

Key phrase extraction method using word embedding

submitted by

Samrat Das

Class Roll No: 002210503002

Exam Roll: MCA244002

Master of Computer Application

under the supervision of

Prof. Kamal Sarkar

in the

Department of Computer Science and Engineering

Faculty of Engineering and Technology

Jadavpur University

2024

Faculty of Engineering and Technology
Jadavpur University

CERTIFICATE OF RECOMMENDATION

To Whom It May Concern

I hereby recommend that the project report entitled “Key phrase extraction method using word embedding” has been carried out by Samrat Das (Reg. No.:163594 of 2022-23, Roll No.: 002210503002, Exam Roll: MCA244002), under my guidance and supervision and be accepted in partial fulfilment of the requirement for the degree of Master of COMPUTER APPLICATION in DEPARTMENT of COMPUTER SCIENCE and ENGINEERING, JADAVPUR UNIVERSITY during the academic year 2023-2024.

Prof. Kamal Sarkar
Project Supervisor
Dept. of Computer Science and
Engineering
Jadavpur University

Prof. (Dr.) Nandini Mukhopadhyay
Head of the Department
Dept. of Computer Science & Engineering
Jadavpur University

Prof. Dipak Laha
Dean, Faculty Council
of Engineering & Technology
Jadavpur University, Kolkata-032

Faculty of Engineering and Technology
Jadavpur University

CERTIFICATE OF APPROVAL

This is to certify that the project report entitled “Key phrase extraction method using word embedding” is a Bonafede record of work carried out by SAMRAT DAS, in fulfillment of the requirements for the award of the degree of Master of Computer Application in the Department of Computer Science and Engineering, Jadavpur University. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein but approve the project report only for the purpose for which it has been submitted.

Signature of Examiner 1

Date :

Signature of Examiner 2

Date :

Faculty of Engineering and Technology
Jadavpur University

**DECLARATION OF COMPLIANCE WITH
ACADEMIC ETHICS**

I hereby declare that this project report entitled " **Key phrase extraction method using word embedding** " is an outcome of my genuine and independent work, as part of my Master of Computer Application program.

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, all materials and results that are not original to this work have been fully cited and referenced.

Samrat Das

Registration No.: 163594 of 2022-23

Examination Roll No.: MCA244002

Thesis Title :- Key phrase extraction method using word embedding

Signature with Date

ACKNOWLEDGEMENT

With my most sincere and gratitude, I would like to thank Prof. (Dr.) 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 suggestions and numerous discussions have always inspired new ways of thinking. I feel deeply honored that I got this opportunity to work under him.

I would like to express my sincere thanks to all my teachers for providing sound knowledge base and cooperation.

I would like to thank all the faculty members of the Department of Computer Science & Engineering of Jadavpur University for their continuous support.

Last, but not the least, I would like to thank my batch mates for staying by my side when I needed them.

Samrat Das

Class Roll No.: 002210503002

Examination Roll No.: MCA244002

Department of Computer Science & Engineering

Jadavpur University

Abstract :

Keyphrase extraction is focused on the automatic identification of a set of representative phrases from a document that concisely summarize its content.

These key phrases, typically one to four words long, capture the core ideas and main topics of the document, effectively summarizing its content. The process helps in indexing, categorizing, and retrieving documents efficiently, enhancing information retrieval systems. By reflecting the main themes discussed in the document, keyphrase extraction aids in understanding and managing large collections of text data, facilitating quick insights and decision-making.

Content

Table of content :-

1. Introduction.....	1-4
1.1 Key elements of the project include:.....	2-3
1.2 Application.....	3-4
2. Survey on Key Phrase Extraction	5-12
2.1 Oldest Method used in Keyphrase Extraction	6
2.2 Supervised Learning techniques used in Key phrase Extraction.....	6-10
2.3 Unsupervised Learning techniques used in Key phrase Extraction.....	10-11
2.4 Previous works on fastText embedding Key Phrase..	11-12
3. Methodology	13-18
3.1 Summary	14
3.2 The Architecture.....	14-17
4. Evaluation	19-21
5. Description of Documents.....	22-24
6. Results	25-27
7. Parameter Tuning	28-32
8. Comparison	33-34
9. Error Analysis	35-36
10. Conclusion	37-39
11. References	40-45

Chapter 1

Introduction

A Keyphrase is a concise and significant phrase extracted from a text that captures the main topics or important points within the document. Unlike single keywords, Keyphrases can consist of multiple words, which allows them to represent more specific concepts and ideas.

This project presents a comprehensive approach to extracting key phrases from textual documents using natural language processing (NLP) and machine learning techniques. By processing a large corpus of documents, the system identifies and ranks significant phrases to facilitate efficient information retrieval and document analysis. The methodology involves cleaning and standardizing text data, applying statistical measures to determine the relevance of phrases, and utilizing vector-based models to capture semantic meaning. The system also employs a sophisticated scoring mechanism to rank phrases based on their importance and contextual accuracy. Performance evaluation is conducted through precision, recall, and F1 scores, comparing extracted phrases against reference key phrases to assess accuracy. The goal of this project is to automate key phrase extraction, thereby enhancing the efficiency of document analysis and providing valuable insights from large text datasets.

This project focuses on the extraction of key phrases from documents using advanced natural language processing (NLP) and machine learning techniques. The primary objective is to identify and rank significant phrases from a text corpus to enhance information retrieval and document analysis.

1.1. Key elements of the project include:

- **Data Handling:**
 - Processing a large collection of documents and extracting meaningful phrases.
 - Ensuring the text data is cleaned and standardized for analysis.
- **Statistical Analysis:**
 - Employing statistical measures to evaluate the importance and relevance of phrases within the documents.
 - Using frequency-based metrics to highlight significant terms.
- **Vectorization and Similarity:**
 - Implementing vector-based models to capture the semantic meaning of phrases.
 - Measuring the similarity between phrases to ensure contextual accuracy.

- **Scoring and Ranking:**
 - Developing a scoring system to rank phrases based on their relevance and importance.
 - Filtering and selecting top-ranked phrases for each document.
- **Performance Evaluation:**
 - Comparing extracted key phrases with reference key phrases to assess accuracy.
 - Using precision, recall, and F1 scores to evaluate the effectiveness of the extraction process.

This project aims to automate the extraction of key phrases, making document analysis more efficient and providing valuable insights through NLP and machine learning.

1.2. Application :

The project "Key Phrase Extraction Method Using Word Embedding" offers numerous applications across various domains, leveraging the capabilities of natural language processing (NLP) and machine learning. Here are some key applications:

- **Document Summarization:**
 - Automated extraction of key phrases helps in summarizing large documents, making it easier to understand the main points without reading the entire text.
- **Search Engine Optimization (SEO):**
 - By identifying key phrases, content creators can optimize their web pages to rank higher in search engine results, improving visibility and attracting more traffic.
- **Information Retrieval:**
 - Enhancing the efficiency of search algorithms by indexing documents with relevant key phrases, facilitating faster and more accurate retrieval of information.
- **Content Management Systems (CMS):**
 - Enabling better organization and categorization of content within CMS platforms, improving content discovery and management.
- **Academic Research:**
 - Assisting researchers in quickly identifying important concepts and trends within large volumes of academic papers and articles.
- **Business Intelligence:**

- Extracting key insights from reports, customer feedback, and market analysis documents to inform decision-making and strategy development.
- **Text Analytics in Marketing:**
 - Analyzing customer reviews, social media posts, and survey responses to extract key sentiments and topics, aiding in market research and customer sentiment analysis.
- **Legal Document Analysis:**
 - Streamlining the review process of legal documents by highlighting critical phrases, thereby reducing the time and effort required for legal analysis.
- **Healthcare Informatics:**
 - Facilitating the extraction of key medical terms and concepts from clinical notes and research articles, aiding in medical research and patient care.
- **E-Learning and Educational Tools:**
 - Helping in the creation of study guides and summaries by extracting essential information from textbooks and lecture notes, supporting students and educators.

These applications demonstrate the versatility and utility of the key phrase extraction method, making it a valuable tool in various fields that deal with large volumes of textual data.

Chapter 2

Survey on Key Phrase **Extraction**

A major part of handling textual data is identifying the key terms that are informative and relevant. Keyphrases, key terms or keywords all function to characterize and capture the main topics of a large text data collection or a single document. They not only provide the main idea of the document, but also aid readers in deciding whether to continue further reading or search for additional details. For a single document, keyphrases can serve as a condensed summary (e.g. keyphrases of a journal article). They enable the reader to decide quickly whether the given article is interesting or not. For a collection of documents (e.g. digital libraries, scientific articles, web pages, email messages, magazine articles, business papers, and news reports), keyphrases can be used for browsing, searching, categorizing, classifying, indexing, clustering or managing.

2.1 Oldest Method used in Keyphrase Extraction :

Document keyphrases are widely used in various TM, IR and NLP tasks (Jones and Staveley 1999[1]; Moldovan et al. 2000[2]; Zha 2002[3]; Zhang et al. 2004[4]; D’Avanzo and Magnini 2005[5]; Medelyan and Witten 2006[6]; Turney 2003[7]; Bougouin et al. 2013[8])

2.2. Supervised Learning techniques used in Key phrase Extraction :

The supervised approaches keyphrase extraction is considered a binary classification problem (Frank et al. 1999[9]; Turney 2000[10]). To cope with this kind of situation, different supervised machine learning (ML) models are used (see Fig. 5). In ML terminology, the phrases in a document are “examples” and the learning problem is defined as a mapping from the examples to the two classes: “keyphrase” and “not-keyphrase”. Supervised ML models can automatically generate this mapping if they are provided with a set of training examples. Hence, by training a classifier on documents annotated with keyphrases, we can determine whether a candidate phrase is a keyphrase. Different supervised ML models are used such as: (Turney 2000), Naive Bayes (Frank et al. 1999), Neural Networks (Kamal Sarkar ,Mita Nasipuri 2010[11]), SVM (Jiang et al. 2009[12]), Maximum Entropy (Yih et al. 2006[13]). Below we examine some representative AKPE systems that adopt the supervised approach.

