Online-Journals.org

iJET International Journal of Emerging Technologies in Learning

iJET | eISSN: 1863-0383 | Vol. 19 No. 1 (2024) | OPEN ACCESS

https://doi.org/10.3991/ijet.v19i01.46079

PAPER

Analysis of Abstractive and Extractive Summarization Methods

Mahira Kirmani[1](✉), **Gagandeep Kaur**[1], **Mudasir Mohd**[2]

[1]University Institute of Computing, Chandigarh University, Punjab, India

[2]Department of Computer Application South Campus, University of Kashmir, Jammu and Kashmir, India

mahirakirmani68@gmail.com

ABSTRACT

This paper explains the existing approaches employed for (automatic) text summarization. The summarizing method is part of the natural language processing (NLP) field and is applied to the source document to produce a compact version that preserves its aggregate meaning and key concepts. On a broader scale, approaches for text-based summarization are categorized into two groups: abstractive and extractive. In abstractive summarization, the main contents of the input text are paraphrased, possibly using vocabulary that is not present in the source document, while in extractive summarization, the output summary is a subset of the input text and is generated by using the sentence ranking technique. In this paper, the main ideas behind the existing methods used for abstractive and extractive summarization are discussed broadly. A comparative study of these methods is also highlighted.

KEYWORDS

textual summarization, structure-based approach, extractive summary, sentence ranking methods, abstractive summary, semantic-based approach

1 INTRODUCTION

The exponentially increasing digital data that is accessible worldwide makes the utilization of an automatic text summarization tool inevitable, as manual text summarization entails a considerable number of impartial and knowledgeable experts. The sole objective of automatic text summarization is to express all information in the input text in a vivid, concise, and comprehensive manner, enabling users to save effort and time. Initially, automatic text summarization techniques were applied to one input document, called single document text summarization. The enormous amount of redundant data present on the web provoked the use of multi-document text summarization [1], where a set of multiple documents served as an input to the system. The [2] process of automatic summarization can be divided into the following steps: (a) Preprocessing of the original text, (b) Intermediate representation, and (c) Generating an output as a summary. The summarist text summarization system

Kirmani, M., Kaur, G., Mohd, M. (2024). Analysis of Abstractive and Extractive Summarization Methods. *International Journal of Emerging Technologies in Learning (iJET)*, 19(1), pp. 86–96. https://doi.org/10.3991/ijet.v19i01.46079

Article submitted 2023-10-20. Revision uploaded 2023-11-22. Final acceptance 2023-11-24.

introduced in [3] implements three phases: (a) topic identification, (b) interpretation, and (c) generation. Textual summarization tasks are generally divided into two classes: abstractive and extractive [3]. Extractive summaries are formed by concatenating the main sentences or phrases of the source document. It is a difficult effort to identify the key sentences in the input document; sentence scoring or ranking algorithms are used to solve this problem. On the other hand, [4] [5] abstractive summaries are the compressed paraphrased version of the input text and thus are not a mere concatenation of the main sentences or phrases present in the input document.

Summaries may be divided into two categories based on the original content: an indicative summary and an informative summary. An indicative summary refers to the main concepts of the input document, while all of the pertinent information reported in the input document is included in the informative summary. [6] [4]. Table 1 briefly describes the summarization types. This paper explains the different methods used for extractive and abstractive summarization. Section 2 shows the related work; Section 3 defines the different extractive summarization methods; Section 4 presents the different abstractive summarization methods; Section 5 gives a conclusion; and Section 6 contains the references.

