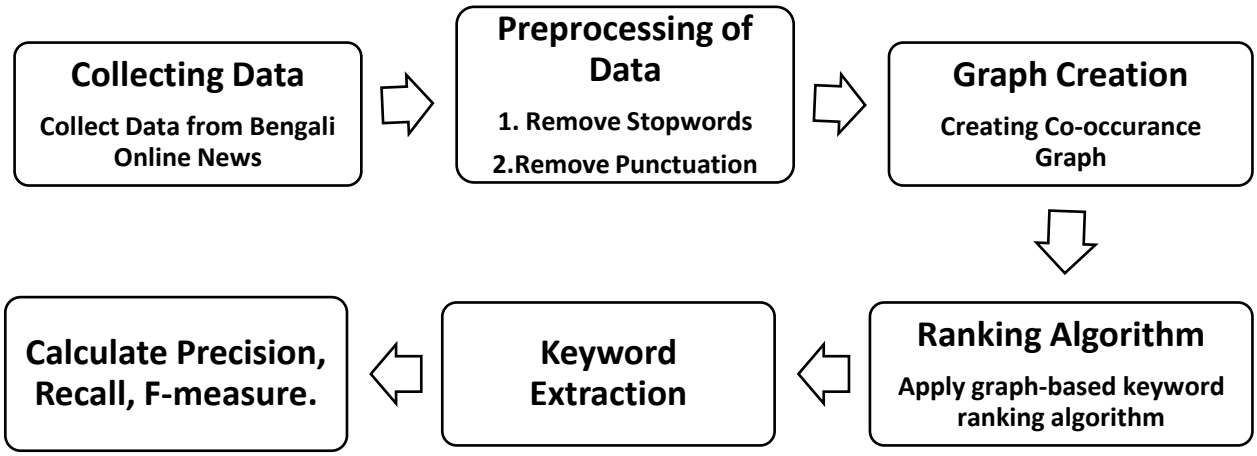


Figure 1

A. Collecting Data

The proposed method's dataset was gathered from a Bengali news portal.

B. Pre-processing of Data

This step includes the process.

Remove Stop Words: Stop words are often used words with little significance. To decrease the noise and prioritise on the most essential data, they are frequently eliminated from text analysis. Bengali Stop Word file is used to remove the stop words.

Remove Punctuation: Commas, periods, and question marks, among others, are essential punctuation symbols for boosting readability and meaning. They provide written language structure and clarity. Sentences without appropriate punctuation might be difficult to understand. Punctuation used correctly enables clear communication and prevents misunderstanding.

C. Graph Creation

Lexical and semantic data can be extracted from texts from many sources using a wide range of techniques. The relationship between words within a text (sentences, paragraphs etc.) is the main emphasis of the co-occurrence notion. Using the pre-processed dataset, we produced a co-occurrence graph. We used the cooccurrence of N = 2 to build the node pair and the creation of edges between each node pair to create the nodes and edges for the graph.

Each node of a graph corresponds to a word of the doc and an edge between two node is the cosine similarity between two vectors of the corresponding words. The vectors of each word are retrieved using FastText.

- **Cosine Similarity:** Similarity measure in data mining refers to the distance between dimensions in a dataset that indicate the features of the data object. There will be a high degree of similarity if this distance is small, but a low degree of similarity if the distance is large.

  Cosine similarity is a data that may be used to assess how similar data objects are, despite their size. Python's Cosine resemblance function may be used to compare two texts for resemblance. Cosine similarity treats each data object in a dataset as a vector. The following is the formula to get the cosine similarity between two vectors:

$$\cos(x, y) = \frac{x.y}{\|x\| * \|y\|}$$

$x.y$ = product (dot) of the vectors 'x' and 'y'.

$\|x\|$ and $\|y\|$= length of the two vectors 'x' and 'y'.

$\|x\| * \|y\|$ = cross product of the two vectors 'x' and 'y'.

Example-

The 'x' vector has values, x = {4, 5, 2, 9}

The 'y' vector has values, y = {2, 4, 1, 1}

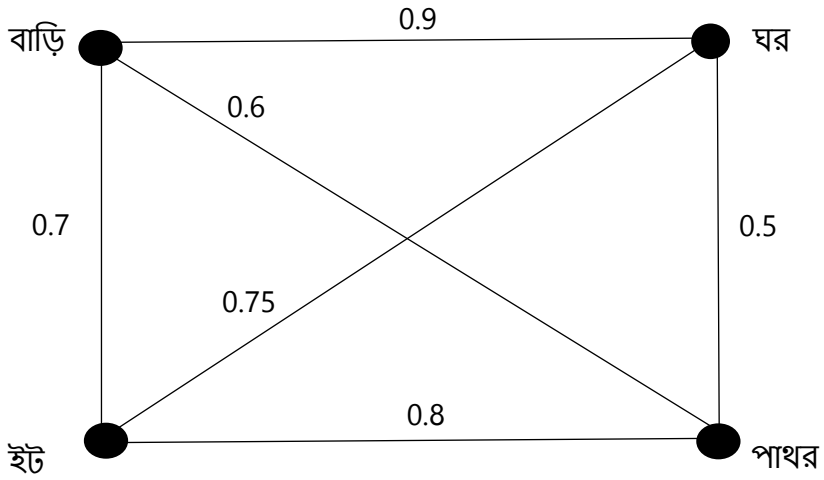
$$x.y = 4 * 2 + 5 * 4 + 2 * 1 + 9 * 1 = 39$$

$$\|x\| = \sqrt{4^2 + 5^2 + 2^2 + 9^2} = 11.22$$

$$\|y\| = \sqrt{2^2 + 4^2 + 1^2 + 1^2} = 4.69$$

$$\cos(x, y) = \frac{39}{(11.22 * 4.69)} = 0.7411$$

- **Missing Vector:** If the word vector of any word in our dataset is not present in the word_to_vec_model, then we calculate the average word vector of all the characters of that word for that word.