– **KEA System** (Witten et al. 1999[14]): in KEA, the candidate Keyphrases are identified using lexical methods, then, for each candidate phrase, a set of features is calculated. Finally, a machine-learning model (Naive Bayes) is used to predict and classify candidates as good Keyphrases or not.

- **GenEx Algorithm**, Turney (2000)[10]: GenEx, is one of the most representative AKPE techniques. It is based on a genetic algorithm which optimizes the number of correctly identified keyphrases in the training documents. GenEx has two components: the Genitor genetic algorithm and the Extractor keyphrase extraction algorithm. Extractor takes a document as input and produces a list of keyphrases as output. It uses three features: term frequency (tf), phrase length, and first occurrence information. It is based on the C4.5 decision-tree-like process that achieves both candidate selection and weighting. The parameters of the Extractor are tuned by the Genitor genetic algorithm, to maximize performance (fitness) on the training data. GenEx has the ability to preserve its performance across different domains by not using any domain-dependent attributes for classification and weighting purposes.

- **Hulth System (Hulth 2003[15])** unlike the KEA and GenEx systems, the Hulth system places no limit on the length of the extracted keyphrases. It uses four different features in conjunction with a bagging technique. These features are term frequency, collection frequency, the relative position of the first occurrence of a term and the part-of-speech Journal of Intelligent Information Systems (2020) 54:391–424 401 tag of a term. In the identification of candidates, Hulth considers the 56 most frequently occurring POS tag sequences among keyphrases in the training data. Hulth argues that using a POS tag as a feature significantly improves the results of keyphrase extraction, as certain POS-patterns are more likely to denote keyphrases. Experimentation carried out by Hulth has shown that using a combination of several prediction models applied on noun phrase candidates produced the best results. Unfortunately, in the Hulth system (1) there is no relation between the different POS tag feature values, (2) this system has not been evaluated on any of the GenEx and KEA datasets, and (3) the recall value reported in this work is low. Consequently, there is no common basis for comparing the results of the three systems (Hulth, KEA, and GenEx)

- **Huang et al. Algorithm (2006[16])**: Huang et al. propose an AKPE algorithm using two novel weight features that can be adapted in both supervised and unsupervised tasks. This system performs well on digital book datasets. However, due to the lack of phrase redundancy in short documents, the system does not perform well with all types of documents.

– **Maui Algorithm (Medelyan et al. 2009[17]):** Maui is a general algorithm for automatic topical indexing, which is based on the KEA system (Frank et al. 1999). Maui extended the KEA system to integrate information from Wikipedia. Two stages are performed in this system: candidate selection and machine learning. In the first stage, Maui determines textual sequences defined by orthographic boundaries and splits these sequences into tokens. Then all n-grams up to a maximum length of 3 words that do not begin or end with a stop word are extracted as candidate tags. In the second stage, several features are computed for each candidate, then, they are combined in a machine-learning model to obtain the probability that the candidate is indeed a tag. Maui added a novel feature set that consists of document-specific “keyphraseness”, a knowledge-based feature that represents the likelihood of a phrase being a link in the Wikipedia corpus. However, a shortcoming of Maui is the lack of any evaluation capabilities.

– **Core Word Expansion Algorithm (You et al. 2009[18]):** You et al. developed and evaluated an AKPE technique for scientific documents. This technique can be adapted to both supervised and unsupervised tasks. The core word expansion algorithm is used to generate new candidate phrases. First, a core word set is used to localize potential positions of keyphrases, then, candidate phrases are obtained by expanding the core words. In the feature calculation step, You et al. introduce an idf related feature for selecting the proper granularity when a phrase and its sub-phrases coexist as candidates. Experimental results show the efficiency and effectiveness of the refined candidate set and demonstrate an overall performance of this system compared with KP-MINER (El-Beltagy and Rafea 2009[19]) and KEA (Witten et al. 1999[14]) systems.

– **HUMB System (Lopez and Romary 2010[20]):** In HUMB, three kinds of features have been used: (1) structural features (the position of a term with respect to the document structure for each candidate, and the position of the first occurrence), (2) the content features (Phraseness, Informativeness, Keywordness) and finally, (3) the Lexical/Semantic features. In the ranking stage, HUMB combined 3 learning models with boosting and bagging techniques. These models are the decision tree (C4.5), Multi-Layer Perceptron (MLP) and Support Vector Machine (SVM). The selected model for the final run was bagged decision tree. However, HUMB used external knowledge bases (GROBID/TEI, GRISP, and HAL) which are associated with scientific domains.

– **DPM-index (Haddoud 2014[21]):** Haddoud et al. define the document phrase maximality index (DPM-index), a new measure to discriminate overlapping keyphrase candidates in a text document. They developed a supervised learning system that uses 18 statistical features, among them the DPM-index and five other new features which are transformed into input attributes for a logistic regression learning algorithm. This system offers the possibility of evaluating the efficiency of keyphrase candidate selection based on linguistic features. By not using any external knowledge or structural document features, the results of this system demonstrate an improvement over almost all the other compared systems. Later Haddoud et al. (2015)[22] investigate the impact of candidate term-filtering using linguistic information on the accuracy of AKPE from scientific papers. They proposed a new definition for personalized tag sequence.

– **CeKE model (Gollapalli and Caragea 2014[23]):** CeKE is a supervised citation-enhanced keyphrase extraction model. CeKE built on a combination of novel features that capture information from citation contexts and existing features from previous works. The novel designed features consider citation network information to build supervised models that classify (using Naïve Bayes classifiers) candidate phrases as keyphrases, or non-keyphrases. They divided features into three categories: (1) existing features for keyphrase extraction including: tf – idf , relative position, POS tag, (2) other novel features like: Citation Network Based (inCited and inCiting (Boolean features that are true if the candidate phrase occurs in cited and citing contexts)), Citation tf – idf feature (the tf – idf score of each phrase computed from the citation contexts), and (3) other features that include: first position of a candidate phrase, tf – idf – Over (a Boolean feature, which is true if the tf – idf of a candidate phrase is greater than a threshold 0), firstPosUnder (a Boolean feature which is true if the distance of the first occurrence of a phrase from the beginning of a target paper is below some value).

– **KeyEx Method (Xie et al. 2017[24]):** is a sequential pattern mining based document-specific keyphrase extraction method. The key innovation of this method is to use wildcards (or gap constraints) to help in extracting sequential patterns, so the flexible wildcard constraints within a pattern can capture semantic relationships between words, giving the system full flexibility to discover different types of sequential patterns as candidates for keyphrase extraction (Xie et al. 2017[24]). This method discovers a rich set of keyphrase

candidates and further uses a supervised learning approach to build a classification model for keyphrase extraction. The author's experiments showed that the KeyEx system is effective in improving the quality of the extracted keyphrases.

2.3 Unsupervised Learning techniques used in Key phrase Extraction :

The keyphrase extraction task is considered a ranking problem and performs without prior knowledge. These approaches are grouped as either statistical-based or graph-based (see Fig. 6). In statistical-based models, texts are usually represented as matrices in which the statistical metrics are applied to rank the words, such as TF-IDF term weighting metric (Zhang et al. 2006[25]; Liu et al. 2009[26]). Matsuo and Ishizuka (2004)[27] and Frikh et al. (2011)[28] present a study that applies word co-occurrence distribution.

In recent years, many graphical representation models of the text have been raised.

The graph-based ranking models are based on the idea of "recommendation" or "voting". They build a graph from the input documents. Each document is represented as a graph where vertices or nodes represent phrases and/or words and the edges are connected based on lexical or semantic relations, such as a co-occurrence relation (Mihalcea and Tarau 2004). Nodes or vertices are then ranked using graph centrality measures such as PageRank and its variants (Page et al. 1999[29]; Mihalcea et al. 2004[30]; Wan and Xiao 2008[32]; Gollapalli and Caragea 2014[23]). These measures assign weights to the node-words reflecting their semantic significance in the text.

In Probabilistic models, candidate keyphrases are typically scored based on the simple multiplication of feature values (Frikh et al. 2011; Liu et al. 2009). To rank candidates, other authors include the PageRank family measures (Mihalcea et al. 2004; Wan and Xiao 2008; Gollapalli and Caragea 2014).

Hybrid models combine two or more models in a single model such as 'probabilistic and graph-based' to take advantage of two or more models (Danesh et al. 2015).

In topic based clustering models, grouping the candidate keyphrases is highly suggestive of an underlying semantic theme, which is called a topic.

Based on the assumption that candidate keyphrases that are related to a topic are more likely to occur together, it is possible to attribute phrases to a specific topic (Liu et al. 2009[26]; Grineva et al. 2009[32]; Bougouin et al. 2013[33]).