2 RELATED WORK

The 1950s saw the start of the automatic text summarization task [10]. It is now over half a century old and is still progressing because of the increased use of digital data. Luhn [10] unfolded the concept of how frequently occurring words can help in determining important sentences. Then Edmundson [6] broadened Luhn's approach by imparting several other features for indicating salient sentences: (a) Frequency or count of the word in the input text; (b) Frequency of the title terms in the sentence of the source document; (c) Position of the sentence; and (d) Count of cue-phrases such as "significantly," and "concluding" [6]. Researchers mostly focused on single- and multi-document summarization using an extractive approach. At that time, Paice was the one who focused on the techniques for language generation. He pinpointed the main problem that sentence extraction algorithms suffered from (that was the unintended inclusion of those sentences that contained references to the sentences absent in the summary), which resulted in inconsistent summaries [11]. This was relatively early research; the part that follows discusses more recent studies in the area of automatic text summarization. Methods such as lexical aggregation used also helped in condensing the input text by replacing two related concepts with another concept; for example, selling and buying are related to each other, so we replaced them with business. For redundancy removal, syntactic aggregation method was used; for example, Sam plays and Lin plays becomes Sam and Lin plays. Summaries generated on the basis of the keywords are called keyword summaries. We must determine the keywords contained in the input document for keyword summarization. [12] describes the methods for keyword identification. Query-focused summarization determines important parts of the input document based on the user-provided query. The similarity between the query and the sentences in the input document is calculated using support vector regression (SVR). Also summarizes multiple documents on the basis of user queries [9]. In order to generate a quality summary, researchers in recent work have focused on employing neural networks and fuzzy logic [13] [14].

Additionally, it was shown that summarizers based on fuzzy logic and neural networks perform better than those based on statistical methods. Neural networks and fuzzy logic were even used for improving and addressing sentence scoring techniques [15]. Recently, [16] [10] used neural networks in order to summarize the news articles. In the training phase, neural network learned how to check important features of sentences. On the basis of these features, input text was filtered by the neural network, and in the end, a summary of the news article was generated. Deep learning approaches for text summarization have also shown considerable results. In [17], a deep auto-encoder is used to generate an extractive query-focused summary for a single document. Ensemble noisy auto-encoder, an extension of deep auto-encoder, creates noisy input by adding random noise to the input representation in order to choose sentences from a cluster of noisy inputs. Experiments were performed on two separate email corpora that were publicly available, and the system was evaluated using Rouge. Lately, [18] has used attentional encoder-decoder recurrent neural networks to frame their abstractive automatic text summarizer. This work also attempted to address serious problems occurring in the basic model by proposing a few novel models. This paper claimed that these models contributed to boosting the system's performance further.

Table 1. Shows the summarization types

Summarization Types	Definition
Extractive	Concatenation of important sentences or phrases of input text. [3]
Abstractive	The compressed paraphrased version of the input text. [7] [8]
Single document	Single document serves as an input to the system.
Multi-document	Multiple documents serve as an input to the system. [1] [9]
Indicative	Points to the main concepts of the input document.
Informative	Includes all the relevant information that is reported in the input document.
Keyword	Consists of set keywords or phrases present in the input text.
Headline	Summarizes input document by a single important sentence.
Generic	Does not make any assumptions regarding domain or genre of the input; determines importance with respect to the contents of the input document
Query Focused	Based on query given by user it determines important sentences from the input document

3 EXTRACTIVE SUMMARIZATION METHODS

To understand the different methods used for extractive summarization, it is preferable to understand the primary stages of the extractive summarization approach. The predominantly extractive approach is divided into three primary stages: 1) intermediate input text representation, 2) calculating sentence ranks or scores, and 3) generating a summary. These stages are interdependent; that is, each stage's output can be used as an input for the next step.

1. Intermediate input text representation: An input text document is considered raw until it is preprocessed and then transformed into a particular format. In order to apply scoring algorithms, raw input needs to be transformed into a scoring algorithm-specific representation. Frequency-based algorithms consider frequent words as keywords; therefore, text segmentation takes place at the word level. The segmented input text is transformed into a table representation, containing

words and their corresponding frequencies. Likewise, sentence length, sentence position, etc., based algorithms segment input text at the sentence level, which represents each sentence as an indicator. Graph-based algorithms may represent the entire input document as a set of interconnected sentences.

