Journal of Multi Disciplinary Engineering Technologies
Volume 18, Issue 01

Extractive Text Summarization Using Latent Semantic Analysis and Diversity Constraints

Aditya Tomar[1], Aditya [1], Amit Saxena[1], Dheeraj Sharma[1], Nupur Chugh[1*], Rakhi Joon[1]

[1]Department of Computer Science and Engineering, Bharati Vidyapeeth's College of Engineering, New Delhi, 110063, India.

*Corresponding author(s). E-mail(s): chughnupur1985@gmail.com;
Contributing authors: rakhi.du.cs@gmail.com;

Abstract

In this study we propose a single document text summarization technique using Latent Semantic Analysis (LSA) and diversity constraint in combination. The suggested method ranks sentences according to queries. Since information retrieval (IR) is not being considered in this instance, the query is generated using the TF-IDF (Term Frequency-Inverse Document Frequency) method. We determine which phrases have the highest IDF in order to create the query vector. We know that LSA uses vectorial semantics to examine the links between documents in a corpus or between sentences within a document and the important terms they convey, resulting in a list of concepts linked to the documents and terms. LSA facilitates the representation of a document's latent structure. Latent Semantic Indexing (LSI) is utilized for phrase selection inside the document. Sentences with scores are arranged with the assistance of LSI. Usually, the sentences with the highest scores are selected for the summary; however, in this case, we compute the diversity among the selected sentences to get the final summary, as a strong summary should have the greatest amount of variety possible. The suggested method is assessed using Dataset Trip advisor. We also implemented Random Forest Classifier after implementing LSA technique.

Keywords: LSA Algorithm, Performance Measurement, Pre-processing and Feature Extraction, Random Forest

1 Introduction

In this study, we propose a single document text summarization technique using Latent Semantic Analysis (LSA) and diversity constraint in combination [1]. The suggested method ranks sentences according to queries generated using the TF-IDF (Term Frequency-Inverse Document Frequency) method [2]. LSA, a technique rooted in vectorial semantics, examines the links between documents in a corpus or between sentences within a document and the important terms they convey, resulting

Published on: November, 2024

in a list of concepts linked to the documents and terms [3]. Latent Semantic Indexing (LSI) is utilized for phrase selection inside the document [4].

Selected sentences from the text corpus with the highest LSA scores are often used in the summary. Nonetheless, this study's computations aim to address a variety of topics within the chosen collection of phrases [5]. A diversified summary incorporates a greater variety of data, making it more comprehensive and practical. Dataset Trip Advisor is used to assess the viability of the suggested approach [6].

Additionally, a Random Forest Classifier is incorporated to enhance the summarizing process in order to complement the LSA methodology [7]. This improves the quality of summaries generated dynamically by using machine learning methods to make the process of choosing and ranking the appropriate language for summarizing the context more efficient [8].

This research's primary goal is to demonstrate an improvement over text summarizing techniques by integrating LSA with diversity limitations and evaluating it in real-world situations [9]. The techniques presented in this study will be tried and tested, and by evaluating the outcomes, more concepts, advancements, and novel techniques for automatic text summarization are to be developed in the future [10].

2 Methods

The Dataset used in this study are briefly described in the following section. It also goes into great detail about the method used to use a convolutional neural network to extract summary. As illustrated in fig. 1, the procedures performed for the suggested solution are as follows:

Fig. 1 Block Diagram for Text Summarization

2.1 Dataset

The dataset, known as "trip-advisor," consists of 7,358 records and 8 fields that aggregate carefully selected conversations and exchanges related to travel. Every row denotes a distinct strand, and the columns comprise crucial details that are necessary for comprehending and managing the content. These include "Thread-ID," which indicates the launch of each thread, "Title," which provides a brief description of the threads' subjects, "UserID," which refers to each subject identity of the users participating in the threads, and "Date," which indicates the post time. Additionally, "Postnum" displays the numbering of postings within threads, which makes it easier to analyze the data in a systematic manner. The "Text" column contains the postings' textual contents, which include the variety of user-shared ideas, discussions, and experiences. On the other hand, short posts or even summaries of threads may appear in the "Summary" column.

2.2 Preprocessing

The functions in the script below are intended to retrieve, parse, and print any information from internet publications.

Fig. 2 Block Diagram for Text Summarization

The "download-article" function's primary goal is to make the process of downloading articles easier by using the supplied URLs. It first verifies that the article hasn't been downloaded earlier by looking through its local directory. If not, it retrieves the article's HTML content using the requests library and saves it as an HTML file. This increases the speed and reliability of the data acquisition process by ensuring that the script receives the most recent version of the article when it is initially acquired [11]. The "parse-article" function form, which is derived from every webpage that was downloaded during each iteration, is displayed following the download process. It might make use of Beautiful Soup to parse the downloaded article's HTML content and extract pertinent data, including the article's ID, URL, headline, section, content, authors, and publication date. The function addresses a specific area of the HTML structure and extracts the items required for the study since CSS selectors are used. To facilitate further processing of the data as it passes through the pipeline, the retrieved data is subsequently converted to dictionary format. The script then sends out information about the particular article when it has been downloaded and parsed. This includes key details such as the publication time and the article text. To maintain readability and conciseness, the reprlib.Repr() class is utilized to limit the length of the printed text, ensuring that the output remains manageable and informative [12, 13].

