International Journal of Engineering Trends and Technology
ISSN: 2231 – 5381 / https://doi.org/10.14445/22315381/IJETT-V70I6P210

Volume 70 Issue 6, 79-89, June 2022

Original Article

Extractive Summarization of Bible Data using Topic Modeling

Vasantha Kumari Garbhapu[1], Prajna Bodapati[2]

[1,2]*Department of Computer Science and Systems Engineering, Andhra University College of Engineering, Visakhapatnam, A.P, India*

[1] vasanthagarbhapu@gmail.com

Received: 23 March 2022 Revised: 14 May 2022 Accepted: 03 June 2022 Published: 27 June 2022

Abstract *- To attain a sense of balance among summary quality and machine readability to preserve the sentence structure and topic similarity, this work presents a statistical and topic modeling-based strategy to extract automatic summarization using the English Bible data set. First, it proposes an algorithm to generate an automatic summary. The measure's core is covered by the Latent Dirichlet Allocation (LDA) method that can capture the most important topics. After that, the summary methods are ranked by the quantity to which the most important topics of their summaries are similar to the most important topics of the reference document. Then, the work focuses primarily on evaluating the summary quality by the ROUGE metric and co-selection measures like Precision, F1 score, and Recall. The evaluation results show that the proposing algorithm has better results with ROUGE score, topic similarity, and manual summary than LSA and TextRank algorithms. Furthermore, this algorithm is competent in computational processing and an understandable method for implementing the English Bible dataset that has not been studied previously.*

Keywords *- Automatic Extract, Latent Dirichlet Allocation (LDA), ROUGE, Summary Evaluation, Text Summarization.*

I. Introduction

Because the exponential increase in information available electronically has developed increasing demand and important to provide better mechanisms to get the information quickly. However, analyzing and understanding the text files is time-consuming, labor-intensive, and tedious [1][2][3]. Text summarization is the process that automatically takes a source text, creates a compressed version of a given text, and presents the most useful information in a condensed form in a way that is sensitive to the task or the user's needs [4][5][6][7[8].

Automatic text summarization deals with selecting which section of the text is the most significant and the problem of generating coherent summaries [9][10][11][12]. There have been two primary methods employed to generate summaries automatically. 1) Extractive and 2) Abstractive. Summarizing extraction minimizes the challenge of summarizing the most important text from the source text. Segments like sentences of the utmost importance extracted from the source text concatenated with each other to form the summary that generates a shorter version of the original text and characterizes it accurately [13][14][15]. In comparison, text summarization based on abstraction may generate sentences unseen in the sources by paraphrasing the extracted content [16]. This abstractive summarization utilizes linguistic methods for a deeper analysis and understanding of the text. Nevertheless, abstractive approaches need deep natural language processing, including semantic representations, natural language generation, and inference, which have yet to reach an established stage. Thus, most researchers prefer to use or investigate more extractive summarization [17][18][19][20][21][22].

Many studies in automatic summarization have also shown that human-quality text summary is incredibly challenging since it involves discourse understanding, language generation, and abstraction. This challenge has been highlighted as a result of many of these efforts. [23]. Research on summarization started with Luhn's extractive paradigm [24]. Later in the mid-90s, the research was motivated partly by statistical approaches to tasks like information extraction (IE) and question answering (QA) [25]. These studies rank document sentences and select sentences with minimum overlap and higher scores [26][27]. This paradigm has been used in the vast majority of the recent research on summarization.

Moreover, evaluating the automatic summaries has been the objective of attention of many researchers over the years. Evaluation chore is considered attractive due to its expenses, informativeness, coherence, and quality [28]. Although the comparison between the system-generated summary and reference summary could be performed manually, it is traditionally made mainly automatic [29]. Human

evaluations are considered very expensive and labor-intensive as they require much human effort [30]. Hence, automatic evaluation measures can be targeted to constitute the proposed dataset. The co-selection measure refers to the F1 score, Precision, and Recall, which are used extensively for defining the proficiency of a system by comparing the system-generated summary to the summary that a human has created. Likewise, the content-based measures, such as unit overlap, ROUGE, cosine similarity, pyramids, etc., [31][32][33] were extensively used. This work focuses primarily on evaluation, considering assessing the summary quality by the ROUGE metric.

2. Methodology

The details of the used methodologies are discussed in detail in the following section and demonstrate how text summarizing can be accomplished. The algorithms automatically utilize several strategies to generate the text corpus summary. The proposed extractive automatic summarization flow chart is shown in Fig.1. However, the summarization system proposed in this research paper is based on topic modeling and statistical methods to increase the information diversity of summaries. It also addresses the topic diversity of the biblical source texts to ensure the summary's quality and reduce the amount of redundancy it contains. This work formulates extractive summarization methods with statistical measures.