Graph

D. Ranking Algorithm

- Step-1: Create Adjacency matrix for the graph

	বাড়ি	ঘর	ইট	পাথর
বাড়ি	0	0.9	0.7	0.6
ঘর	0.9	0	0.75	0.5
ইট	0.7	0.75	0	0.8
পাথর	0.6	0.5	0.8	0

Adjacency Matrix

- Step-2: Calculate Soft Degree of a node

Soft degree of a node
= Summation of all the row elements of the adjacency matrix corresponds to the node

	বাড়ি	ঘর	ইট	পাথর	Soft degree of a node
বাড়ি	0	0.9	0.7	0.6	(0+0.9+0.7+0.6) = 2.2
ঘর	0.9	0	0.75	0.5	(0.9+0+0.75+0.5) = 2.15
ইট	0.7	0.75	0	0.8	(0.7+0.75+0+0.8) = 2.25
পাথর	0.6	0.5	0.8	0	(0.6+0.5+0.8+0) = 1.9

- Step-3: Normalize the Soft degree of a node

$$\textbf{Normalize Soft degree of a node}(\mathbf{i},\mathbf{j}) = \frac{\boldsymbol{Soft\ degree\ of\ node}(i,j) - \boldsymbol{min}}{\boldsymbol{max} - \boldsymbol{min}}$$

Where,
$i =$ Folder name
$j =$ word of the node
$min =$ Minimum of the soft degree among all the nodes of that doc
$max =$ Maximum of the soft degree among all the nodes of that doc

Example:
In the above graph,
min= 1.9
max= 2.25

$$\textbf{Normalize Soft degree of a node}(\mathbf{i},\textbf{বাড়ি}) = \frac{\boldsymbol{Soft\ degree\ of\ node}(i,\textbf{বাড়ি}) - \mathbf{1.9}}{\mathbf{2.25 - 1.9}}$$
$$= \frac{\mathbf{2.2 - 1.9}}{\mathbf{2.25 - 1.9}}$$
$$= \mathbf{0.8571}$$

- Step-4: Calculate Score of a word

$$\textbf{Score of a word} = \textbf{Normalized Soft degree of a node} + \frac{\mathbf{1}}{\sqrt{\boldsymbol{l}}}$$

Where,
$l =$ Sentence no. in which the node word occurs first

Example:

Let, the word "বাড়ি" occurs in the 2nd sentence
Then the value of l for word "বাড়ি" is 2

Therefore,

$$Score\ of\ \text{বাড়ি} = \text{Normalized Soft degree of } \text{বাড়ি} + \frac{1}{\sqrt{2}}$$

$$= 0.8571 + \frac{1}{\sqrt{2}}$$

$$= 1.5642$$

E. Keyword Extraction

The score of each-word in a file are provided by the keyword extraction method. The top k keywords will be chosen, and their compatibility with the topic set will then be determined. We will classify this as a genuine positive if it falls under the topic set, else a false positive. We will manually extract 5 keywords from each file, calculate precision, recall and visually represent the keywords to make it simple to identify the important keywords.

F. Calculate Precision, Recall, F-measure

Precision and recall are performance indicators in information retrieval that relate to data recovered from a collection, corpus, or sample space.

• Precision

In the context of text recognition, precision refers to a system's accuracy in detecting only the pertinent information and rejecting inaccurate or irrelevant information. It evaluates how well the system can produce precise and accurate outcomes. It measures the proportion of true positive outcomes—that is, important information that was correctly detected—to the total of true positives and false positives, or irrelevant information that was wrongly identified. A high precision score means that the system correctly identifies relevant information while minimising the inclusion of irrelevant or inaccurate information, and it also means that the system has a low rate of false positives. This is significant in situations like medical diagnosis, legal document analysis, or information extraction from private documents, where accuracy is essential.

The performance of the system cannot be fully understood by looking at precision alone.It does not consider the possibility of missing relevant information (false negatives), which is measured by recall.

$$Precision_{Top\ K} = \frac{match\ between\ manually\ Selected\ Top\ 5\ keywords\ and\ code\ generated\ Top\ K\ keywords}{K}$$

- **Recall**

Recall is the ability of a system to accurately identify and retrieve pertinent information from a given text in the context of text recognition. It is a gauge of how well the system recognises all the important information contained in the text. The recall measure is frequently used to assess the effectiveness of such systems, especially in jobs where information extraction and retrieval are essential. The system's outputs are compared to ground truth data or a reference set in order to determine recall. The system should be able to detect the right or expected information from the reference data.