2. Calculating sentence ranks or scores: After transforming the input text into a certain intermediate representation, sentence ranking algorithms are applied to it in order to assign scores to the sentences. TF or IDF, sentence position, sentence length, word co-occurrence, lexical similarity, and proper noun are some of the existing sentence scoring algorithms. The sentence scores assigned determine the importance of the sentences; highly scored sentences have greater chances of being selected for the summary.
3. Generating summary: In the last phase, a linear combination of highly ranked sentences forms a summary. The summary size is apparently less than the size of the original text document. In this phase, similarity check algorithms can be employed in order to remove redundancy in the summary.

The subsequent section presents the main sentence ranking methods employed for extractive summarization approach. The various sentence ranking methods are widely categorized as statistical methods and semantical methods [19].

A) Statistical methods

The methods most widely used in the literature for the extractive summarization approach are statistical methods. Statistical methods operate by observing statistics (such as the number of words, probability of a particular word, term frequency (TF)–inverse document frequency (IDF), etc.) of the text document to identify salient sentences. Such methods do not take into consideration the meaning or sense of the words, phrases, or sentences contained in the input text document. The methods are described below.

1. **Word frequency method:** The concept of word frequency is quite old and was unfolded by Luhn. According to this method, the frequency of each word is recorded, and the sentences are sorted in accordance with the noted frequencies. Sentence rank is incremented for every frequent word that appears in the sentence. Thus, sentences containing the most frequent words are said to be salient.
2. **TF-IDF method:** The trouble with the simple frequency method is that prepositions, determiners, and domain-specific words always acquire the highest frequency counts. These words do not play any role in determining the importance of the sentence; instead, they could affect the consistency of the summary. The TF-IDF method eliminates the impact of these words by comparing each word's frequency ($f(w)$) in the input document with its frequency in all the background documents ($bg(w)$).

$$TF_i * IDF_i = f(w) * log(bg/(bg(w))) \quad (1)$$

TF_i is the term frequency, IDF_i is the inverse document frequency (where i indicates the ith word in the input) and bg is the total number of background documents taken.

3. **Sentence length method:** Long sentences sometimes include information that should be included in the summary; hence, sentence length is significant [20] [14]. For the optimal selection of sentences this method constrains shortened and lengthy sentences.

4. **Uppercase method:** This method tries to identify the important words by assigning higher scores to the words containing uppercase alphabets [20] [14]. This method accounts for the importance of acronyms, initials, and proper names.
5. **Sentence position method:** The position of the sentence with respect to the entire input document is used as a criterion in this method to indicate sentence importance [6] [4]. In this paper, the leading sentence of the document is considered important and, therefore, should be the candidate for the final summary.
6. **Cue-phrase method:** This method identifies summary sentences on the basis of cue-phrases (in particular, salient, the best, hardly, the most important, according to the literature, etc.) present in the input sentences.
7. **Proper noun method:** Sentences which contain one or more than one proper nouns are given higher scores.
8. **Numerical data method:** Important information such as bank transactions, amount, balance, event date, time, etc., is always numerical. This method treats those sentences as important ones that incorporate numerical data.
9. **Similarity of title to sentence method:** The method checks the similarity between a sentence and the title of a document. Thus, sentences similar to the title become summary candidates.

Table 2. Presents a comparative study on various extractive text summarization methods

Authors	Year	Input	Methods	Results
Lloret and Palomar	2009	Single Document	Word Frequency	The system performance was improved by 10% over DUC 2002. [28]
Gupta et al.	2011	Single Document	Word Frequency, Cue-Phrase and Sentence Position.	The deficiently connected Sentences were removed from the summary which resulted in more coherent summary. [12]
Kulkarni and Prasad	2010	Single Document	Word Frequency, Cue-Phrase, Numerical Data and Sentence – Title Similarity.	This system performed semantically better than the MS-word Summarizer.
Abuobieda, Salim, Albaham, Osman, and Kumar	2012	Single Document	Numerical Data and Sentence–Title Similarity, Sentence Length, Word Frequency, Sentence Position.	Results in optimal features selection for the summarization process [29]
Satoshi et al.	2001	Single Document	TF-IDF, Sentence Position and Sentence – Title Similarity.	Having compression ratio equal to 10% the system obtained better results compared to lead-based and TF-based systems.
Murdock	2006	Single Document	TF-IDF	Shows that approaches for language modeling that employ Statistical Translation Models are ineffective.
Fattah and Ren	2009	Single Document	Proper Noun, Sentence Position, Sentence Length, Numerical Data and Sentence – Title Similarity.	Promising results were achieved when the system was tested at different compression rates.
Barrera and Verma	2012	Single Document	TextRank, POS Tagging, wordnet	System outperformed baseline and was evaluated on DUC 2002 and set of articles from scientific magazine. [30]