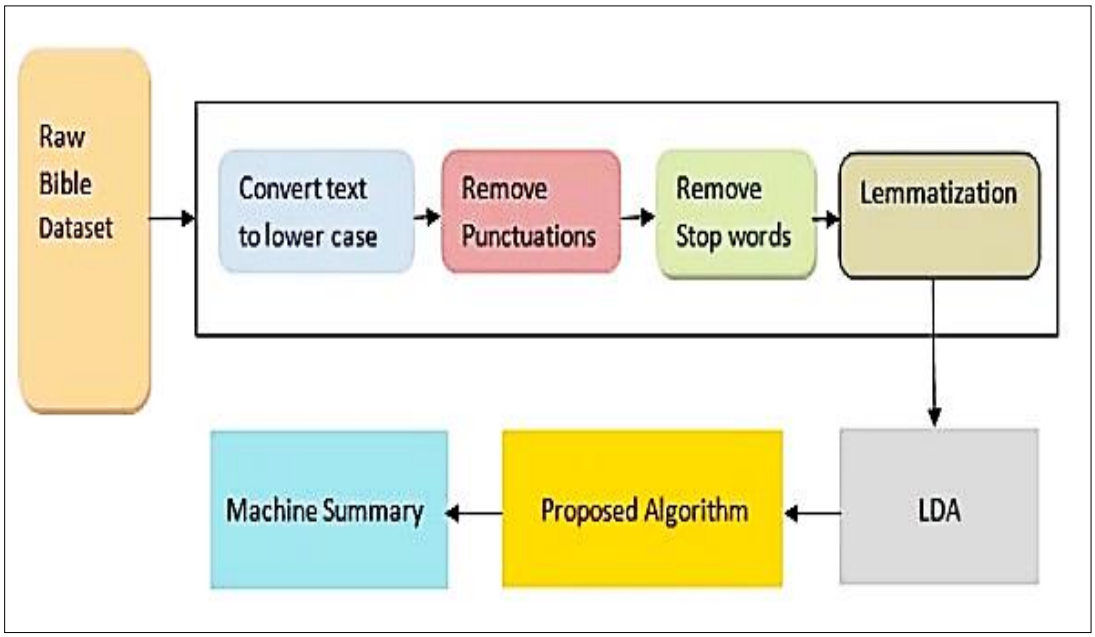

Fig. 1 Flow chart of extractive automatic summarization of Bible text data

2.1. Dataset

The dataset used in this extractive automatic summarization study is from the Holy Bible (NIV- New International Version), which is available online at (http://www.biblegateway.com/passage/?search=Genesis+1 &version=NIV). The text fields in the dataset are different in length, ranging from just a few words to paragraphs containing more than a few sentences each. These text fields describe each incident that was recorded in the dataset. These textual data were mined to discover further information about the events described in the Bible. The book of Genesis is the second biggest book in the Bible and comprises 50 chapters. Each chapter contains 30.66 verses, and the book contains approximately 38262 words.

2.2. Latent Dirichlet Allocation (LDA)

This generative probabilistic model obtains unstructured data as a corpus. Each document is represented as a topic distribution. This method can be represented as follows.

$$P(W, Z, \Theta, \Phi|\alpha, \beta) = \prod_{i=1}^{k} P(\phi_k|\beta) \prod_{j=1}^{M} P(\theta_j|\alpha) \prod_{t=1}^{N_j} P(Z_{j,t}|\theta_j) p(W_{j,t}|\phi_{z_{j,t}})$$

In the above joint distribution, Θ represents topic mixture, Z is the set of topic assignments, W is the words of the corpus, Φ is the topics, and α and β are hyperparameters.

2.3. Latent Semantic Analysis (LSA)

Is used to extract a contextual representation of words using the SVD (Singular Value decomposition) method. SVD divides the TM_{nxm} term_document matrix into three matrices as follows.

$$TM_{nxm} = USV^T$$

Here U is n x r matrix, S is a r x r matrix and V is r x m matrices. Furthermore, rows in V_kS_k are used to correspond to the documents. This new space is used to analyze semantic relatedness between the corpus documents.

2.4. TextRank

It is a graph-based ranking model in which sentences are treated as vertices and based on the relationship between the vertices and draw the edges. Find the score of the vertices V_i and V_j using the following formula

$$S(V_i) = (1-d) + d * \sum_{j \in I_n(V_i)} \frac{1}{(out(V_j)} S(V_j)$$

Here d is a damping factor set between 0 and 1. $I_n(V_i)$ be the set of vertices that points to it. Out (V_j) be the set of vertices that vertex V_j points. Find the scores of all the vertices and sort them based on their final score. To rank or make selections, values are associated with each vertex. [25].

2.5. Compression Ratio

The compression ratio document summary can be evaluated as follows

$$Compression\ ratio = \frac{number\ of\ sentences\ in\ a\ summary}{total\ number\ of\ sentences\ in\ a\ document}$$

2.6. Topic Distribution (θ)

This represents the probability distribution of the topics of the given document. It can be formulated as follows.

$$\theta_j^d = \frac{C_{dj}^{DT} + \alpha}{\sum C_{dj}^{DT} + T\alpha}$$

Here $\mathbf{C_{dj}^{DT}}$ Specifies the number of times a topic j is assigned to words of document d, T is the number of topics, and α is the hyperparameter.

2.7. Topic Diversity

Using this measure, one can determine how closely the summary represents the original document. It can be determined using the cosine similarity between the two topics mixture of summary and original document.

Topic Diversity = cosine similarity (topic mixture (Summary), Topic mixture (Document))

2.8. Redundancy rate

A given summary and document can be measured by finding the number of similar sentences repeated in a document summary. It can be checked while finding the summary as follows. First, add the top-ranked sentence of a document to the summary. Then, while adding the second sentence onwards, check the similarity between the current summary and the sentence to be added. Finally, add that sentence to the summary if the similarity score is less than or equal to the predefined cutoff value. It can reduce the redundancy by doing the same each time by continuing to add new sentences to the summary until it reaches the predefined compression ratio.

2.9. Evaluation metric (*ROUGE*)

Rouge is a Recall-Oriented Understudy for Gisting Evaluation [32] used to determine the quality of automatic summary by counting the number of overlapping 1-gram or n-gram words between automatic summary(machine-generated) and reference summary (human-generated). Generally, it gives the counts as three measures: Precision, Recall, and F1 score.

$$Precision = \frac{number\ of\ overlapping\ words}{total\ words\ in\ automatic\ summary}$$

$$Recall = \frac{number\ of\ overlapping\ words}{Total\ words\ in\ reference\ summary}$$

$$F1\ Score = 2\left(\frac{Precision * Recall}{Precision + Recall}\right)$$

There are different variants of Rouge. They are RougeN and RougeL (Longest Common Subsequences). RougeN evaluated the quality of a summary based on N number of overlapping words. In contrast, RougeL takes the longest common subsequences to assess the summary quality. Therefore, rougeL has two variants; RougeL takes the longest common subsequences at the sentence level. Moreover, RougeLsum considers the longest common subsequences at the summary level.

2.10. Proposed Approach

Input: Bible genesis data (B_d)
Output: Summary (B_s)
Divide the document into chapters
For each chapter B_d
Begin
    Convert all sentences in a chapter to lower case
    Remove noise by removing punctuations,
      stopwords and lemmatization
 end
Fetch cue_words from B_d using LDA.
For each chapter in a B_d
  Begin
    For each sentence in chapter
      begin

score(sentence)=score(NER's)+score(cue_words)+
          similarity_score(text,title)+
Position_score(sentence)
        end
    end
    B_s={ }
    for each chapter in document

```
    begin
      sort the document sentences in score descending order
      sub_summary={ }
      for each sentence in chapter
        begin
          if sub_summary =={ } then
            add sentence to summary
          else
              find similarity(current sub_summary,
                              current_sentence)
              add a sentence to sub_summary if
              similarity score<= pedefined value
        end
      summary= Bs +sub_summary;
    end;
  print BS
```

This approach combines a statistical and topic modeling approach to generate the automatic summary of a source document. The proposed algorithm inputs the noise-free document and highly distributed topic words from the LDA topic modeling algorithm. Firstly, it separates documents into chapters. Each chapter evaluates the score of the sentences by summing up the scores of named entities, the similarity between text and title, the position of the sentence, and cue words. The highly distributed words from the dominated topics of the document from LDA were taken as cue words to improve the similarity of topics between the generated summary and the source document.