The ratio of true positive results—the relevant information that was correctly identified—to the total of true positives and false negatives—the important information that the system missed—is used to calculate recall. It gives a sign of how effectively the system gathers the necessary data. High recall shows that the system is effective in locating and retrieving the necessary data. A high recall score does not necessarily imply good precision because it does not take into account the quantity of false positives (i.e., irrelevant material mistakenly classified as important). Contrarily, precision evaluates how well the system is able to recognise just pertinent information while excluding irrelevant data. Recall and precision are frequently combined to assess the overall effectiveness of text recognition systems.

$$Recall_{Top\ K} = \frac{match\ between\ manually\ Selected\ Top\ 5\ keywords\ and\ code\ generated\ Top\ K\ keywords}{Number\ of\ manually\ selected\ keywords(5)}$$

Another metric that can be used to provide a fair assessment is the F-measure, which combines recall and precision.

- **F-measure**

F-measure is a measure of a test's accuracy. It is calculated from the precision and recall of the test. F-measure is the harmonic mean of the
precision and recall.

$$\mathrm{F-measure}_{Top\ K} = \frac{2 * Precision_{Top\ K} * Recall_{Top\ K}}{Precision_{Top\ K} + Recall_{Top\ K}}$$

$$\text{Average F} - \text{measure}_{Top\ K} = \frac{1}{Number\ of\ Docs} \sum \text{F} - \text{measure}_{Top\ K}$$

Example: For a Doc we manually retrieve these keywords:
"**স্মার্টফোন, ব্র্যান্ড, OnePlus, ডিসপ্লে, ক্যামেরা**"

And our model retrieves these keywords:
"**সম্প্রতি, Pro, প্রিমিয়াম, ব্র্যান্ড, করবে,**
কয়েক, ফোনের, মধ্যেই, স্মার্টফোন, নতুন,
সুপার, 2K, AMOLED, আগেই, অন্যতম"

$$Precision_{Top\ 5} = \frac{1}{5} = 0.2$$

$$Recall_{Top\ 5} = \frac{1}{5} = 0.2$$

$$\text{F} - \text{measure}_{Top\ 5} = \frac{2 * 0.2 * 0.2}{0.2 + 0.2} = 0.2$$

$$Precision_{Top\ 10} = \frac{2}{10} = 0.2$$

$$Precision_{Top\ 10} = \frac{2}{5} = 0.4$$

$$\text{F} - \text{measure}_{Top\ 10} = \frac{2 * 0.2 * 0.4}{0.2 + 0.4} = 0.27$$

$$Precision_{Top\ 15} = \frac{2}{15} = 0.13$$

$$Precision_{Top\ 15} = \frac{2}{5} = 0.4$$

$$\text{F} - \text{measure}_{Top\ 15} = \frac{2 * 0.13 * 0.4}{0.13 + 0.4} = 0.196$$

For precision and recall, we implemented the method on just 50 documents. So, when we want to calculate the average F-measure, we have to add all the F-measures and then divide it by 50.

3.2. Key-phrase Extraction

Our proposed key-phrase extraction using graph has several steps (1) Collecting Data, (2) Pre-processing of Data, (3) Graph Creation, (4) Ranking Algorithm, (5) Key-phrase Extraction, (6) Calculate Precision, Recall, F-measure. A block diagram of the Proposed method is shown in Figure2.

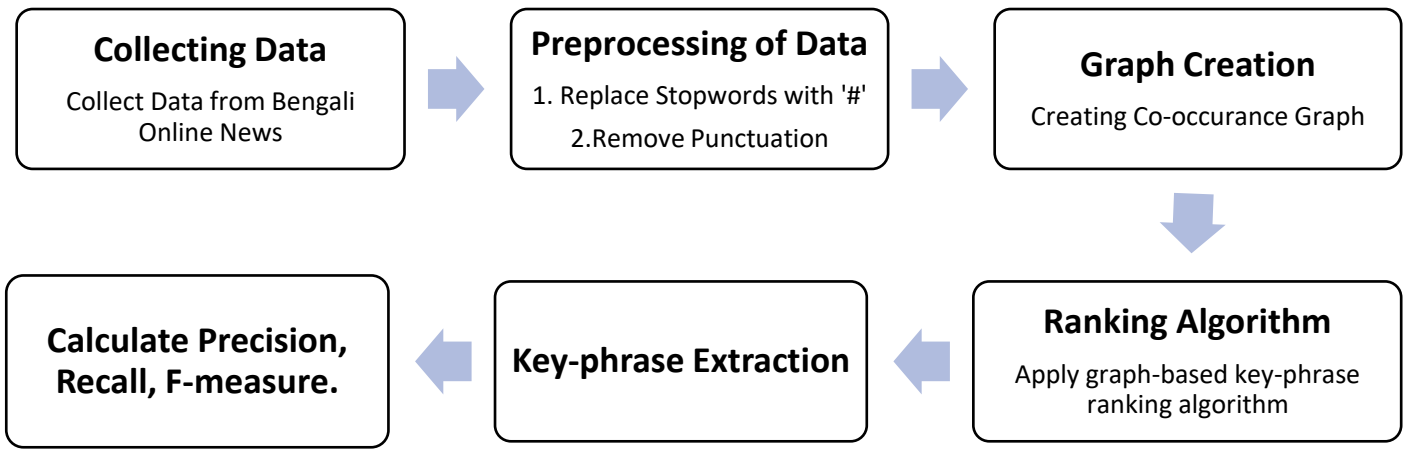


Figure 2

A. Collecting Data

The proposed method's dataset was gathered from a Bengali news portal.