B) Semantical methods

The summarizers that use statistical methods for the extraction of salient information, to some extent, fail to generate coherent summaries as they do not explore the meaning of the input text. Semantical methods such as emotion used in [19] [13] generate rational summaries by understanding the sentiment or emotion of every sentence in the input document.

4 ABSTRACTIVE SUMMARIZATION METHODS

Methods for abstractive text summarization are broadly categorized as structure-based methods and semantic-based methods.

a) **Structure-based methods:** Structure based methods represent input document using structures like trees, templates, cognitive schemas, etc. Important information is then encoded in these structures. The structure-based methods include:

1. **Tree based:** In this method, the input text document is represented as a dependency tree. Important information is identified by applying different algorithms, such as the theme intersection, etc. Finally, for summary generation, language generators are used.

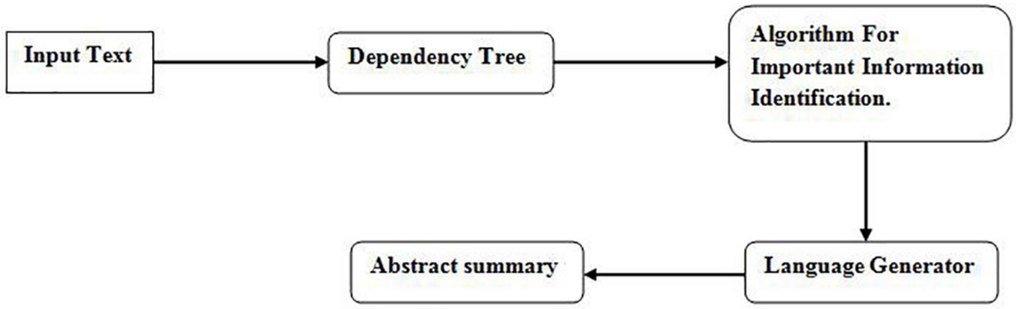

Fig. 1. Block diagram of tree-based method [14] [28]

2. **Template based:** In this method, the input text document is represented as a template. For mapping text snippets into template slots, extraction rules or linguistic patterns are used. Important data is indicated by text snippets.

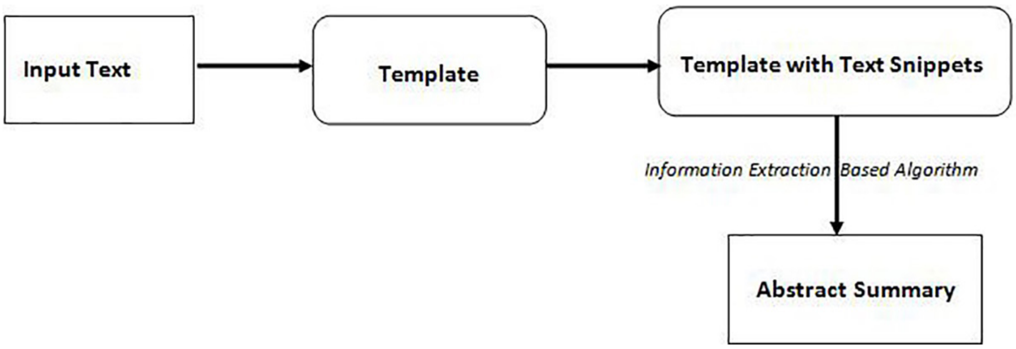

Fig. 2. Block diagram of template-based method [9]

3. **Multimodal-semantic method:** This method takes input in the form of both text and images. This multimodal input document is represented by a

semantic model that clearly apprehends the conception and the relationship among them. Some measures are used to score the important concepts. Finally, concepts chosen for summary are framed as sentences.