The proposed algorithm selects the sentences as follows. Firstly it adds the highest scored sentence to sub summary set. Then, adding another high-scored sentence to the sub summary checks the minimum similarity of that sentence; in this case, it adds to the sub summary and decreases the redundancy. In this manner, it continues adding other subsequent highly scored sentences to the sub-summary until it reaches the specified compression ratio. Finally, it generates the final summary by summing up all sub-summaries.

2.11. Data preprocessing

The data (Book of Genesis) was preprocessed by changing to lower case letters and removing punctuations and stopwords. Further, the data was processed into a vocabulary and morphological analysis of words (Lemmatization). However, even though the Bible data was structured, it is possible to some of the terms contained within it does not help to attain the proposed objectives. Therefore, cleaning up the data by removing the noise using various preprocessing methods is necessary.

2.12. Topic modeling

After preprocessing the bible text, information was concentrated on a topic word search (cue words). The term "hidden topics" in a text document is referred to as "topic modeling" [34]. This approach, termed topic modeling, investigates the words associated with a bible text document to generate a summary of candidate sentences.

2.13. Sentence selection

Following the completion of the topic modeling, the bible text document is converted into a collection of candidate sentences to design an important evaluation function of candidate sentences. Following this step, a desired summary compression ratio is used to choose the candidate sentences with the highest importance scores to obtain an initial machine summary.

3. Results

The proposed approach (Figure 1) was implemented and carried out using Python-based open-source technologies (GENSIM package is used to develop the LDA to fetch the cue words to rank the sentences, and NLTK (Natural Language Toolkit) is used for removing noise and parts_of_speech tagging). To find distance metrics, the SKLEARN library was used. The implementation is performed by utilizing a textual document for automatic summarization, and this document is chosen to carry out the experiments; the dataset accounts for the creation, life on earth, the beginning of sin, the fallen state of the world, the requirement for a redeemer, and the promise of His coming. The Book of Genesis is the first book of the Bible and the Old Testament. These revolve around the covenants that link God to his chosen people and the selected people to the Promised Land.

This work illustrated and summed up the abovementioned methods in the following subsections. The document compares the results with other available algorithms per the generated summary results obtained from the source text. At a specified compression rate, maximum accuracy is obtained. The scalability of the proposed work in the performance parameters, i.e., ROUGE scores, compression ratio, and Topic similarity, are evaluated for three different scenarios. First, the ROUGE scores of the various summarization methodologies provided are listed for each of the four cases in Table 1. The proposed work evaluated four variants of ROUGE, Rouge1, Rouge2, RougeL, and RougeLsum against their corresponding Precision, Recall, and F1 scores at different compression ratios.

Table 1 also shows the correlations achieved by ROUGE scores, and the best results for the F1 score have been bolded for readability. For example, looking at the unigram-based variant (Single word similarity), ROUGE 1, it is observed that the highest F1 score (0.512946) at a 2% compression

ratio, followed by (0.470629) and (0.372326) at 6% and 10% compression ratio with proposed approach respectively. Furthermore, based on the ROUGE2, ROUGEL, and ROUGELsum scores, it can be seen that the proposed system performs better when compared to other systems in terms of bigram and long common subsequence similarity. The proposed algorithm observed the topic similarity (Table 2) at different compression ratios against LSA and TextRank. When the percentage of compression ratio increases, the topic similarity also increases slightly.

The graphical and numerical patterns of the results are presented in Figure 2 to figure 5, respectively. The experimental results are shown in the Figures and tables. The evaluation parameters, Precision, Recall, and F1 score, perform best with the proposed algorithm. As shown in Figure 2, the other algorithms, such as LSA and TextRank, slightly compete with the proposed system. In addition, the recall rate showed high values; however, the overall F1 score is different at 2%, 6%, and 10% compression ratios.

Moreover, the stability of performance of the proposed system is maintained, and even if the summary compression ratios were increased from 6% to 10%, the results showed improved performance (Figures 3 and 4). It can be shown from figure 5 that the proposed LDA-based approach that was used to generate the summary has good topic coherence as the original document at various compression ratios ranging from 2% to 10%.

When seen from a human expert's perspective, the generated summary quality is good when compressed to 2%. However, when the compression ratio is 6 and 10 %, each algorithm's summary quality is more or less similar. Figure 6 illustrates this point perfectly. Furthermore, Table 3 contains a list of synonyms, which, compared to the human summary generated by the machine, do not show many similarities. Although they are different words, the meanings they convey are the same.