B. Pre-processing of Data

This step includes the process:

- **Replace Stop Words with '#':** Stop words are often used words with little significance. To decrease the noise and prioritise on the most essential data, they are frequently eliminated from text analysis. But here for key-phrase extraction we replace the Stop Words with '#'. Bengali Stop Word file is used to replace the stop words.

- **Remove Punctuation:** Commas, periods, and question marks, among others, are essential punctuation symbols for boosting readability and meaning. They provide written language structure and clarity. Sentences without appropriate punctuation might be difficult to understand. Punctuation used correctly enables clear communication and prevents misunderstanding.

C. Graph Creation

Lexical and semantic data can be extracted from texts from many sources using a wide range of techniques. The relationship between words within a text (sentences, paragraphs etc.) is the main emphasis of the co-occurrence notion. Using the pre-processed dataset, we produced a co-occurrence graph. We used the cooccurrence of N = 2 to build the node pair and the creation of edges between each node pair to create the nodes and edges for the graph.
Phrases are split by '#', then these phrases are used to create the graph. Each node of a graph corresponds to a phrase of the doc and an edge between two node is the cosine similarity between two vectors of the corresponding phrase. The vector of each word is retrieved using FastText. The average vector of every word of that phrase is used as a vector of that phrase.

- **Cosine Similarity:** Similarity measure in data mining refers to the distance between dimensions in a dataset that indicate the features of the data object. There will be a high degree of similarity if this distance is small, but a low degree of similarity if the distance is large.
Cosine similarity is a data that may be used to assess how similar data objects are, despite their size. Python's Cosine resemblance function may be used to compare two texts for resemblance. Cosine similarity treats each data object in a dataset as a vector. The following is the formula to get the cosine similarity between two vectors:

$$\cos(x, y) = \frac{x.y}{\|x\| * \|y\|}$$

$x.y$ = product (dot) of the vectors 'x' and 'y'.

$\|x\|$ and $\|y\|$= length of the two vectors 'x' and 'y'.

$\|x\| * \|y\|$ = cross product of the two vectors 'x' and 'y'.

Example-

The 'x' vector has values, x = {4, 5, 2, 9}

The 'y' vector has values, y = {2, 4, 1, 1}

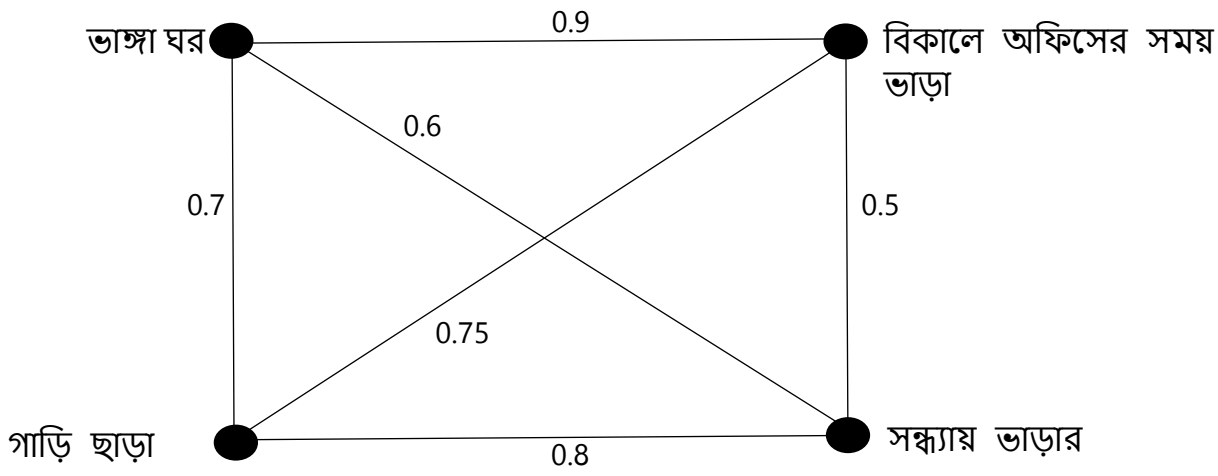
$$x.y = 4 * 2 + 5 * 4 + 2 * 1 + 9 * 1 = 39$$

$$\|x\| = \sqrt{4^2 + 5^2 + 2^2 + 9^2} = 11.22$$

$$\|y\| = \sqrt{2^2 + 4^2 + 1^2 + 1^2} = 4.69$$

$$\cos(x, y) = \frac{39}{(11.22 * 4.69)} = 0.7411$$

- **Missing Vector:** If the word vector of any word for the phrase in our dataset is not present in the word_to_vec_model, then we calculate the average word vector of all the characters of that word for that word.