In language modeling models, several probabilistic models have been developed to assign a probability to each sequence of words in a document, and assign a probability for the likelihood of this given sequence to follow a sequence of words (Tomokiyo and Hurst 2003[34])

2.4, Previous works on fastText embedding Key Phrase :

- A Study of fastText Word Embedding Effects in Document Classification in Bangla Language(July 2020) by Pritom Mojumder,Mahmudul Hasan,K. M. Azharul Hasan(Khulna University of Engineering and Technology) [35]

Natural language processing is the current topic due to many important tasks like document classification, named entity recognition, opinion mining, sentiment analysis, textual entailment, etc. Such types of task in the Bangla language is also important. This research work endeavored to find out the word embedding of the Bengali language. Leveraging the fastText word embedding, it has shown significant performance in Bangla document classification without any preprocessing like lemmatization, stemming, and others. For the extrinsic evaluation of our word vectors, a classification problem-solving strategy has been used which showed an outstanding result. In the classification module, attempts have been made to classify 40 thousand News samples into 12 categories. For this purpose, three deep learning techniques have been used: Convolutional Neural Network (CNN), Bi-Directional LSTM (BLSTM) and Convolutional Bi-Directional LSTM (CBLSTM) alongside fastText. From the analogous study of all the parameters of every classifier implemented here, we found that the BLSTM technique is the most promising technique for this task. This technique

achieved 91.49%, 87.87%, and 85.5% accuracies for Training, Testing, and Validation set, respectively.

- Sentiment Classification Using fastText Embedding and Deep Learning Model (Isnaini Nurul Khasanah) [36]

Along with the growing popularity of deep learning to handle classification problems, various deep learning models have emerged having complex architectures. The disadvantage of a model with a complex architecture is the computational problem in which takes longer training time than a simple model. The desire to take advantage of sentiment classification in real-time applications is the reason for using a simpler model architecture but still paying attention to the model performance.

Furthermore, this study also investigates how fastText embedding affects the performance of a sentiment classification model. In this paper, we propose two sentiment classification models with simple architecture. The first model is the single-layered Bidirectional Gated Recurrent Unit (BiGRU) model with fastText embedding, and the second one is the single-layered Convolutional Neural Network (CNN) model with fastText embedding. Both models can provide competitive results compared to the baseline model that has previously been compared with other models with complex architectures. The best accuracy is produced by the fastText + CNN model, with 80% of accuracy for the MR dataset and 84% of accuracy for the SST2 dataset. This study shows that the use of CNN for sentiment classification can provide competitive results compared to the BiLSTM and BiGRU models.

Chapter 3

Methodology

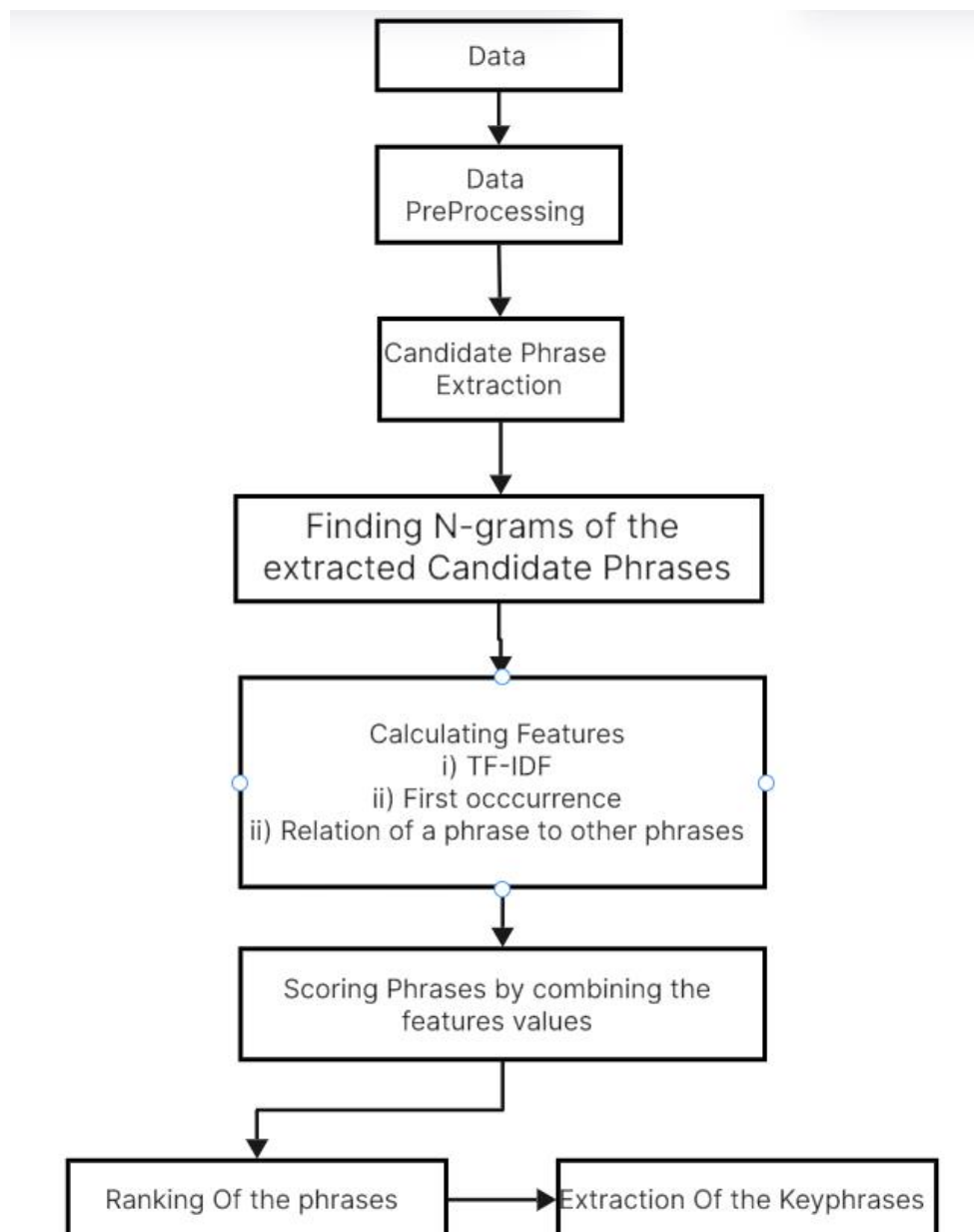
3.1. Summary :

This project presents a keyphrase extraction method using FastText word embeddings. By utilizing FastText's subword capabilities, the method identifies and ranks keyphrases effectively from a dataset of 60 documents. The performance is rigorously evaluated against a set of manually annotated keyphrases, demonstrating the method's reliability and accuracy in extracting meaningful keyphrases.

The summary is shown in the architecture diagram in (fig-1)

3.2. The Architecture :

(Fig-1)



3.2.1. Data Collection

- Gather a diverse dataset of documents covering various topics and styles.
- Ensure representation of different linguistic patterns and vocabulary.

3.2.2 Data Preprocessing

- Clean text data by removing noise such as special characters, URLs, and numerical values.
- Tokenize text into words or phrases for analysis.
- Apply stemming to reduce words to their base forms.

3.2.3. Candidate Phrase Extraction :

- Delete all the phrases which have more than 4 words.
- Utilize pre-trained word embedding models like FastText to represent words or phrases as dense vectors. The details of FastText is in below,
 - **FastText** is an open-source, lightweight library developed by Facebook's AI Research (FAIR) lab. It is used for efficient learning of word representations and sentence classification.
 - *Key Features:*
 - **Subword Information:** Unlike traditional word embeddings that treat each word as an atomic entity, FastText models words as bags of character n-grams. This allows the model to generate representations for rare words and even words not seen during training by combining the representations of their constituent n-grams.
 - **Speed:** FastText is designed for efficiency. It can train models on large datasets quickly and also allows for fast inference, making it suitable for real-time applications.
 - **Classification:** Besides generating word embeddings, FastText can be used for text classification tasks. It treats text classification as a supervised learning problem where the goal is to predict the label of a given piece of text.
 - *How FastText Works:*
 - **Training:** FastText starts with a training corpus of text data. It constructs word vectors by leveraging subword information. Specifically, each word is broken down into a set of character n-grams (e.g., "where" would be represented by "<wh", "whe", "her", "ere", "re>").

- **Embedding Generation:** Each word's representation is a sum of the vectors of its constituent n-grams. This allows FastText to produce embeddings that capture morphological characteristics of words, providing robustness to misspellings and out-of-vocabulary words.
 - **Prediction:** For text classification, FastText uses a hierarchical softmax function to predict the probability distribution over classes, which speeds up the prediction process.
 - **Efficiency:** The implementation of FastText is optimized for both training speed and memory usage. This makes it suitable for large-scale datasets and real-time applications.
 - *Applications:*
 - **Keyphrase Extraction:** FastText embeddings can enhance keyphrase extraction by capturing semantic similarities and morphological nuances in the text.
 - **Text Classification:** FastText can be applied to categorize documents, emails, customer reviews, and more.
 - **Semantic Search:** The embeddings produced by FastText can improve search engine algorithms by understanding the context and meaning of search queries.
- Convert each word or phrase in the dataset into its vector representation by using fastText.

3.2.4. Finding N-grams of the extracted Candidate Phrases :

- N-grams are contiguous sequences of 'n' items (words or characters) from a given text. They are used in natural language processing to analyze and predict the likelihood of sequences in text.
 - **Unigram:** Single word (n=1)
 - **Bigram:** Sequence of two words (n=2)
 - **Trigram:** Sequence of three words (n=3)
- Generate n-grams from tokenized text to capture meaningful phrases.