4. **Information item-based method:** This technique transforms the information provided in the input text into an abstract representation. From this abstract representation, the contents of the summary are selected.
5. **Semantic graph-based method:** The process creates a rich semantic graph (RSG) from the supplied text document. Then reduction of the rich semantic graph takes place. Finally, this reduced, rich semantic graph acts as the basis for the final abstract summary generation [21].
6. **Ontology-based method:** The domain-related documents can be coherently summarized by ontology-based methods because ontology can better represent a domain as each domain possesses a knowledge structure.
7. **Lead and body phrase:** The technique attempts to rebuild the lead sentence by inserting or substituting phrases that have similar triggers in the body and the lead sentences.

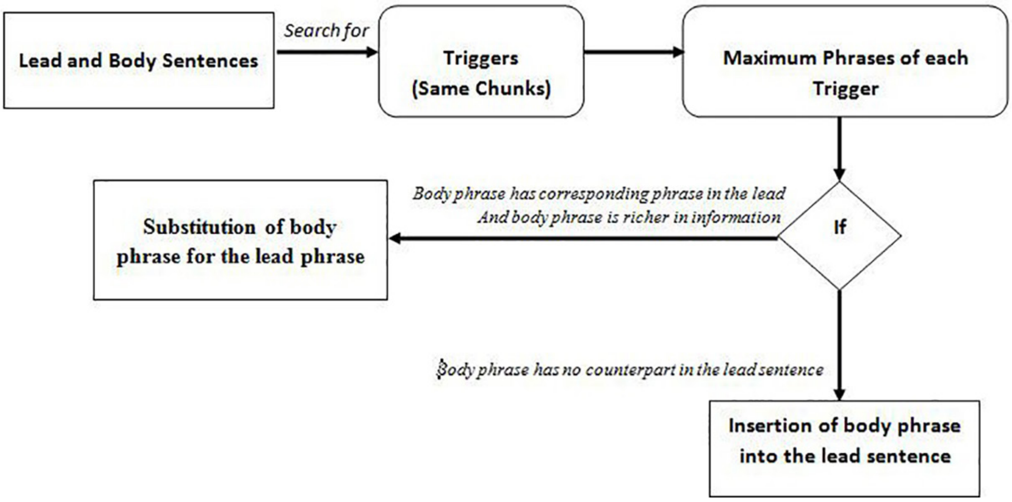

Fig. 3. Block diagram of body and lead phrase method [5]

8. **Rule-based method:** Using this approach, the original a text document is represented as list of aspects and categories. Information extraction rules are used to generate candidates. Then the best candidates are selected by the content selection module. Finally, a summary is generated using generation patterns.
9. **Semantic-based methods:** Semantic-based methods transforms the input document into a semantic representation. This intermediate representation is then supplied to the natural language generation system (NLGS). NLGS processes the linguistic data to identify verb phrases and noun phrases.

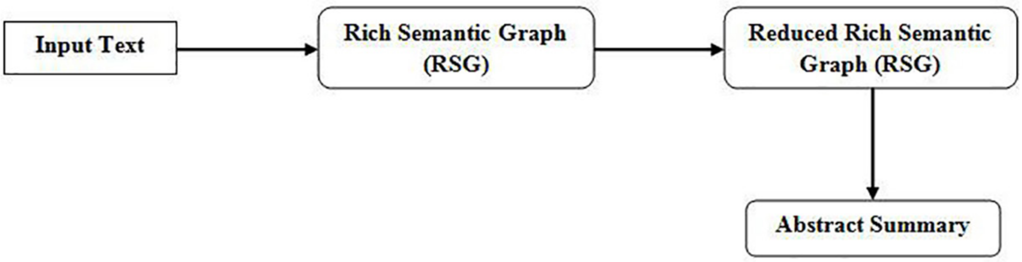

Fig. 4. Block diagram of semantic graph based method [4]

10. **Distributional Semantic techniques:** Distributional semantic models (DSM), often referred to as "distributional similarity" models, are predicated on the idea that it is possible to deduce a word's meaning from its usage—that is, from how frequently it appears in text—at least to a certain extent. By statistically analyzing the situations in which words occur, these models dynamically construct semantic representations in the form of high-dimensional vector spaces. Distributional semantic models make use of the distributional hypothesis, which claims that words deployed in the same context express equivalent meanings. These models are broad and useful for any application because they are trained on huge external datasets. Because they are not domain-specific, these models are adaptable. These characteristics make these models standout selections for the semantics extraction issue.