Graph

D. Ranking Algorithm

- Step-1: Create Adjacency matrix for the graph

	ভাঙ্গা ঘর	বিকালে অফিসের সময় ভাড়া	গাড়ি ছাড়া	সন্ধ্যায় ভাড়ার
ভাঙ্গা ঘর	0	0.9	0.7	0.6
বিকালে অফিসের সময় ভাড়া	0.9	0	0.75	0.5
গাড়ি ছাড়া	0.7	0.75	0	0.8
সন্ধ্যায় ভাড়ার	0.6	0.5	0.8	0

Adjacency Matrix

- Step-2: Calculate Soft Degree of a node

	ভাঙ্গা ঘর	বিকালে অফিসের সময় ভাড়া	গাড়ি ছাড়া	সন্ধ্যায় ভাড়ার	Soft degree of a node
ভাঙ্গা ঘর	0	0.9	0.7	0.6	(0+0.9+0.7+0.6) = 2.2
বিকালে অফিসের সময় ভাড়া	0.9	0	0.75	0.5	(0.9+0+0.75+0.5) = 2.15
গাড়ি ছাড়া	0.7	0.75	0	0.8	(0.7+0.75+0+0.8) = 2.25
সন্ধ্যায় ভাড়ার	0.6	0.5	0.8	0	(0.6+0.5+0.8+0) = 1.9

Soft degree of a node
= Summation of all the row elements of the adjacency matrix corresponds to the node

- Step-3: Normalize the Soft degree of a node

$$\textbf{Normalize Soft degree of a node}(\mathbf{i},\mathbf{j}) = \frac{\boldsymbol{Soft\ degree\ of\ node}(i,j) - \boldsymbol{min}}{\boldsymbol{max} - \boldsymbol{min}}$$

Where,
$i =$ Folder name
$j =$ word of the node
$min =$ Minimum of the soft degree among all the nodes of that doc
$max =$ Maximum of the soft degree among all the nodes of that doc

Example:
In the above graph,
min= 1.9
max= 2.25

$$\textbf{Normalize Soft degree of a node}(\mathbf{i}, \text{ভাঙ্গা ঘর}) = \frac{\boldsymbol{Soft\ degree\ of\ node}(i, \text{ভাঙ্গা ঘর}) - \mathbf{1.9}}{\mathbf{2.25 - 1.9}}$$
$$= \frac{\mathbf{2.2 - 1.9}}{\mathbf{2.25 - 1.9}}$$
$$= \mathbf{0.8571}$$

- Step-4: Calculate Score of a word

$$\textbf{Score of a word} = \textbf{Normalized Soft degree of a node} + \frac{\mathbf{1}}{\sqrt{\boldsymbol{l}}}$$

Where,
$l =$ Sentence no. in which the node phrase occurs first

Example:

Let, the phrase "ভাঙ্গা ঘর" occurs in the 2nd sentence

Then the value of l for word "ভাঙ্গা ঘর" is 2

Therefore,

$$Score\ of\ \text{ভাঙ্গা ঘর} = \text{Normalized Soft degree of ভাঙ্গা ঘর} + \frac{1}{\sqrt{2}}$$

$$= 0.8571 + \frac{1}{\sqrt{2}}$$

$$= 1.5642$$

E. Key-phrase Extraction

The score of each-phrase in a file are provided by the key-phrase extraction method. The top k key-phrases will be chosen, and their compatibility with the topic set will then be determined. We will classify this as a genuine positive if it falls under the topic set, else a false positive. We will manually extract 5 key-phrases from each file, calculate precision, recall and visually represent the key-phrases to make it simple to identify the important key-phrases.

F. Calculate Precision, Recall, F-measure

Precision and recall are performance indicators in information retrieval that relate to data recovered from a collection, corpus, or sample space.

• Precision

In the context of text recognition, precision refers to a system's accuracy in detecting only the pertinent information and rejecting inaccurate or irrelevant information. It evaluates how well the system can produce precise and accurate outcomes.

It measures the proportion of true positive outcomes—that is, important information that was correctly detected—to the total of true positives and false positives, or irrelevant information that was wrongly identified.

A high precision score means that the system correctly identifies relevant information while minimising the inclusion of irrelevant or inaccurate information, and it also means that the system has a low rate of false positives. This is significant in situations like medical diagnosis, legal document analysis, or information extraction from private documents, where accuracy is essential.

The performance of the system cannot be fully understood by looking at precision alone.

It does not consider the possibility of missing relevant information (false negatives), which is measured by recall.

$$Precision_{Top\ K} = \frac{match\ between\ manually\ Selected\ Top\ 5\ keywords\ and\ code\ generated\ Top\ K\ keywords}{K}$$

- **Recall**

Recall is the ability of a system to accurately identify and retrieve pertinent information from a given text in the context of text recognition. It is a gauge of how well the system recognises all the important information contained in the text.

The recall measure is frequently used to assess the effectiveness of such systems, especially in jobs where information extraction and retrieval are essential.

The system's outputs are compared to ground truth data or a reference set in order to determine recall. The system should be able to detect the right or expected information from the reference data.

The ratio of true positive results—the relevant information that was correctly identified—to the total of true positives and false negatives—the important information that the system missed—is used to calculate recall. It gives a sign of how effectively the system gathers the necessary data. High recall shows that the system is effective in locating and retrieving the necessary data. A high recall score does not necessarily imply good precision because it does not take into account the quantity of false positives (i.e., irrelevant material mistakenly classified as important). Contrarily, precision evaluates how well the system is able to recognise just pertinent information while excluding irrelevant data. Recall and precision are frequently combined to assess the overall effectiveness of text recognition systems.

$$Recall_{Top\ K} = \frac{match\ between\ manually\ Selected\ Top\ 5\ keywords\ and\ code\ generated\ Top\ K\ keywords}{Number\ of\ manually\ selected\ keywords(5)}$$

Another metric that can be used to provide a fair assessment is the F-measure, which combines recall and precision.