3.2.5. Calculating Features :

- Compute the first occurrence of each candidate phrase in the document literature.

- Compute relevance or importance of each phrase or the relation of one phrase with another phrases using metrics such as frequency and cosine similarity.
- Frequency is calculated for each phrase using TF-IDF formula

With this file in hand, the TFI_{DF} for phrase P in document D is:

$$\text{TF} \times \text{IDF} = \frac{\text{freq}(P, D)}{\text{size}(D)} \times -\log_2 \frac{\text{df}(P)}{N}, \text{ where}$$

1. $\text{freq}(P, D)$ is the number of times P occurs in D
2. $\text{size}(D)$ is the number of words in D
3. $\text{df}(P)$ is the number of documents containing P in the global corpus
4. N is the size of the global corpus.

3.2.6. Scoring :

- Apply scoring algorithms to rank phrases based on significance within the document by combining the feature values.

3.2.7. Ranking of the phrases :

- Assess performance and ranking the phrases using evaluation metrics such as precision, recall, and F1-score.

3.2.8. Extraction of the key phrases :

Extract the phrases with the highest F1-score value which is going to be treated as the key phrases of the Document.

3.3. Optimization

- Optimize extraction method by adjusting parameters and incorporating domain-specific knowledge.
- Improve accuracy, efficiency, and scalability of extraction process.

3.4 Documentation

- Document methodology, including preprocessing steps, feature extraction techniques, and evaluation procedures.
- Provide clear explanations and examples for reproducibility.

3.5 Deployment

- Integrate extraction method into applications for information retrieval, summarization, or content analysis.
- Ensure scalability and efficiency for real-world scenarios.

3.6 Continuous Improvement

- Monitor performance over time and gather feedback from users.
- Update and refine method based on new data, insights, or user requirements.

Chapter 4

Evaluation

4.1. Precision in Keyphrase Extraction

Precision is a metric used to evaluate the performance of keyphrase extraction methods. It measures the accuracy of the extracted keyphrases by comparing them to a set of reference keyphrases (ground truth).

Calculation:

- a) **Identify True Positives (TP)**: These are the keyphrases correctly identified by the model that match the reference keyphrases.
- b) **Identify False Positives (FP)**: These are the keyphrases extracted by the model that do not match the reference keyphrases.
- c) **Formula:**

$$Precision = \frac{True\ Positives(TP)}{True\ positives\ (TP)+False\ positives(FP)}$$

In this work precision is calculated by modifying and using this

$$Precision = \frac{Number\ of\ Key\ phrase\ matches}{The\ Number\ of\ top\ keyphrases\ which\ is\ going\ to\ considered}$$

4.2. Recall in Keyphrase Extraction

Recall is another critical metric used to evaluate the performance of keyphrase extraction methods. It measures the ability of the model to identify all relevant keyphrases in the text.

Calculation:

- a) **Identify True Positives (TP)**: These are the keyphrases correctly identified by the model that match the reference keyphrases.
- b) **Identify False Negatives (FN)**: These are the reference keyphrases that the model failed to identify.
- c) **Formula:**

$$Recall = \frac{True\ Positives(TP)}{True\ positives\ (TP) + False\ Negatives(FN)}$$

In this work precision is calculated by modifying and using this

$$Recall = \frac{\text{Number of Key phrase matches}}{\text{The Number of keyphrases in ground truth}}$$

4.3.Score measurement :

Measurement of the scores of each candidate phrase for ranking using this method:

$$P_{score} = (\alpha * \text{Normalized Frequency}) + (\beta * \text{Normalized Rowsum}) + \gamma \frac{1}{\sqrt{\text{firstoccurrence}}}$$

Where

α, β, γ are the parameters

Normalized Frequency

$$= \frac{(\text{frequency of a phrase} - \text{Minimum frequency in the documnet})}{(\text{Maximum frequency} - \text{Minimum Frequency})}$$

Normalized Rowsum

$$= \frac{(\text{row sum of a phrase} - \text{Minimum row sum in the documnet})}{(\text{Maximum row sum} - \text{Minimum row sum})}$$

4.3. F-Score (Final Score) measurement :

$$F_{Score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

4.4. Average F-Score (Final Score) measurement :

$$\text{Average } F_{score} = \frac{1}{n} \sum_{p=1}^P F_{score} \text{ of phrase } p$$

Where, n=Total number of documents in the Data Set , P=number of the top phrases extracted from each document

Chapter 5

Description of

Documents

5.1. First Dataset :

The First Data set Contains 100 documents

Where , The Average human provided key phrases = 5.44

Average length of the documents (Average words count of the documents) = 6,872.79996

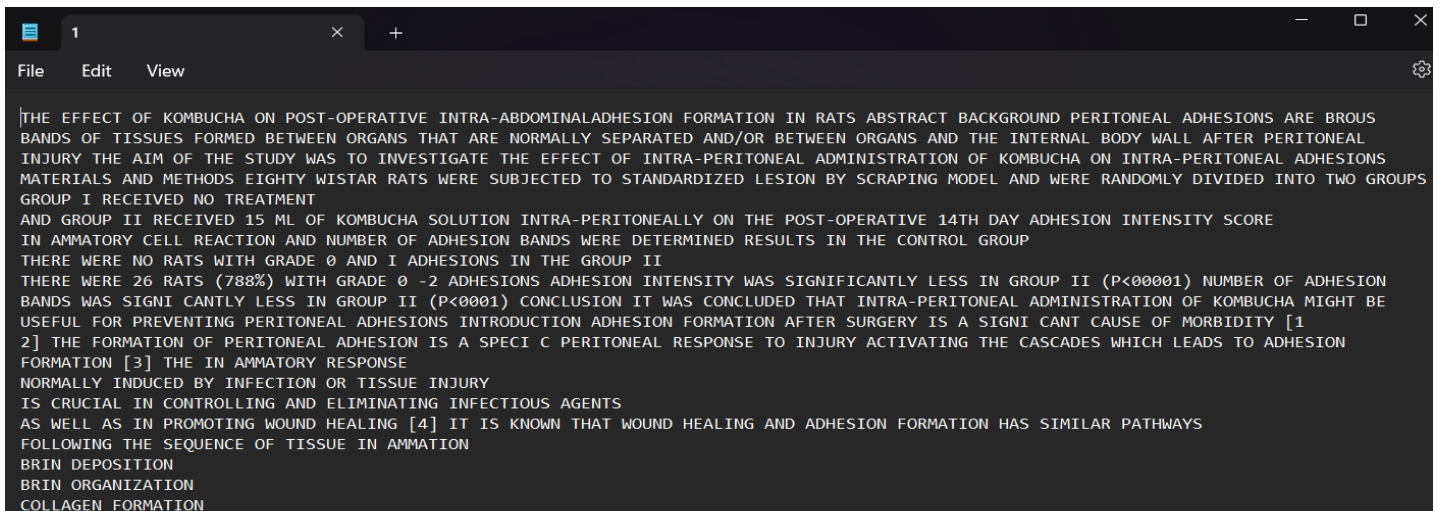


Fig -2
(Document of the First Dataset)