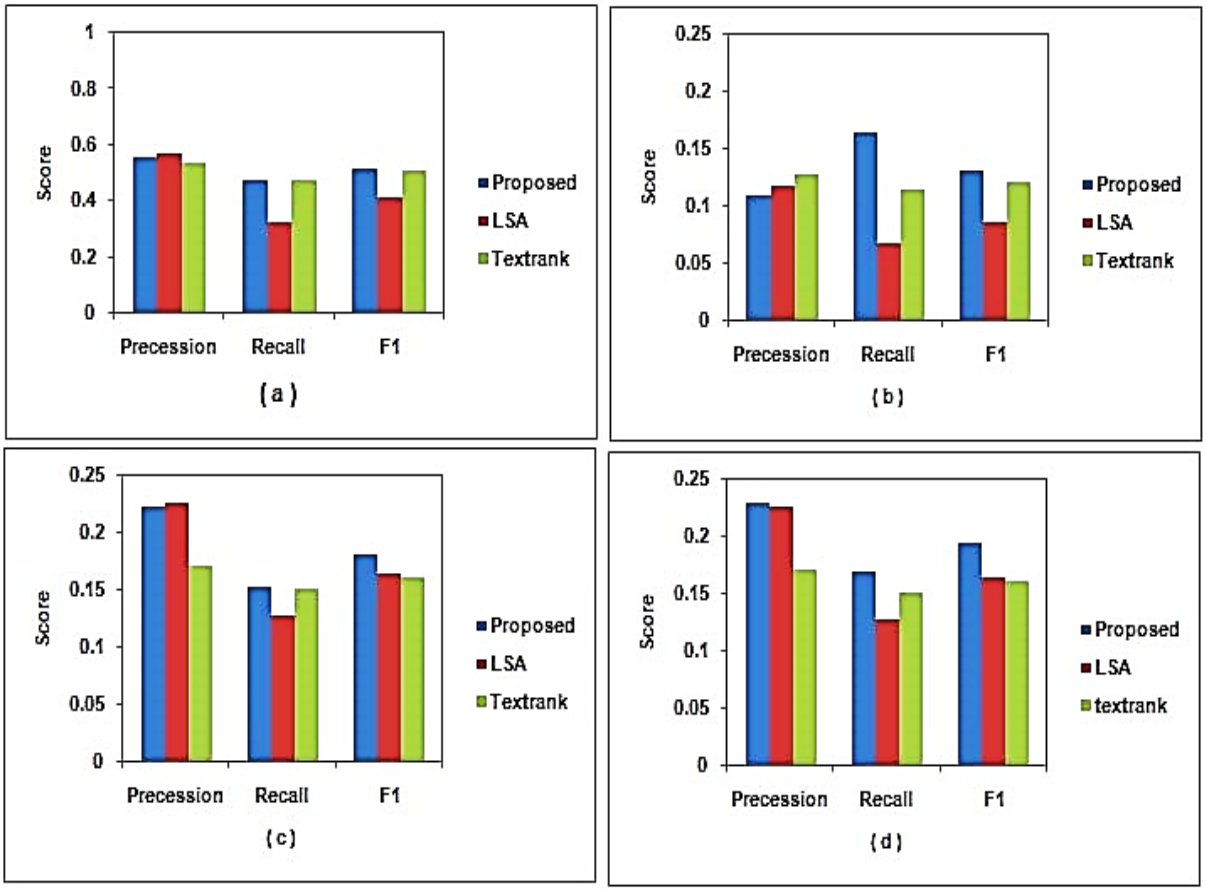

Fig. 2 Evaluation results on summary quality using ROUGE scores (a: ROUGE1, b: ROUGE 2, c: ROUGEL, d: ROUGELsum) at a 2% compression ratio

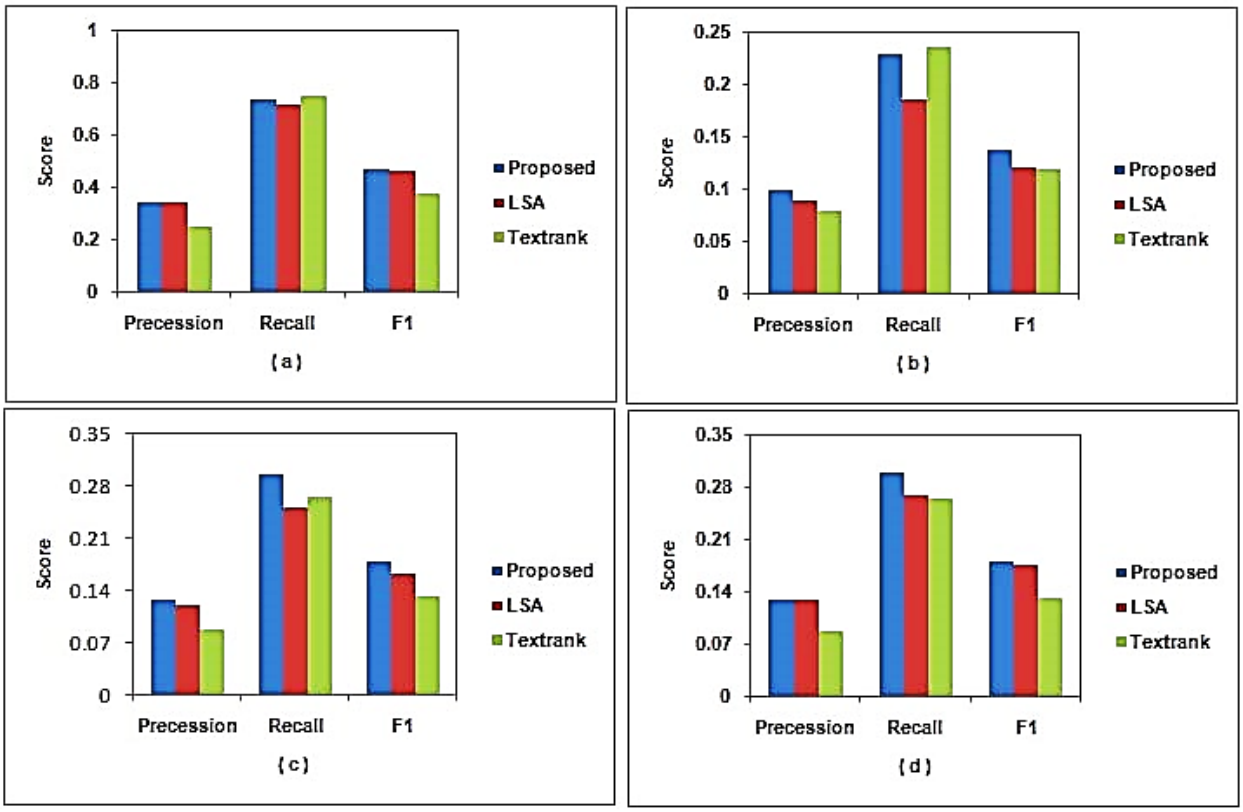

Fig. 3 Evaluation results on summary quality using ROUGE scores (a: ROUGE1, b: ROUGE 2, c: ROUGEL, d: ROUGELsum) at a 6% compression ratio