- **F-measure**

F-measure is a measure of a test's accuracy. It is calculated from the precision and recall of the test. F-measure is the harmonic mean of the precision and recall.

$$\mathrm{F-measure}_{Top\ K} = \frac{2 * Precision_{Top\ K} * Recall_{Top\ K}}{Precision_{Top\ K} + Recall_{Top\ K}}$$

$$\text{Average F-measure}_{Top\ K} = \frac{1}{Number\ of\ Docs} \sum \mathrm{F-measure}_{Top\ K}$$

Example: For a Doc we manually retrieve these key-phrases:
“দেশে oneplus 11 লঞ্চ সম্পর্কে কোন,
বিশ্বের অন্যতম জনপ্রিয় প্রিমিয়াম স্মার্টফোন ব্র্যান্ড oneplus,
পরবর্তী ফ্ল্যাগশিপ লঞ্চ,
লেটেস্ট snapdragon 8 gen 2 চিপসেট,
সংস্থার পরবর্তী ফ্ল্যাগশিপ oneplus 11 লঞ্চ”

Our model retrieves these keyphrases:
1. **দেশে oneplus 11 লঞ্চ সম্পর্কে কোন,**
2. **100 w সুপার ফাস্ট চার্জিং দিচ্ছে চিনা সংস্থাটি 8 gb ram 128 gb স্টোরেজ,**
3. **snapdragon 8 gen 2 চিপসেটoneplus nord n300 20000 টাকার কমে,**
4. **থাকতে চলেছে 67 ইঞ্চি 2k amoled ডিসপ্লেঅ্যামাজনে হাজির,**
5. **অন্যতম গুরুত্বপূর্ণ বাজার হলে,**
6. **ট্রিপল ক্যামেরা প্রাইমারি ক্যামেরায় 50 mp sony imx890 সেন্সর থাকছে,**
7. **ফোনের স্টিরিয়ো স্পিকারে,**
8. **বিশ্বের অন্যতম জনপ্রিয় প্রিমিয়াম স্মার্টফোন ব্র্যান্ড oneplus,**
9. **16 gb ram 256 gb স্টোরেজে,**
10. **লেটেস্ট snapdragon 8 gen 2 চিপসেট,**
11. **ফোনের পিছনে,**
12. **ফোনে থাকতে চলেছে snapdragon 8 gen 2 চিপসেট,**
13. **সংস্থার পরবর্তী ফ্ল্যাগশিপ oneplus 11 লঞ্চ,**
14. **32 mp sony imx709 সেন্সর**

$$Precision_{Top\ 5} = \frac{1}{5} = 0.2$$

$$Recall_{Top\ 5} = \frac{1}{5} = 0.2$$

$$\mathrm{F-measure}_{Top\ 5} = \frac{2 * 0.2 * 0.2}{0.2 + 0.2} = 0.2$$

$$Precision_{Top\ 10} = \frac{2}{10} = 0.2$$

$$Precision_{Top\ 10} = \frac{2}{5} = 0.4$$

$$\text{F} - \text{measure}_{Top\ 10} = \frac{2 * 0.2 * 0.4}{0.2 + 0.4} = 0.27$$

$$Precision_{Top\ 15} = \frac{4}{15} = 0.27$$

$$Precision_{Top\ 15} = \frac{4}{5} = 0.8$$

$$\text{F} - \text{measure}_{Top\ 15} = \frac{2 * 0.27 * 0.8}{0.27 + 0.8} = 0.404$$

For precision and recall, we implemented the method on just 50 documents. So, when we want to calculate the average F-measure, we have to add all the F-measures and then divide it by 50.

Chapter 4
EXPERIMENTS

- **Description of the Dataset**
  We gathered Bengali news from several Bengali news portals in order to carry out the experimental evaluation of the suggested techniques.
- **Dataset 1:**
  We produced 300 datasets with 4 distinct titles of Bengali text documents as given in Table 1. The dataset undergoes pre-processing to prepare it for the keyword extraction technique.

Sl No	Category	No of Documents
1	Astrology	75
2	Dunia	75
3	Sports	75
4	Tech	75

Table 1

- **Dataset 2:**
  We produced another 1756 datasets with 37 distinct titles of Bengali text documents as given in Table 1.
  The dataset undergoes pre-processing to prepare it for the keyword extraction technique.