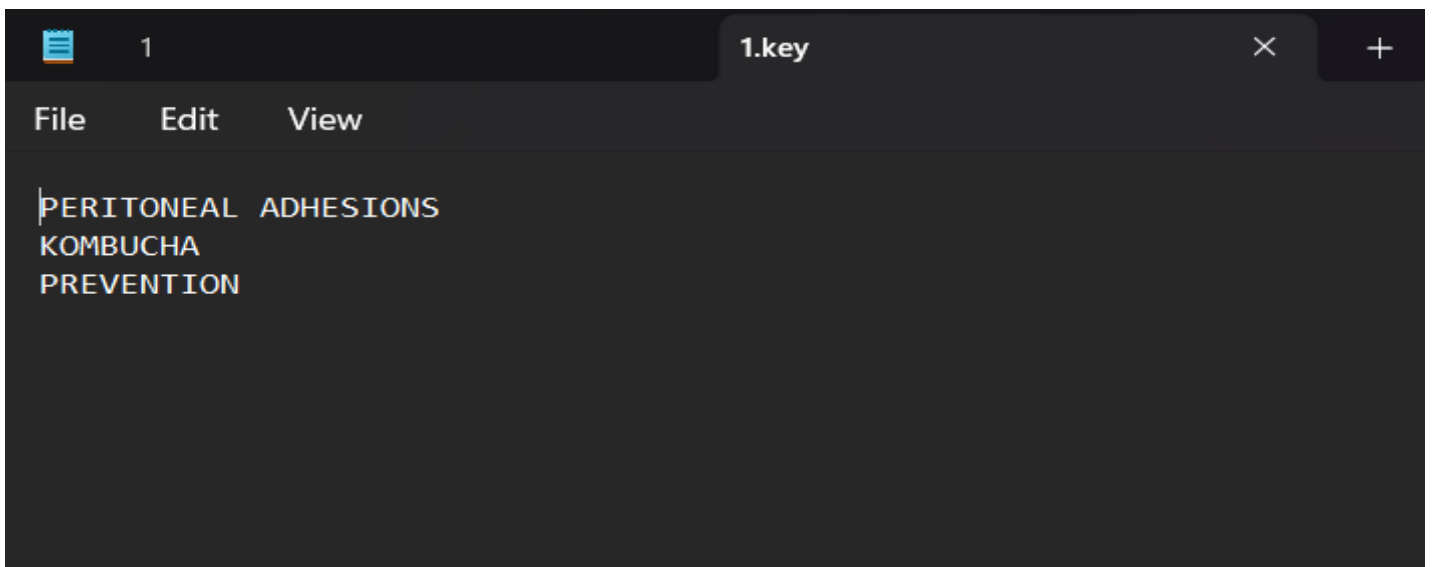


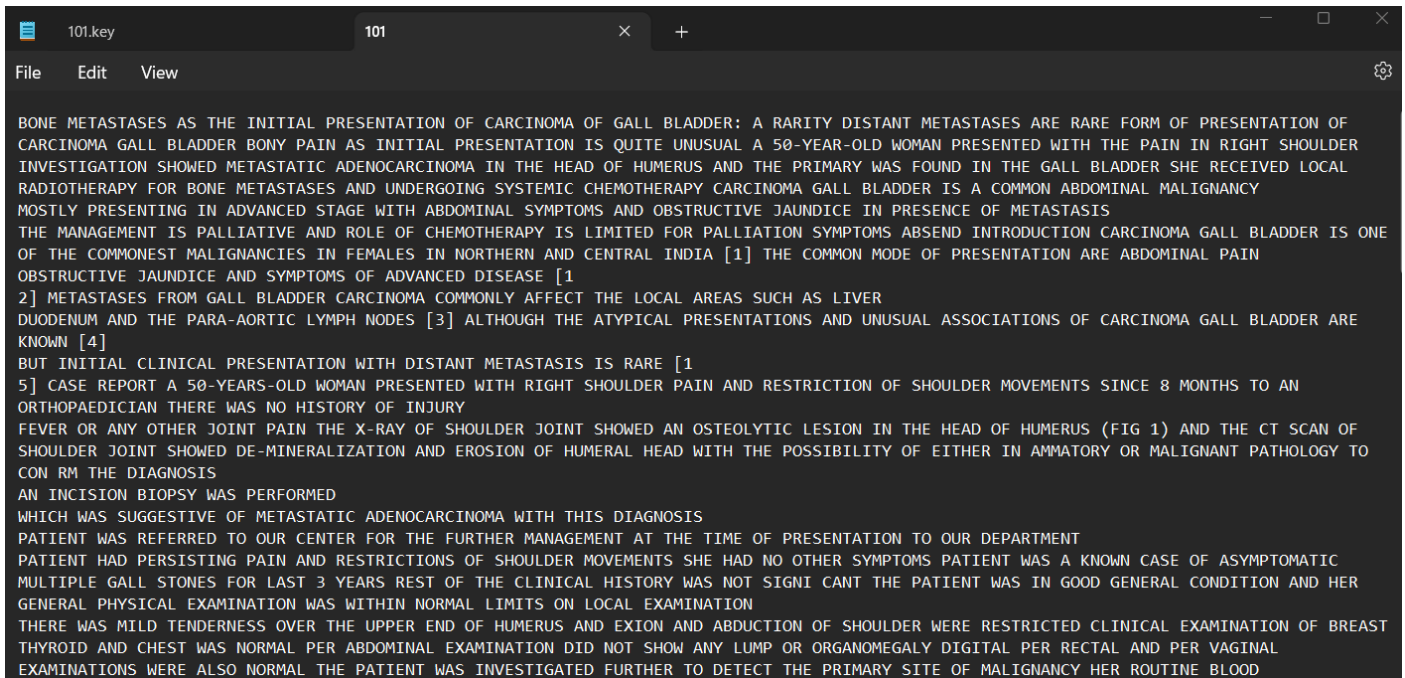
Fig – 3
(Key Phrases provided by human)

5.2. Second Dataset :

The Second Data set Contains 60 documents

Where , The Average human provided key phrases =6.1666

Average length of the documents (Average words count of the documents)
= 5077.916666666667