Some of semantic distributional models are discussed in detail as follows

i. **Word2vec:** A two-layer neural network model called Word2Vec can provide excellent text semantics. A word is converted by the model into a multidimensional vector space embedding. As an output, the model creates a vector from a word, thus the name. The vectors generated are detailed semantic expansions of the original word. Skip-gram and continuous bag of vectors (CBOW) are the two architectures that Word2Vec offers. In contrast to the skip-gram, which forecasts the context from the given word, the CBOW model forecasts the word from its context [21–23].
ii. **Glove:** It is a method of unsupervised learning that creates word-to-vector representations. It establishes the paradigm for converting the frequency of terms that co-occur in the whole of the data. The inference is made using data from collected global word-word co-occurrence statistics [24] [25].
iii. **FasText:** It's an open-source, free program that teaches users how to utilize classifiers and text representations. It is based on the approximation approach, dimension reduction, and n-gram characteristics. The input tokens are converted into n-gram characters. It is a tool for classifying phrases and effectively learning token representations [26].
iv. **BioBERT:** The term stands for bidirectional encoder representations from transformers for biomedical text mining. It is an advanced language representation model for the biomedical sector, trained in advance, using a large biomedical corpus [27].

Table 3. Presents a comparative analysis on various abstractive text summarization methods

Authors	Year	Input	Methods	Results
Barzilay and McKeown	1999	Multiple Documents	Tree based	The system was able to correctly identify 74%, 69%, 74%, and 56% of predicate-argument structures, the subjects, the main verbs, and the other constituents in the list respectively [14].
Barzilay and McKeown	2005	Multiple Documents	Tree based	Summary that is grammatically strict [31].
Harabagiu and Lacatusu	2002	Single and Multiple Documents	Template based	Evaluated GISTEXTER using DUC 2002 and resulted in coherent and organized summary [9].
Lee and Jian	2005	Single Document	Ontology based	Showed that for summarization of news articles news agent operates effectively [20].
Tanaka and Kinoshita	2009	Single Document	Body and Lead phrase	Operations such as insertion and replacements are performed on phrases [5].
Greenbacker	2011	Includes both text and images Multimodal Document	Multimodal Semantic Model	Abstract summaries that incorporate concepts obtained by graphical data [4].

(Continued)

Table 3. Presents a comparative analysis on various abstractive text summarization methods *(Continued)*

Authors	Year	Input	Methods	Results
Genest and Lapalme	2011	Multiple Documents	INIT based	Evaluated system using TAC 2010; average performance was satisfactory [7].
Moawad and Aref	2012	Single Document	Semantic Graph based	Reduced input text document to almost 50% [8].
Genest and Lapalme	2012	Multiple Documents	Rule based	Results in high density information summary [1].
Yash Sharma et al.	2017	Multiple Documents	Word2vec	Test papers with a number between 50 and 284 were used to determine results for the ROUGE 1, ROUGE 2, and ROUGE L tests. The table shows the ROUGE scores' 95% confidence intervals [22].
Enise Karakoc et al.	2019	Single Document	FasText	ROUGE scores were outperformed by semantic similarity scores in terms of performance [23].
Mohd Mudasir et al.	2020	Single Document	Word2vec, Clustering Algorithm, NLTK, NLP	34%, 7%, and 20% were the values of precision respectively [21].
S Kulkarni et al.	2020	Single document	Glove	Glove is used to construct corpora using second-order random walks and calculate graph node embedding [25].