Sl No	Category	No of Documents	Sl No	Category	No of Documents
1	Agriculture	50	20	Weather	50
2	Business	50	21	World_and_international	50
3	Health	50	22	Cinema	50
4	Labor_and_Employment	50	23	Computer	50
5	Law	50	24	Cricket	50
6	Miscellaneous	50	25	Crime	50
7	Music	50	26	Defence	50
8	Politics	50	27	Economy	50
9	Public_lands_and_water_management	24	28	Education	50
10	Religion	50	29	Banking	50
11	Science	50	30	Election	50
12	Social_welfare	11	31	Electronics	50
13	Space	46	32	Energy	35
14	Sports_other_than_football_and_cricket	50	33	Entertainment	54
15	Caste	42	34	Environment	50
16	Technology	50	35	Family issues	50
17	Transportation	44	36	Finance	50
18	Travel	50	37	Football	50
19	Government_Operations	50			

Table 2

- **Experimental Setup**

  First, we pre-processed data in order to evaluate the performance of the proposed method. After that, using the pre-processed dataset, we produced a co-occurrence graph. Then we create an adjacency matrix for the graph. In order to determine the nodes' scores, we first calculate the soft degree of every node, normalise the soft degrees, and then calculate the score of every node. Finally, we chose the top k-scored nodes to extract the Bengali text's keywords and key-phrases from both techniques. These methods allow us to express the document's context.

Chapter 5
EVALUATION AND RESULTS

The performance is evaluated using the precision, recall & F-measure that represents the correctly classified keywords (or key-phrases) over total keywords (or key-phrases) as represented using the F-measure. Here, the correctly extracted keywords (or key-phrases) by the algorithm are used to calculate the precision, recall & F-measure.

We have chosen the top k extracted keywords (or key-phrases) for assessment, with k equating 5, 10, and 15 keywords (or key-phrases). In this example, the top 5 keywords (or key-phrases) are represented by k = 5, the top 10 keywords (or key-phrases), and the top 15 keywords (or key-phrases). Accordingly, we estimated the Precision, Recall, and F-measure. Table 3 and Table 4 show the Precision, Recall, F-measure and average F-measure findings for each dataset for the top 15 keywords and top 15 key-phrases, respectively. Table 5 and Table 6 show the average F-measure findings for the top 5, top 10, top 15 keywords and top 5, top 10, top 15 key-phrases, respectively.

Top 15				
Doc number	Number of match	pre(top15)	recal(top15)	F-measure(top-15)=2pr/(p+r)
f1	2	0.13	0.40	0.20
f2	4	0.27	0.80	0.40
f3	2	0.13	0.40	0.20
f4	4	0.27	0.80	0.40
f5	3	0.20	0.60	0.30
f6	3	0.20	0.60	0.30
f7	3	0.20	0.60	0.30
f8	3	0.20	0.60	0.30
f9	3	0.20	0.60	0.30
f10	4	0.27	0.80	0.40
f11	4	0.27	0.80	0.40
f12	4	0.27	0.80	0.40
f13	3	0.20	0.60	0.30
f14	2	0.13	0.40	0.20
f15	3	0.20	0.60	0.30
f16	4	0.27	0.80	0.40
f17	4	0.27	0.80	0.40
f18	4	0.27	0.80	0.40
f19	3	0.20	0.60	0.30

f20	3	0.20	0.60	0.30
f21	5	0.33	1.00	0.50
f22	2	0.13	0.40	0.20
f23	2	0.13	0.40	0.20
f24	3	0.20	0.60	0.30
f25	3	0.20	0.60	0.30
f26	4	0.27	0.80	0.40
f27	2	0.13	0.40	0.20
f28	3	0.20	0.60	0.30
f29	3	0.20	0.60	0.30
f30	3	0.20	0.60	0.30
f31	3	0.20	0.60	0.30
f32	3	0.20	0.60	0.30
f33	2	0.13	0.40	0.20
f34	3	0.20	0.60	0.30
f35	4	0.27	0.80	0.40
f36	2	0.13	0.40	0.20
f37	4	0.27	0.80	0.40
f38	2	0.13	0.40	0.20
f39	3	0.20	0.60	0.30
f40	2	0.13	0.40	0.20
f41	4	0.27	0.80	0.40
f42	3	0.20	0.60	0.30
f43	4	0.27	0.80	0.40
f44	4	0.27	0.80	0.40
f45	3	0.20	0.60	0.30
f46	3	0.20	0.60	0.30
f47	3	0.20	0.60	0.30
f48	4	0.27	0.80	0.40
f49	3	0.20	0.60	0.30
f50	3	0.20	0.60	0.30
sum of F-Measure				15.70
Average F-measure				0.31

Table 3

Top 15				
Doc number	Number of match	pre(top15)	recal(top15)	F-measure(top-15)=2pr/(p+r)
f1	4	0.27	0.80	0.40
f2	3	0.20	0.60	0.30
f3	5	0.33	1.00	0.50
f4	4	0.27	0.80	0.40
f5	3	0.20	0.60	0.30