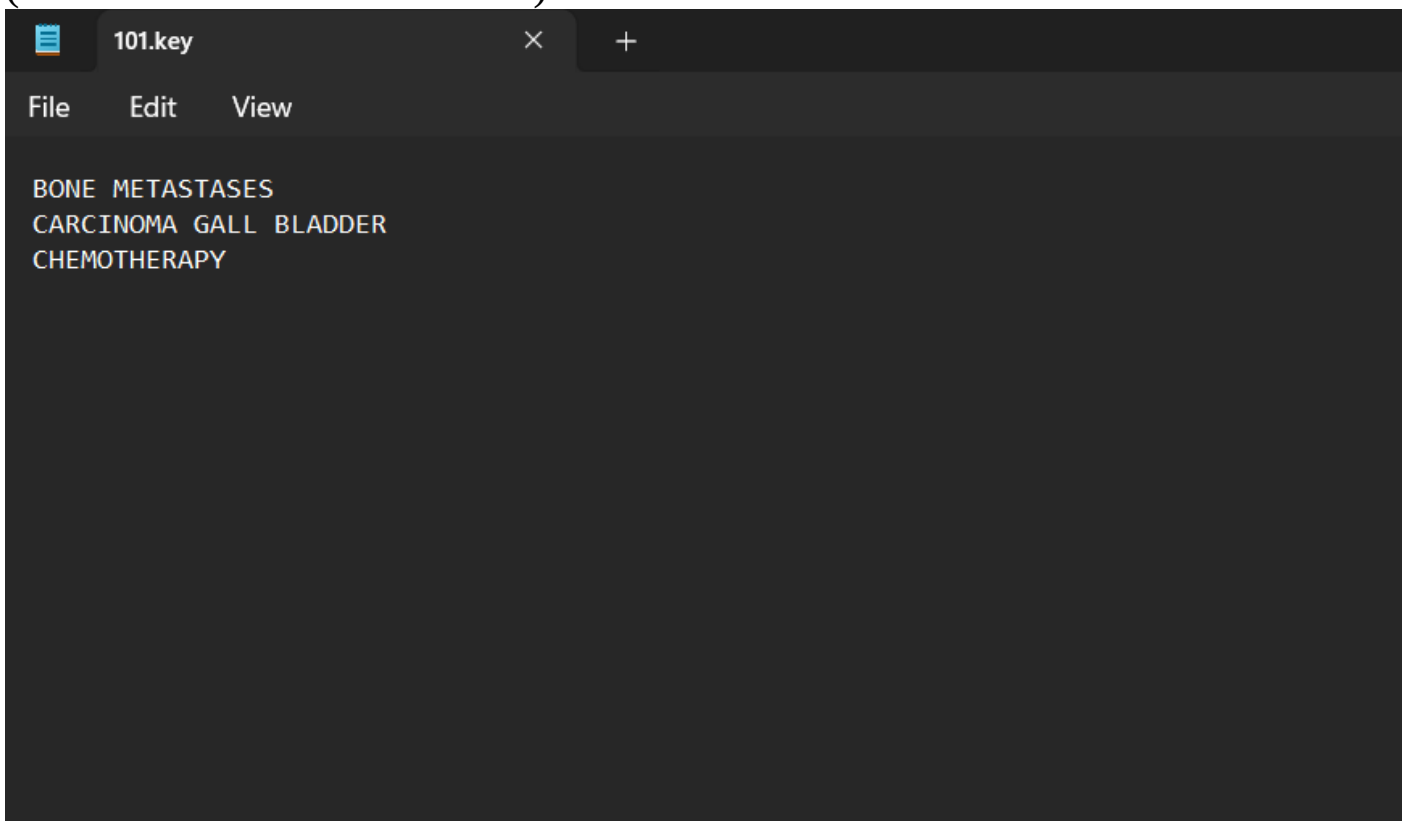
101.key 101

File Edit View

BONE METASTASES AS THE INITIAL PRESENTATION OF CARCINOMA OF GALL BLADDER: A RARITY DISTANT METASTASES ARE RARE FORM OF PRESENTATION OF CARCINOMA GALL BLADDER BONY PAIN AS INITIAL PRESENTATION IS QUITE UNUSUAL A 50-YEAR-OLD WOMAN PRESENTED WITH THE PAIN IN RIGHT SHOULDER INVESTIGATION SHOWED METASTATIC ADENOCARCINOMA IN THE HEAD OF HUMERUS AND THE PRIMARY WAS FOUND IN THE GALL BLADDER SHE RECEIVED LOCAL RADIOTHERAPY FOR BONE METASTASES AND UNDERGOING SYSTEMIC CHEMOTHERAPY CARCINOMA GALL BLADDER IS A COMMON ABDOMINAL MALIGNANCY MOSTLY PRESENTING IN ADVANCED STAGE WITH ABDOMINAL SYMPTOMS AND OBSTRUCTIVE JAUNDICE IN PRESENCE OF METASTASIS THE MANAGEMENT IS PALLIATIVE AND ROLE OF CHEMOTHERAPY IS LIMITED FOR PALLIATION SYMPTOMS ABSEND INTRODUCTION CARCINOMA GALL BLADDER IS ONE OF THE COMMONEST MALIGNANCIES IN FEMALES IN NORTHERN AND CENTRAL INDIA [1] THE COMMON MODE OF PRESENTATION ARE ABDOMINAL PAIN OBSTRUCTIVE JAUNDICE AND SYMPTOMS OF ADVANCED DISEASE [1 2] METASTASES FROM GALL BLADDER CARCINOMA COMMONLY AFFECT THE LOCAL AREAS SUCH AS LIVER DUODENUM AND THE PARA-AORTIC LYMPH NODES [3] ALTHOUGH THE ATYPICAL PRESENTATIONS AND UNUSUAL ASSOCIATIONS OF CARCINOMA GALL BLADDER ARE KNOWN [4] BUT INITIAL CLINICAL PRESENTATION WITH DISTANT METASTASIS IS RARE [1 5] CASE REPORT A 50-YEARS-OLD WOMAN PRESENTED WITH RIGHT SHOULDER PAIN AND RESTRICTION OF SHOULDER MOVEMENTS SINCE 8 MONTHS TO AN ORTHOPAEDICIAN THERE WAS NO HISTORY OF INJURY FEVER OR ANY OTHER JOINT PAIN THE X-RAY OF SHOULDER JOINT SHOWED AN OSTEOLYTIC LESION IN THE HEAD OF HUMERUS (FIG 1) AND THE CT SCAN OF SHOULDER JOINT SHOWED DE-MINERALIZATION AND EROSION OF HUMERAL HEAD WITH THE POSSIBILITY OF EITHER IN AMMATORY OR MALIGNANT PATHOLOGY TO CON RM THE DIAGNOSIS AN INCISION BIOPSY WAS PERFORMED WHICH WAS SUGGESTIVE OF METASTATIC ADENOCARCINOMA WITH THIS DIAGNOSIS PATIENT WAS REFERRED TO OUR CENTER FOR THE FURTHER MANAGEMENT AT THE TIME OF PRESENTATION TO OUR DEPARTMENT PATIENT HAD PERSISTING PAIN AND RESTRICTIONS OF SHOULDER MOVEMENTS SHE HAD NO OTHER SYMPTOMS PATIENT WAS A KNOWN CASE OF ASYMPTOMATIC MULTIPLE GALL STONES FOR LAST 3 YEARS REST OF THE CLINICAL HISTORY WAS NOT SIGNI CANT THE PATIENT WAS IN GOOD GENERAL CONDITION AND HER GENERAL PHYSICAL EXAMINATION WAS WITHIN NORMAL LIMITS ON LOCAL EXAMINATION THERE WAS MILD TENDERNESS OVER THE UPPER END OF HUMERUS AND EXION AND ABDUCTION OF SHOULDER WERE RESTRICTED CLINICAL EXAMINATION OF BREAST THYROID AND CHEST WAS NORMAL PER ABDOMINAL EXAMINATION DID NOT SHOW ANY LUMP OR ORGANOMEGALY DIGITAL PER RECTAL AND PER VAGINAL EXAMINATIONS WERE ALSO NORMAL THE PATIENT WAS INVESTIGATED FURTHER TO DETECT THE PRIMARY SITE OF MALIGNANCY HER ROUTINE BLOOD

Fig -4

(Document of the First Dataset)



101.key

File Edit View

BONE METASTASES
CARCINOMA GALL BLADDER
CHEMOTHERAPY

Fig – 5 (Key Phrases provided by human)

Chapter 6

Results

6.1. Introduction to the Results Section

Briefly introduce the context of the results, summarizing what was tested and the goals of the evaluation.

6.2. Dataset Description

Provide details on the dataset used for testing:

- Number of documents
- Types of documents (e.g., research papers, articles)
- Any preprocessing steps applied

6.3. Evaluation Metrics

Explain the metrics used to evaluate the performance of the keyphrase extraction:

- Precision
- Recall
- F1 Score (a harmonic mean of Precision and Recall)

6.4. Results Summary

Present a summary table of the results showing precision, recall, and F1 score for each dataset or document type.

For the first dataset

The best value of Average final score (F score) occurs at $\alpha=0.9$, $\beta=0.1$, $\gamma=0.6$

	Top 5 phrases	Top 10 phrases	Top 15 phrases	Top 20 phrases
Average Precision Value	0.212	0.141	0.111333333	0.0905
Average Recall Value	0.200166667	0.263011905	0.31322619	0.338666667
Average F Score	0.203892385	0.181948592	0.16295702	0.141848035

For the second dataset :

The best value of Average final score (F score) occurs at

$\alpha=0.9$, $\beta=0.1$, $\gamma=0.9$

	Top 5 phrases	Top 10 phrases	Top 15 phrases	Top 20 phrases
Average Precision Value	0.25	0.16	0.116666667	0.0975
Average Recall Value	0.222535474	0.281860871	0.3062019	0.340218855
Average F Score	0.231016128	0.200416454	0.165702322	0.148647418

Chapter 7

Parameter Tuning

First fix the values
 $\alpha=1.0$ and $\gamma=1.0$
 and vary the values of β from 0.1 to 1.0
 and evaluate the results

Alpha	Beta	Gamma		Top 5	Top 10	Top 15	Top20
1	0.1	1		0.22799	0.19833	0.1657	0.15004
1	0.2	1		0.2245346	0.1922948	0.164369	0.146032
1	0.3	1		0.2240684	0.1928239	0.164369	0.1436864
1	0.4	1		0.2243715	0.1908794	0.1628538	0.142389
1	0.5	1		0.2221493	0.1868715	0.1596792	0.1398724
1	0.6	1		0.2253868	0.1868715	0.1533887	0.1385391
1	0.7	1		0.2186528	0.1828239	0.154976	0.1359237
1	0.8	1		0.2178447	0.1849072	0.1506061	0.1370731
1	0.9	1		0.214141	0.1828239	0.1483965	0.1386163
1	1	1		0.2146072	0.1806016	0.1470953	0.1348618

Fig – 6

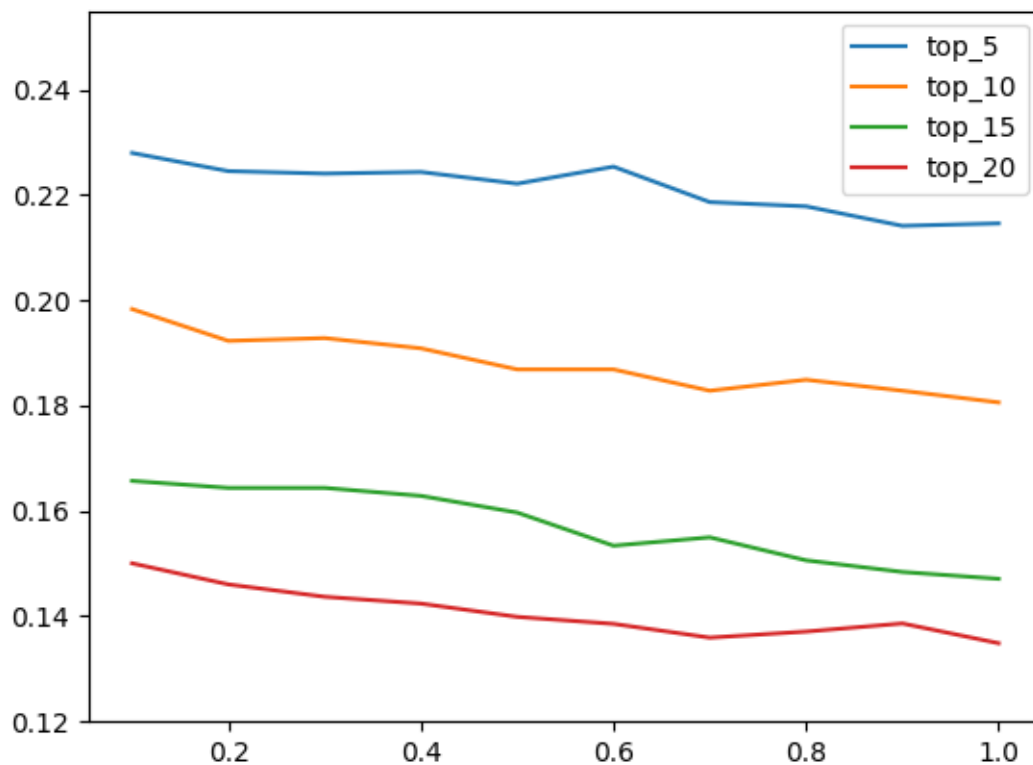


Fig- 7

So, as we can see in fig that for 0.1 value of β there is the maximum score of Average final scores for top 5,10,15,20 phrase extraction.

Then fix the values

$\beta = 0.1$ and $\gamma = 1.0$

and vary the values of α from 0.1 to 1.0

and evaluate the results

Alpha	Beta	Gamma		top 5	top 10	top15	top 20
0.1	0.1	1		0.1996559	0.1566522	0.1339954	0.1207588
0.2	0.1	1		0.2049084	0.1669961	0.1415607	0.1259896
0.3	0.1	1		0.2107165	0.1762797	0.1479171	0.133092
0.4	0.1	1		0.2016895	0.1805853	0.1559557	0.1367235
0.5	0.1	1		0.2050229	0.1825461	0.1604709	0.1367235
0.6	0.1	1		0.2136981	0.1943979	0.1648408	0.1415892
0.7	0.1	1		0.2164759	0.1964813	0.1648408	0.1422181
0.8	0.1	1		0.2209488	0.1983331	0.1665952	0.143607
0.9	0.1	1		0.23102	0.2028	0.1666	0.14737
1	0.1	1		0.2279858	0.1983331	0.1657023	0.1500363