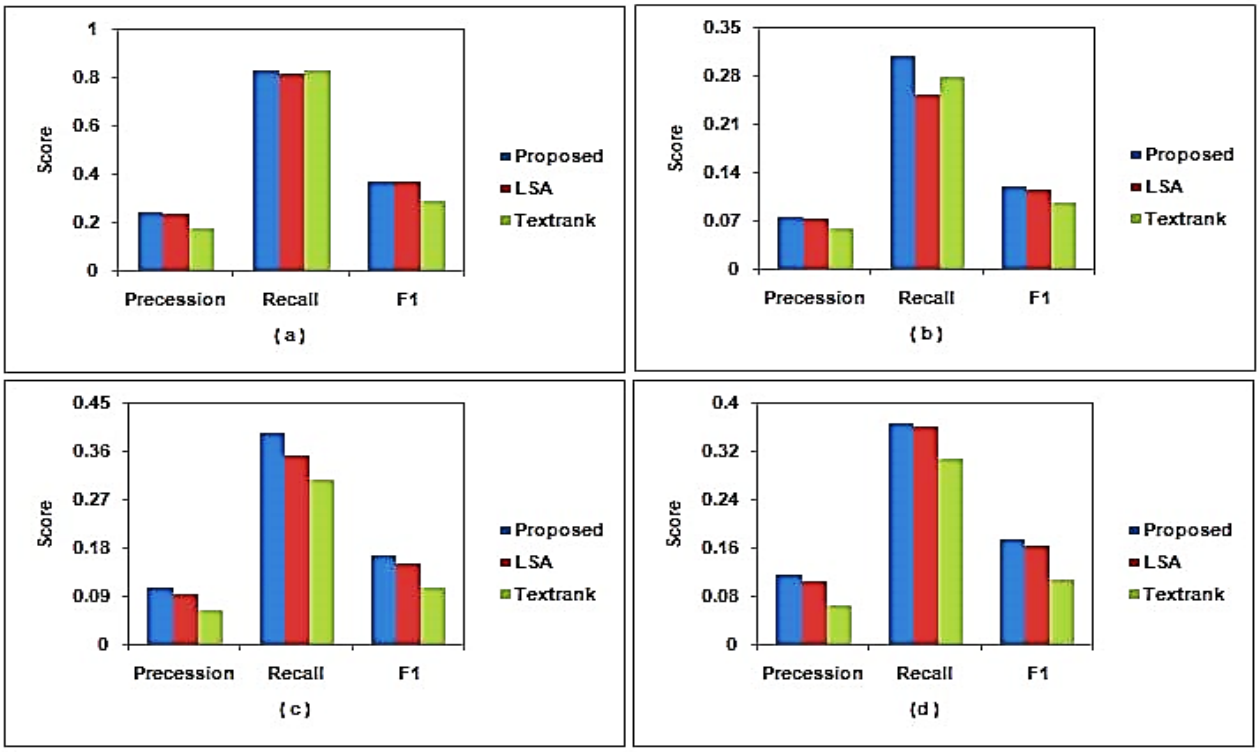

Fig. 4 Evaluation results on summary quality using ROUGE scores (a: ROUGE1, b: ROUGE 2, c: ROUGEL, d: ROUGELsum) at a 10% compression ratio

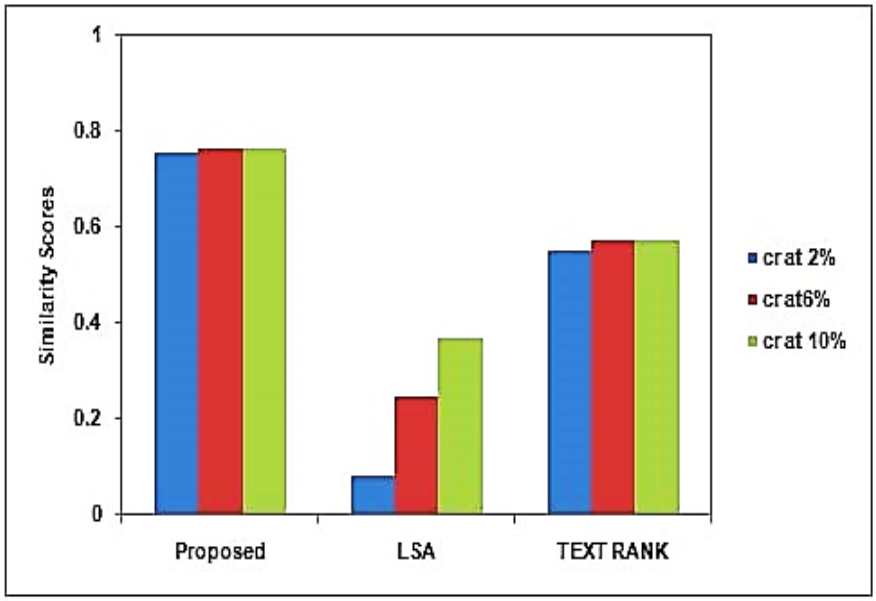

Fig. 5 Topic similarity at various compression ratios against Proposed, LSA, and TextRank algorithms

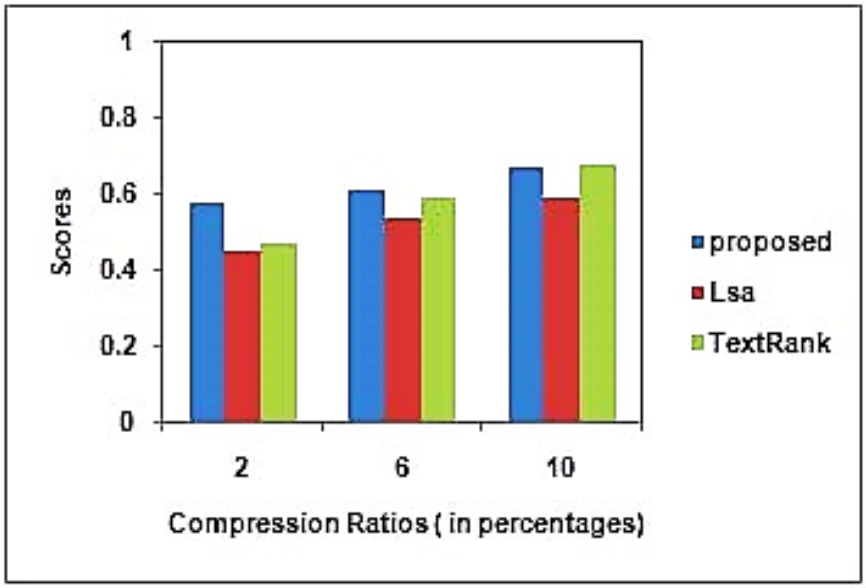

Fig. 6 Manual summary scores of the proposed and existing summarizers at different compression ratios