5 CONCLUSION

Manual text summarization entails a considerable number of impartial and knowledgeable experts and a lot of time. However, digital data, which is accessible worldwide is increasing exponentially, making the utilization of an automatic text summarization tool inevitable in order to achieve coherent summaries in less time. Automatic text summarization approaches are widely categorized as extractive approaches and abstractive approaches. This paper presents a review of both extractive as well as abstractive approaches. Different extractive and abstractive methods are explored, and a comparative analysis of the different methods implemented in the literature is presented.

6 REFERENCES

[1] P.-E. Genest and G. Lapalme, "Fully abstractive approach to guided summarization," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2012, pp. 354–358.

[2] R. Aliguliyev, "Automatic document summarization by sentence extraction," vol. 12, no. 5, pp. 5–15, 2007.

[3] E. Hovy and C.-Y. Lin, "Automated text summarization and the SUMMARIST system," in *Proceedings of the TIPSTER Text Program*, Association for Computational Linguistics, Baltimore, Md, USA, 1998, pp. 197–214.

[4] C. Greenbacker, "Towards a framework for abstractive summarization of multimodal documents," in *Proceedings of the ACL 2011 Student Session*, pp. 75–80, 2011.

[5] H. Tanaka, A. Kinoshita, T. Kobayakawa, T. Kumano, and N. Kato, "Syntax-driven sentence revision for broadcast news summarization," in *Proceedings of the 2009 Workshop on Language Generation and Summarisation (UCNLG + Sum 2009)*, 2009, pp. 39–47. https://doi.org/10.3115/1708155.1708163

[6] H. P. Edmundson, "New methods in automatic extracting," *Journal of the ACM (JACM)*, vol. 16, no. 2, pp. 264–285, 1969. https://doi.org/10.1145/321510.321519
[7] P.-E. Genest and G. Lapalme, "Framework for abstractive summarization using text-to-text generation," in *Proceedings of the Workshop on Monolingual Text-to-Text Generation*, 2011, pp. 64–73.
[8] I. F. Moawad and M. Aref, "Semantic graph reduction approach for abstractive text summarization," in *2012 Seventh International Conference on Computer Engineering & Systems (ICCES)*. IEEE, 2012, pp. 132–138. https://doi.org/10.1109/ICCES.2012.6408498
[9] S. M. Harabagiu and F. Lacatusu, "Generating single and multi-document summaries with gistexter," in *Document Understanding Conferences*, 2002, pp. 11–12.
[10] H. P. Luhn, "The automatic creation of literature abstracts," *IBM Journal of Research and Development*, vol. 2, no. 2, pp. 159–165, 1958. https://doi.org/10.1147/rd.22.0159
[11] A. Nenkova, K. McKeown *et al.*, "Automatic summarization," *Foundations and Trends® in Information Retrieval*, vol. 5, no. 2–3, pp. 103–233, 2011. https://doi.org/10.1561/1500000015
[12] V. Gupta and G. S. Lehal, "A survey of text summarization extractive techniques," *Journal of Emerging Technologies in Web Intelligence*, vol. 2, no. 3, pp. 258–268, 2010. https://doi.org/10.4304/jetwi.2.3.258-268
[13] Q. Wu, X. He, Q. V. Nguyen, W. Jia, and M. Huang, "2008 IEEE 8th international conference on computer and information technology," *CIT*, p. 295, 2008.
[14] R. Barzilay, K. McKeown, and M. Elhadad, "Information fusion in the context of multi-document summarization," in *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pp. 550–557, 1999. https://doi.org/10.3115/1034678.1034760
[15] L. Chengcheng, "Automatic text summarization based on rhetorical structure theory," in *2010 International Conference on Computer Application and System Modeling (ICCASM 2010)*, IEEE, 2010, vol. 13, pp. V13-595–V13-598. https://doi.org/10.1109/ICCASM.2010.5622918
[16] M. Yousefi-Azar and L. Hamey, "Text summarization using unsupervised deep learning," *Expert Systems with Applications*, vol. 68, pp. 93–105, 2017. https://doi.org/10.1016/j.eswa.2016.10.017
[17] R. Nallapati, B. Zhou, C. Gulcehre, B. Xiang *et al.*, "Abstractive text summarization using sequence-to-sequence rnns and beyond," *ArXiv Preprint ArXiv:1602.06023*, 2016. https://doi.org/10.18653/v1/K16-1028
[18] I. K. Bhat, M. Mohd, and R. Hashmy, "Sumitup: A hybrid single-document text summarizer," in *Soft Computing: Theories and Applications: Proceedings of SoCTA 2016*, Springer, vol. 1, pp. 619–634, 2018. https://doi.org/10.1007/978-981-10-5687-1_56
[19] J. Kupiec, J. Pedersen, and F. Chen, "A trainable document summarizer," in *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in information retrieval*, pp. 68–73, 1995. https://doi.org/10.1145/215206.215333
[20] C.-S. Lee, Z.-W. Jian, and L.-K. Huang, "A fuzzy ontology and its application to news summarization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 35, no. 5, pp. 859–880, 2005. https://doi.org/10.1109/TSMCB.2005.845032
[21] M. Mohd, R. Jan, and M. Shah, "Text document summarization using word embedding," *Expert Systems with Applications*, vol. 143, p. 112958, 2020. https://doi.org/10.1016/j.eswa.2019.112958
[22] Y. Sharma, G. Agrawal, P. Jain, and T. Kumar, "Vector representation of words for sentiment analysis using glove," in *2017 International Conference on Intelligent Communication and Computational Techniques (ICCT)*. IEEE, pp. 279–284, 2017. https://doi.org/10.1109/INTELCCT.2017.8324059
[23] E. Karakoç and B. Yılmaz, "Deep learning based abstractive turkish news summarization," in *2019 27th Signal Processing and Communications Applications Conference (SIU)*. IEEE, pp. 1–4, 2019. https://doi.org/10.1109/SIU.2019.8806510