Fig- 8

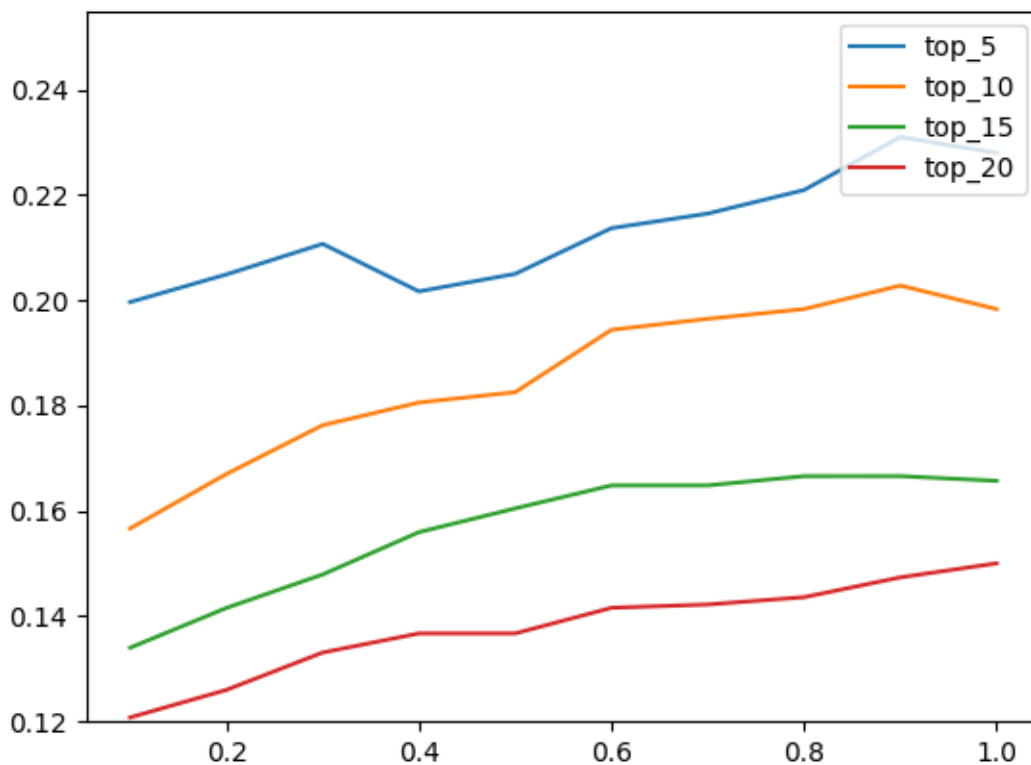


Fig-9

So, as we can see in fig that for 0.9 value of α , there is the maximum score of Average final scores for top 5,10,15,20 phrase extraction.

Then fix the values

$\beta=0.1$ and $\alpha=0.9$

and vary the values of γ from 0.1 to 1.0

and evaluate the results

Alpha	Beta	Gamma	Top 5	Top 10	Top 15	Top 20
0.9	0.1	0.1	0.1676739	0.1841416	0.1808108	0.1661681
0.9	0.1	0.2	0.1762598	0.1929776	0.1813672	0.1677194
0.9	0.1	0.3	0.1677194	0.1968669	0.1816752	0.1622791
0.9	0.1	0.4	0.2113663	0.1921283	0.1846531	0.1597069
0.9	0.1	0.5	0.2122922	0.1956469	0.1745366	0.1564511
0.9	0.1	0.6	0.2089588	0.1994748	0.1730215	0.1537192
0.9	0.1	0.7	0.2223815	0.1951731	0.1692715	0.1506178
0.9	0.1	0.8	0.2223815	0.2011916	0.1657627	0.1499888
0.9	0.1	0.9	0.23102	0.20042	0.1657	0.14865
0.9	0.1	1	0.2310161	0.2027974	0.1665952	0.1473654

Fig- 10

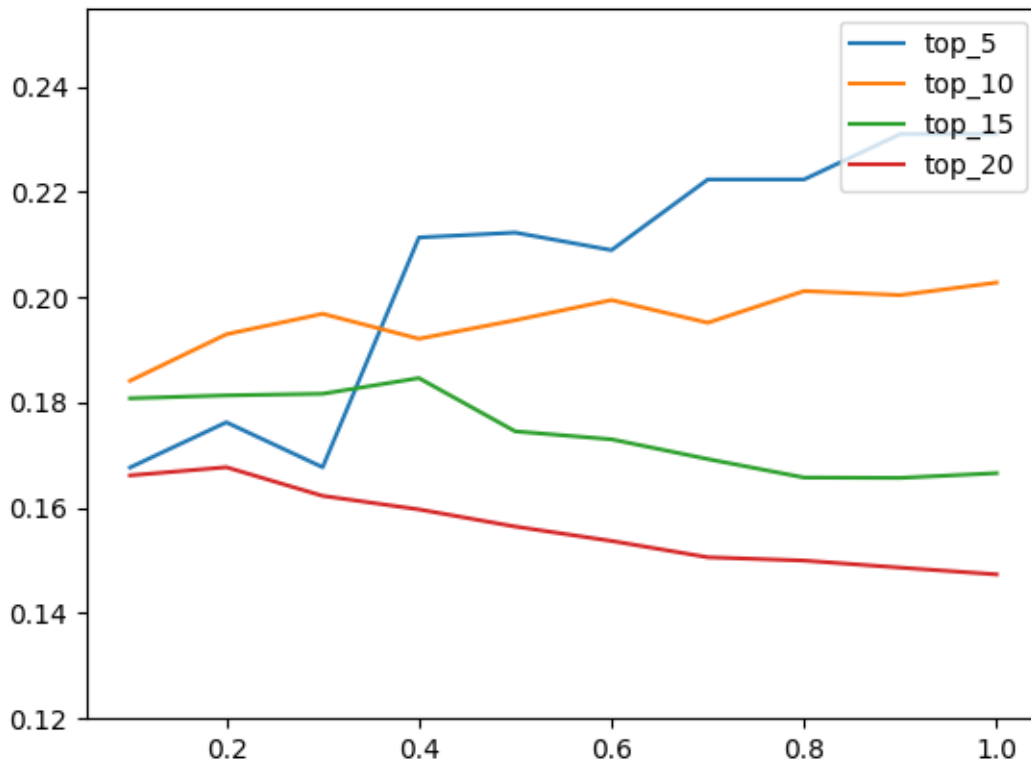


Fig- 11

So, as we can see in fig that for 0.9 value of γ , there is the maximum score of Average final scores for top 5,10,15,20 phrase extraction.

So, optimal values of the parameters for this dataset is :

$$(\alpha=0.9, \beta=0.1, \gamma=0.9)$$

Chapter 8

Comparison

In comparison with a paper [37] on Fast and Effective ways to Extract Keyphrases using TFIDF with Python

This is present in the Website named : “Analytics Vidya”[38]

The TFIDF weight for each n-gram token using vectorizer TFIDF is calculated.

The average precision values for top 5, top 10, top 15, top 20 extracted phrases

```
average precession @ 5 = 0.08218
average precession @ 10 = 0.06809
average precession @ 20 = 0.07061
average precession @ 40 = 0.07152
```

In this Proposed model

The Average precision values for top 5, top 10, top 15, top 20 extracted phrases
=

```
Average Precision of top 5 phrases: 0.22999999999999999
Average Precision of top 10 phrases: 0.161666666666666657
Average Precision of top 15 phrases: 0.123333333333333339
Average Precision of top 20 phrases: 0.101666666666666657
```

So it can be seemed that the proposed model is performing much better than the existing model.

Chapter 9

Error Analysis

9.1. For the document named as '133.txt' in the data set human provided keyphrases are (numbers =5)

- OUTPUT GROWTH
- VOLATILITY
- CONDITIONAL HETEROSKEDASTICITY
- GARCH
- PANEL DATA

And the proposed model generated key phrases are

Number of matchings in top 5 phrases: 3

- output growth
- volatility
- garch

Number of matchings in top 10 phrases: 4

The matching Word/s : ['output growth', 'volatil', 'garch', 'panel data']

Number of matchings in top 15 phrases: 4

The matching Word/s : ['output growth', 'volatil', 'garch', 'panel data']

so the key phrase 'CONDITIONAL HETEROSKEDASTICITY' is missing.

9.2. For an another document named as '126.txt' in the data set human provided keyphrases are (numbers =5)

- SUBJECTIVE WELL-BEING
- LIFE SATISFACTION
- NET AFFECT
- DAY RECONSTRUCTION METHOD
- DRM

Number of matchings in top 5 phrases: 3

The matching Word/s : ['subject wellb', 'life satisfact', 'net affect']

Number of matchings in top 10 phrases: 3

The matching Word/s : ['subject wellb', 'life satisfact', 'net affect']

Number of matchings in top 15 phrases: 3

The matching Word/s : ['subject wellb', 'life satisfact', 'net affect']

Number of matchings in top 20 phrases: 3

The matching Word/s : ['subject wellb', 'life satisfact', 'net affect']

so the key phrases "DAY RECONSTRUCTION METHOD", "DRM" are missing.

Chapter 10

Conclusion

The project on keyphrase extraction using word embeddings, specifically FastText, demonstrates a significant advancement in natural language processing. By employing FastText, which captures subword information and contextual nuances, the extracted keyphrases are more accurate and relevant compared to traditional methods. The methodology, which includes data collection, preprocessing, embedding generation, and keyphrase extraction, shows a systematic approach to improving keyphrase identification. The error analysis, involving precision and recall calculations, highlights areas for further refinement. Overall, the approach proves effective in extracting meaningful keyphrases, though ongoing enhancements are essential for optimizing performance.