Table 1. Correlation of Proposed, LSA, and TextRank algorithms with Precision, Recall, and F1 score at different compression ratios

Algorithm	**Evaluation metric**	**COMPRESSION RATIO AT 2%**			**COMPRESSION RATIO AT 6%**			**COMPRESSION RATIO AT 10%**		
		Precession	Recall	F1 Score	Precession	Recall	F1 Score	Precession	Recall_	F1_Score
PROPOSED	**ROUGE 1**	0.556338	0.475833	**0.512946**	0.345214	0.739167	**0.470629**	0.239991	0.830000	**0.372326**
LSA	ROUGE 1	0.566372	0.320000	0.408946	0.345214	0.718333	0.466324	0.237552	0.815000	0.367877
TEXTRANK	ROUGE 1	0.536654	0.475833	0.504417	0.250696	0.750833	0.375887	0.175303	0.831667	0.289569
PROPOSED	**ROUGE 2**	0.108604	0.164220	**0.130743**	0.098947	0.229333	**0.138247**	0.075126	0.310217	**0.120959**
LSA	ROUGE 2	0.118168	0.066722	0.085288	0.089343	0.185988	0.120704	0.074101	0.254379	0.114770
TEXTRANK	ROUGE 2	0.127940	0.113428	0.120248	0.079043	0.236864	0.118531	0.059030	0.280234	0.097519
PROPOSED	**ROUGE L**	0.223944	0.152500	**0.181442**	0.127232	0.295000	**0.177786**	0.105681	0.395933	**0.166832**
LSA	ROUGE L	0.227139	0.128333	0.164004	0.120156	0.250833	0.162480	0.095417	0.351667	0.150106
TEXTRANK	ROUGE L	0.171053	0.151667	0.160777	0.088481	0.265000	0.132666	0.064992	0.308333	0.107355
PROPOSED	**ROUGE Lsum**	0.229944	0.169500	**0.195149**	0.129232	0.299000	**0.180465**	0.115681	0.365833	**0.175779**
LSA	ROUGE Lsum	0.227139	0.128333	0.164004	0.130156	0.270833	0.175818	0.105417	0.361667	0.163250
TEXTRANK	ROUGE Lsum	0.171053	0.151667	0.160777	0.088481	0.265000	0.132666	0.064992	0.308333	0.107355

Table 2. The topic similarity of Proposed, LSA, and TextRank algorithms at different compression ratios

Algorithm	**COMPRESSION RATIO**		
	2%	6%	10%
Proposed	0.75380313	0.76545960	0.7654596
LSA	0.08000000	0.24826726	0.3686027
TEXT RANK	0.55172644	0.57172644	0.5717263

Table 3. Change of words in machine-generated summary to human summary with the same meaning but not appropriately matched in a sentence

S.NO	MACHINE SUMMARY	HUMAN SUMMARY
1	Mankind	Humankind
2	Charge over	Rule over
3	Likeness	Image
4	Helper	Partner
5	Lord	God
6	God	Lord
7	Blameless	Righteous
8	Punished	Cursed
9	Vault	Heavens
10	Bag	Sack

4. Discussion

This section aims to explain the implementation of the proposed work by performing text summarization on a Bible text, the book of Genesis document, to achieve a human-level summary. After preprocessing the bible text, the topic modeling module identifies the several latent topics presented in the source text. Finally, the proposed model generates a summary for each identified text and combines it to form the generated summary from the document considered for summarization. Further, summary quality is evaluated in Precision, Recall, and F1 scores and compared with the various existing summarizers by the ROUGE metric at different compression ratios, topic similarity, and human-made summary quality.

The main results showed that the proposed approach had higher evaluation results than the existing approach with increased F1 score values. In addition, it outperforms the existing system by providing a better quality summary. Even though the results observed considerable differences between the representations, overall, the final ROUGE scores are more similar than expected. The source bible text document was preprocessed to identify and significantly make the document noise-free and clean the source text. It includes converting text to lower case (the model not to differentiate between words at the beginning and the middle of sentences), removing punctuations, removal of stopwords, and lemmatization (vocabulary and morphological analysis of words), which helped us to get maximum performance of summarization process and computation easy without significant loss of information.

Further, the LDA method generated the processed Bible text's topic words (cue words) to obtain the summary candidate sentences. Therefore, this research study used this LDA method for topic modeling as this method was found to be the most capable computationally and interpretable technique in implementing the English Bible dataset in our previously published research work [35]. According to the evaluation results, the LDA obtains a performance that is 75 % more than that of the LSA when using document similarity within the corpus and document similarity with the unseen document. In addition, the coherence score and word associations demonstrated by LDA were superior to those demonstrated by LSA.

This study showed that topic modeling could be beneficial for sentence selection to improve the topic similarity between machine-generated and source text documents. First, the LDA algorithm [37] [41] captures the topic words related to the bible text. Next, it facilitates the generated summary replicating the source text's complex context better. Then, it utilizes the most important candidate sentences by applying some heuristic methods established based on the topic diversity, stylistic features, and redundancy rate of a biblical text sentence. This method enables the generated summary to obtain at a specified compression ratio. Based on the produced summaries at different compression ratios for the input text by several ROUGE-metrics, it outperforms the proposed approach, as illustrated by the F1 scores (Table 1). This result proves that the proposed system performed better than the existing study methods based on the similarity score approach. Furthermore, the topics extracted from the source text were consistent in information retrieval and extraction.

Additionally, a sophisticated method such as Latent Semantic Analysis (LSA) [38] was applied to the present framework of topics-based sentence representation projected into the word-topic vectors into lower-dimensional spaces. However, it has been observed that an LSA-based algorithm is unsupervised, and they do not perform well while creating shorter summaries. As stated in the study of Witbrock and Mittal in 1999 [39], there is a claim that extractive summarization by the LSA method is not very efficient when creating concise short summaries. However, the TextRank algorithm, used for automatically summarizing large amounts of text, is an example of an unsupervised ranking system based on a graph used to score sentences and can also be used for keyword extraction and sentence extraction. In this case, they extracted sentences with the application of TextRank. It is developed so that its performance can be carefully investigated due to its internal implementation of the Page Rank algorithm and the development of the similarity matrix. [40]. Hence, The amount of data may affect the comparison and extraction of topics; hence, a larger dataset may improve the analysis's overall performance with LSA and TextRank.