f6	4	0.27	0.80	0.40
f7	4	0.27	0.80	0.40
f8	4	0.27	0.80	0.40
f9	4	0.27	0.80	0.40
f10	5	0.33	1.00	0.50
f11	4	0.27	0.80	0.40
f12	4	0.27	0.80	0.40
f13	4	0.27	0.80	0.40
f14	4	0.27	0.80	0.40
f15	4	0.27	0.80	0.40
f16	5	0.33	1.00	0.50
f17	5	0.33	1.00	0.50
f18	4	0.27	0.80	0.40
f19	2	0.13	0.40	0.20
f20	4	0.27	0.80	0.40
f21	4	0.27	0.80	0.40
f22	3	0.20	0.60	0.30
f23	4	0.27	0.80	0.40
f24	4	0.27	0.80	0.40
f25	2	0.13	0.40	0.20
f26	4	0.27	0.80	0.40
f27	3	0.20	0.60	0.30
f28	3	0.20	0.60	0.30
f29	5	0.33	1.00	0.50
f30	4	0.27	0.80	0.40
f31	4	0.27	0.80	0.40
f32	2	0.13	0.40	0.20
f33	4	0.27	0.80	0.40
f34	4	0.27	0.80	0.40
f35	2	0.13	0.40	0.20
f36	5	0.33	1.00	0.50
f37	3	0.20	0.60	0.30
f38	5	0.33	1.00	0.50
f39	5	0.33	1.00	0.50
f40	3	0.20	0.60	0.30
f41	5	0.33	1.00	0.50
f42	5	0.33	1.00	0.50
f43	3	0.20	0.60	0.30
f44	5	0.33	1.00	0.50
f45	5	0.33	1.00	0.50
f46	4	0.27	0.80	0.40
f47	5	0.33	1.00	0.50
f48	4	0.27	0.80	0.40
f49	4	0.27	0.80	0.40
f50	5	0.33	1.00	0.50

Sum of F-measure			19.80
Average F-measure			0.40

Table 4

	Average F-measure
Top 5	0.252
Top 10	0.3067
Top 15	**0.31**

Table 5

	Average F-measure
Top 5	0.38
Top 10	**0.43067**
Top 15	0.40

Table 6

The maximum average F-measure is obtained by top-15 keywords and top-10 key-phrases

Chapter 5
CONCLUSION AND FUTURE WORKS

Keyword extraction and key-phrase extraction help the reader reduce the time to understand the context of the dataset. In this paper, we have presented the keyword extraction and key-phrase extraction approaches for creating nodes using the co-occurrence graph and then applied a graph-based algorithm to define the score of the nodes. The procedures were applied to four different datasets, and we selected top-k-scored keywords. The keyword extraction procedure and key-phrase extraction procedure performed efficiently in identifying keywords and key-phrases and had a formidable average F-measure, this method can also be applied in micro-blogging datasets.

Graph-based algorithms have produced promising outcomes when it comes to extracting keywords and key-phrases. Here are some probable future study areas for these algorithms that scholars and practitioners might look into:

1. **Improved graph construction:** The quality of the graph representation has a significant impact on the efficiency of graph-based algorithms. By using domain-specific knowledge, contextual information, or external resources like WordNet or Wikipedia, researchers might investigate strategies to improve graph building. This can result in keyword and key-phrase extraction that is more precise and relevant.
2. **Applications with a specified domain:** Future research on adapting graph-based algorithms to particular topics or languages might be interesting. The graph building and algorithm settings may be customised to certain domains in order to improve the extraction of domain-specific keywords and key-phrases because different areas have different features and terminology.
3. **Multilingual keyword/key-phrase extraction:** Another topic that may be investigated is extending graph-based algorithms to handle multilingual text. The algorithms may be changed to extract keywords and key-phrases from a variety of language sources while taking into account the various languages and their unique linguistic characteristics.

References

[1] Matsuo, Y. and Ishizuka, M., 2002. Keyword extraction from a document using word co-occurrence statistical information. *Transactions of the Japanese Society for Artificial Intelligence*, *17*(3), pp.217-223.

[2] Chen, P.I. and Lin, S.J., 2010. Automatic keyword prediction using Google similarity distance. *Expert Systems with Applications*, *37*(3), pp.1928-1938.

[3] Amin, M.R. and Chakraborty, M., 2018, September. Algorithm for bengali keyword extraction. In *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)* (pp. 1-5). IEEE.

[4] Das, B., 2012. Extracting collocations from bengali text corpus. *Procedia Technology*, *4*, pp.325-329.

[5] Abulaish, M., Showrov, M.I.H. and Fazil, M., 2018, November. A layered approach for summarization and context learning from microblogging data. In *Proceedings of the 20th International Conference on Information Integration and Web-based Applications & Services* (pp. 70-78).

[6] Banko, M. and Moore, R.C., 2004. Part-of-speech tagging in context. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics* (pp. 556-561).

[7] Bharti, S.K., Vachha, B., Pradhan, R.K., Babu, K.S. and Jena, S.K., 2016. Sarcastic sentiment detection in tweets streamed in real time: a big data approach. *Digital Communications and Networks*, *2*(3), pp.108-121.

[8] Jonathan, F.C. and Karnalim, O., 2018. Semi-supervised keyphrase extraction on scientific article using fact-based sentiment. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, *16*(4), pp.1771-1778.

[9] Karnalim, O., 2016, November. Software keyphrase extraction with domain-specific features. In *2016 International Conference on Advanced Computing and Applications (ACOMP)* (pp. 43-50). IEEE.

[10] LeCun, Y., Bengio, Y. and Hinton, G., 2015. Deep learning. nature, 521 (7553), 436-444. *Google Scholar Google Scholar Cross Ref Cross Ref*, p.25.

[11] Bengio, Y., Lamblin, P., Popovici, D. and Larochelle, H., 2006. Greedy layer-wise training of deep networks. *Advances in neural information processing systems*, *19*.

[12] Lafferty, J., McCallum, A. and Pereira, F.C., 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

[13] Zhang, C., 2008. Automatic keyword extraction from documents using conditional random fields. *Journal of Computational Information Systems*, *4*(3), pp.1169-1180.