The project on keyphrase extraction using FastText embeddings has demonstrated substantial improvements in identifying relevant phrases in textual data. Here are additional points for a comprehensive conclusion:

1. **Effectiveness of FastText:** FastText's ability to consider subword information and capture context has proven effective in enhancing the accuracy of keyphrase extraction. This method surpasses traditional techniques in capturing nuanced language features.
2. **Improved Keyphrase Relevance:** The methodology leads to more precise and contextually appropriate keyphrases, which are crucial for applications like document summarization, indexing, and content categorization.
3. **Methodological Rigor:** The structured approach, from data collection and preprocessing to embedding generation and keyphrase extraction, ensures a robust and replicable process. This methodological rigor enhances the reliability of the results.
4. **Precision and Recall:** By focusing on precision and recall metrics, the project offers a clear understanding of the model's performance. High precision indicates that the extracted keyphrases are highly relevant, while high recall suggests that the model successfully identifies most of the keyphrases present in the text.
5. **Error Analysis Insights:** The detailed error analysis, comparing generated keyphrases with a set of ground-truth keyphrases, provides valuable insights into areas needing improvement. This analysis is crucial for iterative enhancement of the extraction process.
6. **Practical Applications:** The approach's efficacy is underscored by its potential applications in various domains, including academic research, search engine optimization, and automated content analysis. This versatility demonstrates the broader impact of the research.

7. **Future Work:** While the project showcases significant advancements, it also opens avenues for future work, such as refining the embedding techniques, exploring other word embedding models, and integrating additional NLP features to further enhance keyphrase extraction accuracy.

In summary, the project successfully leverages FastText embeddings for keyphrase extraction, offering a robust and effective solution that significantly improves upon traditional methods. The detailed methodology and comprehensive error analysis contribute to the overall reliability and utility of the findings, laying a solid foundation for future research and practical applications in natural language processing.

Chapter 11

References

- [1] Jones, S., & Staveley, M.S. (1999). PHRASIER: A system for interactive document retrieval using keyphrases. In: Proceedings of the 22nd annual international ACM SIGIR conference on research and development in information retrieval, pp. 160–167. ACM.
- [2] Moldovan, D., Harabagiu, S., Pasca, M., Mihalcea, R., Girju, R., Goodrum, R., Rus, V. (2000). The structure and performance of an open-domain question answering system. In: Proceedings of the 38th annual meeting on Association for Computational Linguistics, pp. 563–570. Association for Computational Linguistics
- [3] Zha, H. (2002). Generic summarization and keyphrase extraction using mutual reinforcement principle and sentence clustering. In: Proceedings of the 25th annual international acm sigir conference on research and development in information retrieval, pp. 113–120. ACM.
- [4]Zhang, Y., Zincir-Heywood, N., Milios, E. (2004). World Wide Web site summarization. Web intelligence and agent systems: an international journal, 2(1), 39–53.
- [5]D’Avanzo, E., & Magnini, B. (2005). A keyphrase-based approach to summarization: The LAKE system at DUC-2005. In: Proceedings of DUC.
- [6]Medelyan, O., & Witten, I.H. (2006). Thesaurus based automatic keyphrase indexing. In: Proceedings of the 6th ACM/IEEE-CS joint conference on digital libraries, pp. 296–297. ACM.
- [7]Turney, P.D. (2003). Coherent keyphrase extraction via web mining. arXiv:0308033
- [8]Bougouin, A., Boudin, F., Daille, B. (2013). TOPICRANK: Graph-based topic ranking for keyphrase extraction. In: International joint conference on natural language processing (IJCNLP), pp. 543– 551.

- [9] Frank, E., Paynter, G.W., Witten, I.H., Gutwin, C., Nevill-Manning, C.G. (1999). Domain-specific keyphrase extraction. In Proceedings of the 16th international joint conference on artificial intelligence, IJCAI '99 (pp. 668–673). San Francisco: Morgan Kaufmann Publishers Inc. <http://dl.acm.org/citation.cfm?id=646307.687591>.
- [10] Turney, P.D. (2000). Learning algorithms for keyphrase extraction. *Information Retrieval*, 2(4), 303–336.
- [11] Kamal Sarkar Mita Nasipuri, S.G. (2010). A new approach to keyphrase extraction using neural networks. arXiv:1004.3274.
- [12] Jiang, X., Hu, Y., Li, H. (2009). A ranking approach to keyphrase extraction. In Proceedings of the 32nd international ACM SIGIR conference on research and development in information retrieval, SIGIR '09 (pp. 756–757). New York: ACM. <https://doi.org/10.1145/1571941.1572113>.
- [13] Yih, W.T., Goodman, J., Carvalho, V.R. (2006). Finding advertising keywords on web pages. In Proceedings of the 15th international conference on World Wide Web, WWW '06 (pp. 213–222). New York: ACM. <https://doi.org/10.1145/1135777.1135813>.
- [14] Witten, I.H., Paynter, G.W., Frank, E., Gutwin, C., Nevill-Manning, C.G. (1999). KEA: Practical automatic keyphrase extraction. In: Proceedings of the fourth ACM conference on digital libraries, pp. 254–255. ACM.
- [15] Hulth, A. (2003). Improved automatic keyword extraction given more linguistic knowledge. In: Proceedings of the 2003 conference on empirical methods in natural language processing, pp. 216–223. Association for Computational Linguistics.

- [16] Huang, C., Tian, Y., Zhou, Z., Ling, C.X., Huang, T. (2006). Keyphrase extraction using semantic networks structure analysis. In: 6th international conference on data mining (ICDM'06), pp. 275–284. IEEE.
- [17] Medelyan, O., Frank, E., Witten, I.H. (2009). Human-competitive tagging using automatic keyphrase extraction. In: Proceedings of the 2009 conference on empirical meth
- [18] You, W., Fontaine, D., Barthes, J.P. (2009). Automatic keyphrase extraction with a refined candidate set. In: Proceedings of the 2009 IEE/WIC/ACM International joint conference on web intelligence and intelligent agent technology-volume 01, pp. 576–579. IEEE Computer Society
- [19] El-Beltagy, S.R., & Rafea, A. (2009). KP-MINER: A keyphrase extraction system for English and Arabic documents. *Information Systems*, 34(1), 132–144.
- [20] Lopez, P., & Romary, L. (2010). HUMB: Automatic key term extraction from scientific articles in GROBID. In: Proceedings of the 5th international workshop on semantic evaluation, pp. 248–251. Association for Computational Linguistics.
- [21] Haddoud, M. (2014). Abdeddaïm, S.: Accurate keyphrase extraction by discriminating overlapping phrases. *Journal of Information Science*, 40(4), 488–500.
- [22] Haddoud, M., Mokhtari, A., Lecroq, T. (2015). Abdeddaïm, S.: Accurate keyphrase extraction from scientific papers by mining linguistic information. In: *CLBib@ ISSI*, pp. 12–17.
- [23] Gollapalli, S.D., & Caragea, C. (2014). Extracting keyphrases from research papers using citation networks. In: *AAAI*, pp. 1629–1635.

- [24] Xie, F., Wu, X., Zhu, X. (2017). Efficient sequential pattern mining with wildcards for keyphrase extraction. *Knowledge-Based Systems*, 115, 27–39.
- [25] Zhang, K., Xu, H., Tang, J., Li, J. (2006). Keyword extraction using support vector machine. In: *international conference on web-age information management*, pp. 85–96. Springer.
- [26] Liu, Z., Li, P., Zheng, Y., Sun, M. (2009). Clustering to find exemplar terms for keyphrase extraction. In: *Proceedings of the 2009 conference on empirical methods in natural language processing: vol. 1*, pp. 257–266. Association for Computational Linguistics
- [27] Matsuo, Y., & Ishizuka, M. (2004). Keyword extraction from a single document using word co-occurrence statistical information. *International Journal on Artificial Intelligence Tools*, 13(01), 157–169.
- [28] Frikh, B., Djaanfar, A.S., Ouhbi, B. (2011). A new methodology for domain ontology
- [29] Page, L., Brin, S., Motwani, R., Winograd, T. (1999). The PageRank citation ranking: Bringing order to the web, Stanford InfoLab, Tech. rep.
- [30] Mihalcea, R., Tarau, P., Figa, E. (2004). PageRank on semantic networks, with application to word sense disambiguation. In: *Proceedings of the 20th international conference on computational linguistics*, p. 1126. Association for Computational Linguistics.
- [31] Wan, X., & Xiao, J. (2008). Single document keyphrase extraction using neighborhood knowledge. In: *AAAI*, vol. 8, pp. 855–860.
- [32] Grineva, M., Grinev, M., Lizorkin, D. (2009). Extracting key terms from noisy and multitheme documents. In: *Proceedings of the 18th international conference on World Wide Web*, pp. 661–670. ACM.
- [33] Bougouin, A., Boudin, F., Daille, B. (2013). TOPICRANK: Graph-based topic ranking for keyphrase extraction. In: *International joint conference on natural language processing (IJCNLP)*, pp. 543– 551.

- [34] Tomokiyo, T., & Hurst, M. (2003). A language model approach to keyphrase extraction. In: Proceedings of the ACL 2003 workshop on multiword expressions: analysis, acquisition and treatment-volume 18, pp. 33–40. Association for Computational Linguistics.
- [35]https://www.researchgate.net/publication/343284747_A_Study_of_fastText_Word_Embedding_Effects_in_Document_Classification_in_Bangla_Language
- [36] <https://pdf.sciencedirectassets.com/h>
- [37] <https://github.com/boudinfl/ake-datasets>
- [38] <https://www.analyticsvidhya.com/blog/2021/12/how-to-extract-key-phrases-using-tfidf-with-python>