In this study, the generated summary captures most of the sentences, scoring 0.51 for ROUGE 1 and 0.13 for ROUGE 2 on the F1 measure, respectively. As a result of this, it is considered a good summary. However, as shown in Fig 5 and Table 2, the topic similarity explains how the machine-generated summary topics are determined by

calculating cosine similarity between topic distributions of summary with the source document. As we have ranked the sentence based on the most dominant topic words, the summary can easily capture the topics of the source document. To sum up, anytime users increase the size of the machine-generated summary, the topic similarity should also grow because it includes more topics from the source document. This similarity is because more topics are being summarized. LSA showed the worst topic similarity in this study, whose values vary from 0.08 to 0.36 from a compression ratio of 2% to 10%.

In comparison, the summary generated by the proposed algorithm's topic similarity has been maintained consistently, for example, whose values vary from 0.75 to 0.76 for compression ratios of 2% to 10%. Moreover, TextRank algorithm values stand between the LSA and proposed algorithm, respectively. Finally, the proposed algorithm can cover more topics from the source document. Furthermore, Figure 6 and Table 3 suggest that the generated summary does not seem reasonable in certain aspects compared with human summaries. One possible reason for this effect is that human annotators are responsible for generating reference summaries. On the other hand, this study's approach is more progressive and is based on extractive summarization. Because of this, it selects exact sentences from the document included in the automatically generated summary; however, it cannot restructure the sentences with their actual meaning.

5. Conclusion

This paper investigated and proposed an extractive summarization method for a bible data text document. As many tasks are involved in extracting text summarization to generate a summary, this study compared the proposed approach with other existing methods. The appropriate use of LDA based topic modeling algorithm captures the main topics of the source text. The generated summary was analyzed, and the results were compared to some of the most relevant and cutting-edge evaluation metrics. Three evaluation tasks were conducted on the generated summary to assess the performance of the summarizers, first by the ROUGE metric. They obtained a high F1 score of (0.512946) with a compression ratio of 2%. Secondly, the topic similarity of the machine-generated summary is determined by calculating cosine similarity between topic distributions of summary with the proposed algorithm's source document and maintaining consistency. Those values vary from 0.75 to 0.76 for compression ratios at 2% to 10%, respectively. Finally, this study assessed the system summaries manually. It graded them concerning their text quality, indicating that the proposed summarization model performed well in non-redundancy in summary at a higher compression ratio. These results showed that the proposed model has been quite successful. It confirms the initial hypothesis with promising results that could be readily used in practice and as a springboard for further research on summarization.

Conflict of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

References

[1] Hu, Ya-Han, Yen-Liang Chen and Hui-Ling Chou, Opinion Mining from Online Hotel Reviews – a Text Summarization Approach, Information Processing & Management. 53(2) (2017) 436–49. Doi:10.1016/j.ipm.2016.12.002.
[2] Oussous, Ahmed, Fatima-Zahra Benjelloun, Ayoub Ait Lahcen and Samir Belfkih, Big Data Technologies: A Survey, Journal of King Saud University - Computer and Information Sciences. 30(4) (2018) 431–48. Doi:10.1016/j.jksuci.2017.06.001.
[3] Uma, C, S Krithika, and C Kalaivani, A Survey Paper on Text Mining Techniques, International Journal of Engineering Trends and Technology. 40(4) (2016) 225–29. https://doi.org/10.14445/22315381/ijett-v40p237.
[4] Ye, Shiren, Tat-Seng Chua, Min-Yen Kan and Long Qiu, Document Concept Lattice for Text Understanding and Summarization, Information Processing & Management. 43(6) (2007) 1643–62. Doi:10.1016/j.ipm.2007.03.010.
[5] Steinberger, Josef, Massimo Poesio, Mijail A. Kabadjov and Karel Ježek, Two Uses of Anaphora Resolution in Summarization, Information Processing & Management. 43(6) (2007) 1663–80. Doi:10.1016/j.ipm.2007.01.010.
[6] Lloret, Elena, Laura Plaza and Ahmet Aker, Analyzing the Capabilities of Crowdsourcing Services for Text Summarization, Language Resources and Evaluation. 47(2) (2012) 337–69. Doi:10.1007/s10579-012-9198-8.
[7] Etemad, Abdul Ghafoor, Ali Imam Abidi and Megha Chhabra, A Review on Abstractive Text Summarization Using Deep Learning, 9th International Conference on Reliability, Infocom Technologies and Optimization Trends and Future Directions, ICRITO. (2021). Doi:10.1109/icrito51393.2021.9596500.
[8] Vishal Gupta, and Gurpreet Singh Lehal, Features Selection and Weight Learning for Punjabi Text Summarization, International Journal of Engineering Trends and Technology. 2(2) (2011) 45–48.
[9] Chaudhary, Nidhi, and Shalini Kapoor, Key Phrase Extraction Based Multi-Document Summarization, International Journal of Engineering Trends and Technology. 13(4) (2014) 148–53. https://doi.org/10.14445/22315381/ijett-v13p232.
[10] Gambhir, Mahak and Vishal Gupt, Recent Automatic Text Summarization Techniques: A Survey, Artificial Intelligence Review. 47(1) (2016) 1–66. Doi:10.1007/s10462-016-9475-9.
[11] Allahyari, Mehdi, Seyedamin Pouriyeh, Mehdi Assefi, Saeid Safaei, Elizabeth D, Juan B, and Krys Kochut, Text Summarization Techniques: A Brief Survey, International Journal of Advanced Computer Science and Applications. 8(10) (2017). Doi:10.14569/ijacsa.2017.081052.
[12] Mekuria, Getahun Tadesse, and Aniket S Jagtap, Automatic Amharic Text Summarization Using NLP Parser, International Journal of Engineering Trends and Technology. 53(1) (2017) 52–58. https://doi.org/10.14445/22315381/ijett-v53p210.