[24] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, 2014. https://doi.org/10.3115/v1/D14-1162

[25] S. Kulkarni, J. K. Katariya, and K. Potika, "Glovenor: Glove for node representations with second order random walks," in *2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. IEEE, 2020, pp. 536–543. https://doi.org/10.1109/ASONAM49781.2020.9381347

[26] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," *ArXiv Preprint ArXiv:1607.01759*, 2016. https://doi.org/10.18653/v1/E17-2068

[27] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, "Biobert: A pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, 2020. https://doi.org/10.1093/bioinformatics/btz682

[28] E. Lloret and M. Palomar, "A gradual combination of features for building automatic summarisation systems," in *International Conference on Text, Speech and Dialogue*, 2009, pp. 16–23. https://doi.org/10.1007/978-3-642-04208-9_6

[29] A. Abuobieda *et al.*, "Text summarization features selection method using pseudo Genetic-based model," in *2012 International Conference on Information Retrieval & Knowledge Management*, Kuala Lumpur, Malaysia, 2012, pp. 193–197. https://doi.org/10.1109/InfRKM.2012.6204980

[30] A. Barrera and R. Verma, "Combining syntax and semantics for automatic extractive single-document summarization," in *Computational Linguistics and Intelligent Text Processing*, 2012, pp. 366–377. https://doi.org/10.1007/978-3-642-28601- 8_31

[31] R. Barzilay and K. McKeown, "Sentence fusion for multidocument news summarization," *Computational Linguistics*, vol. 31, no. 3, pp. 297–328, 2005. https://doi.org/10.1162/089120105774321091

7 AUTHORS

Mahira Kirmani is a Research scholar at University Institute of Computing, Chandigarh University, Punjab, India (E-mail: mahirakirmani68@gmail.com; ORCID: 0000-0001-5651-4332).

Gagandeep Kaur is an Assistant Professor at University Institute of Computing, Chandigarh University, Punjab, India (E-mail: gagandeepkaurlogani@gmail.com; ORCID: 0000-0002-1513-8446).

Mudasir Mohd is a Senior Assistant Professor in the Department of Computer Application South Campus, University of Kashmir, J&K, India (E-mail: mudasir.mohammad@kashmiruniversity.ac.in; ORCID: 0000-0003-1597-146X).