[13] Abdel-Salam, Shehab and Ahmed Rafea, Performance Study on Extractive Text Summarization Using Bert Models, Information. 13(2) (2022) 67. Doi:10.3390/info13020067.
[14] Sarker, Goutam, Antara Pal, and Saswati Das, A New Method of Text Categorization and Summarization with Fuzzy Confusion Matrix, International Journal of Engineering Trends and Technology. 49(2) (2017) 107–14. https://doi.org/10.14445/22315381/ijett-v49p217.
[15] El-Gedawy, Madeeh Nayer, Comparing PMI-Based to Cluster-Based Arabic Single Document Summarization Approaches, International Journal of Engineering Trends and Technology. 11(8) (2014) 379–83. https://doi.org/10.14445/22315381/ijett-v11p274.
[16] Anitha, Raahavi, Rehapriadarsini and Sudarshana S, Abstractive Text Summarization, Journal of Xidian University. 14(6) (2020) 854–57. Doi:10.37896/jxu14.6/094.
[17] Rawat, Mukesh, Mohd Hamzah Siddiqui, Mohd Anas Maan, Shashaank Dhiman, and Mohd Asad, Text Summarization Using Extractive Techniques, Process Mining Techniques for Pattern Recognition. (2022) 107–19. Doi:10.1201/9781003169550-9.
[18] Mishra, Ritwik and Tirthankar Gayen, Automatic Lossless-Summarization of News Articles with Abstract Meaning Representation, Procedia Computer Science. 135 (2018) 178–85. Doi:10.1016/j.procs.2018.08.164.
[19] Rodríguez-Vidal, Javier, Jorge Carrillo-de-Albornoz, Enrique Amigó, Laura Plaza, Julio Gonzalo and Felisa Verdejo, Automatic Generation of Entity-Oriented Summaries for Reputation Management, Journal of Ambient Intelligence and Humanized Computing, 11(4) (2019) 1577–91. Doi:10.1007/s12652-019-01255-9.
[20] Dhankhar, Sunil, and Mukesh Kumar Gupta, Automatic Extractive Summarization for English Text: A Brief Survey, Proceedings of Second Doctoral Symposium on Computational Intelligence. (2021) 183–98. Doi:10.1007/978-981-16-3346-1_15.
[21] Bhole, Pankaj, and Dr. A.J Agrawal, Single Document Text Summarization Using Clustering Approach Implementing for News Article, International Journal of Engineering Trends and Technology. 15(7) (2014) 364–68. https://doi.org/10.14445/22315381/ijett-v15p270.
[22] Deshpande, Anjali R, and Lobo LMRJ, Text Summarization Using Clustering Technique, International Journal of Engineering Trends and Technology. 4(8) (2013) 3348–51.
[23] Torres-Moreno, Juan-Manuel, Automatic Text Summarization: Some Important Concepts, Automatic Text Summarization. (2014) 23–52. Doi:10.1002/9781119004752.ch2.
[24] Luhn H. P, The Automatic Creation of Literature Abstracts, IBM Journal of Research and Development. 2(2) (1958) 159–65. Doi:10.1147/rd.22.0159.
[25] Gupta, Vishal and Gurpreet Singh Lehal, A Survey of Text Summarization Extractive Techniques, Journal of Emerging Technologies in Web Intelligence. 2(3) (2010). Doi:10.4304/jetwi.2.3.258-268.
[26] Carbinell, Jaime and Jade Goldstein, The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries, ACM SIGIR Forum. 51(2) (2017) 209–10. Doi:10.1145/3130348.3130369.
[27] Nomoto, Tadashi and Yuji Matsumoto, Supervised Ranking in Open-Domain Text Summarization, Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02. (2001). Doi:10.3115/1073083.1073161.
[28] Mani, Inderjeet, Automatic Summarization, Natural Language Processing. (2001). Doi:10.1075/nlp.3.
[29] Kiyoumarsi, Farshad, Evaluation of Automatic Text Summarizations Based on Human Summaries, Procedia - Social and Behavioral Sciences. 192 (2015) 83–91. Doi:10.1016/j.sbspro.2015.06.013.
[30] van der Lee, Chris, Albert Gatt, Emiel van Miltenburg and Emiel Krahmer, Human Evaluation of Automatically Generated Text: Current Trends and Best Practice Guidelines, Computer Speech & Language. 67 (2021) 101-151. Doi:10.1016/j.csl.2020.101151.
[31] Radev, Dragomir R., Simone Teufel, Horacio Saggion, Wai Lam, John Blitzer, Hong Qi, Arda Çelebi, Danyu Liu and Elliott Drabek, Evaluation Challenges in Large-Scale Document Summarization, Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - ACL '03. (2003). Doi:10.3115/1075096.1075144.
[32] Lin and Chin-Yew, ROUGE: A Package for Automatic Evaluation of Summaries, Text Summarization Branches Out, Barcelona, Spain: Association for Computational Linguistics. (2004) 74-81.
[33] Sarker, Goutam, Antara Pal, and Saswati Das. A Modified Optimal Clustering Technique for Image Categorization and Summarization, International Journal of Engineering Trends and Technology. 49(2) (2017) 99–106. https://doi.org/10.14445/22315381/ijett-v49p216.
[34] Blei, David M, Andrew Y Ng and Michael I Jordan, Latent Dirichlet Allocation, Journal of Machine Learning Research. 3 (2003) 993–1022.
[35] Garbhapu, VK and Bodapati P, A Comparative Analysis of Latent Semantic Analysis and Latent Dirichlet Allocation Topic Modeling Methods Using Bible Data, Indian Journal of Science and Technology. 13(44) (2020) 4474–82. Doi:10.17485/ijst/v13i44.1479.
[36] Dhivya J, Saritha A, A System for Detecting Network Intruders in Real-Time, SSRG International Journal of Computer Science and Engineering. 3(5) (2016) 34-37.
[37] Blei, David M, Probabilistic Topic Models, Communications of the ACM. 55(4) (2012) 77–84. Doi:10.1145/2133806.2133826.
[38] Landauer, Thomas K, Peter W. Foltz and Darrell Laham, An Introduction to Latent Semantic Analysis, Discourse Processes. 25(2-3) (1998) 259–84. Doi:10.1080/01638539809545028.
[39] Witbrock, Michael J. and Vibhu O. Mittal, Ultra-Summarization (Poster Abstract), Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '99. (1999). Doi:10.1145/312624.312748.
[40] Mihalcea, Rada, Tarau and Paul, TextRank: Bringing Order into Text, Proceedings of the Conference on Empirical Methods in Natural Language Processing. (2004) 404–11.
[41] Griffiths, Thomas L. and Mark Steyvers, A Probabilistic Approach to Semantic Representation, Proceedings of the Twenty-Fourth Annual Conference of the Cognitive Science Society. (2019) 381–86. Doi:10.4324/9781315782379-102.