While the script serves as a valuable starting point for obtaining and extracting information from news articles, it is acknowledged that additional preprocessing steps may be necessary for more comprehensive analysis or downstream natural language processing tasks. These steps could encompass tasks such as tokenization, removing stop words, stemming or lemmatization, and handling missing or irrelevant data. We have used sentence segmentation divides the entire text into sentences, which are then stored in an array together with their associated sentence places. Tokenization breaks down phrases into words for some feature computations. Stop word and punctuation removal removes common words like the, an, a, but, and, or, as well as any punctuation. We also used TF-ISF (Term Frequency - Inverse Document Frequency) is important for information retrieval systems. This study focuses on text summary for our Trip Advisor dataset.

2.3 Latent Semantic Analysis

Latent Semantic Analysis (LSA) stands as a pivotal natural language processing method, widely employed for tasks such as information retrieval and text summarization. Fundamentally, latent semantics is the idea that words and phrases in a document represent deeper, more significant ideas than just a list of words put together at random. The way the LSA algorithm works is that it examines the complex interactions that occur between words and the surrounding context of a particular document. It does this by identifying latent ideas via analysis and quantifying their importance through the application of a complex mathematical method called singular value decomposition (SVD). With the aid of SVD, LSA efficiently distills a document to highlight the key ideas while resulting in the least amount of text possible.

2.4 Random Forest

Random Forest is a well-known, strong, and efficient machine learning tool for regression and classification. Random Forest works well for extractive text summarization because it chooses and ranks

phrases that best represent the primary ideas of the text. In order to get the final choice, a number of decision trees that were trained using random subsets of data combine to form an ensemble learning technique. Among its benefits are reduced overfitting, increased accuracy, and resistance to data noise. Sentences are assessed in extractive summary based on their length, placement within the paragraph, and resemblance to the text's title. Each sentential input is classified as either relevant or irrelevant to the summary generation in a labeled dataset, which is used as a training set. Several decision trees are shown to emerge as a result of training on different subsets of the provided data. The trained model may then forecast which of the new paragraphs would contain pertinent sentences and may even provide scores or probability of the particular sentences in the target documents that would indicate their inclusion in the summary.

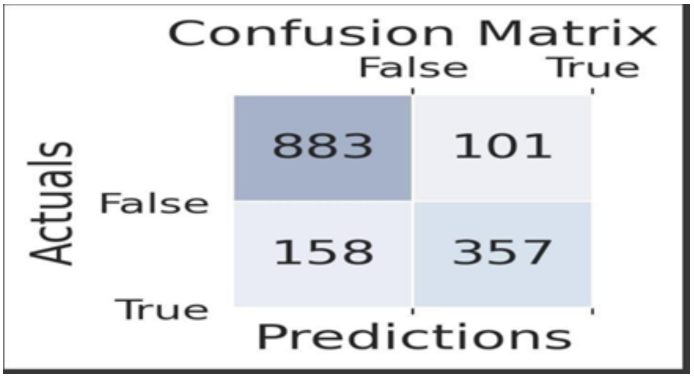

Fig. 3 Block Diagram for Text Summarization

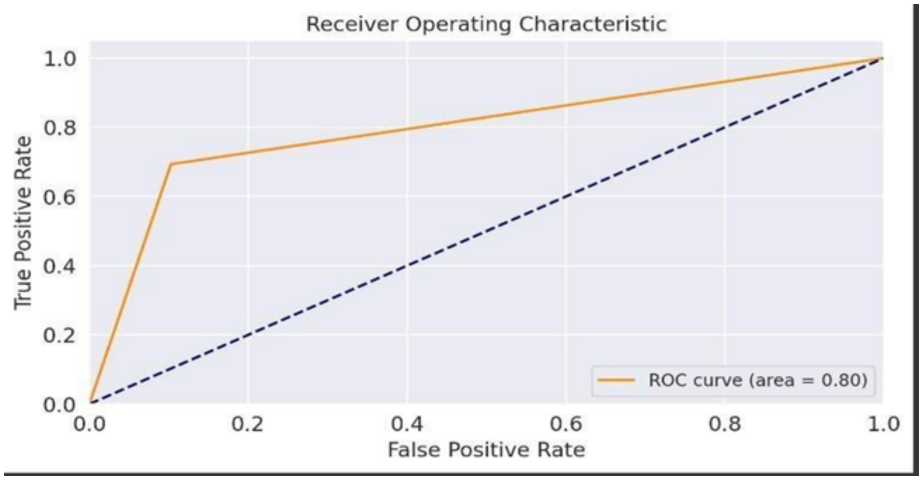

Fig. 4 ROC Curve representing ROUGH Score and Accuracy

3 Experimental Results

3.1 Performance Evaluation

The performance evaluation of our text summarizing project provides useful insights into the effectiveness of the various methodologies used. The amount of summary sentences created influences

the precision and recall of Latent Semantic Analysis (LSA). LSA with a greater number of summary phrases, particularly 10, has better precision (0.12) and recall (0.26) than LSA with only three summary sentences, which has poorer precision (0.03) and recall (0.05). Similarly, using indicator representation for summarizing produces different effects depending on the amount of summary sentences created. While the indication representation with three summary phrases has a good recall (0.83), accuracy remains poor (0.06). However, with ten summary phrases, accuracy increases (0.11) and recall declines (0.31) as shown in Table 1.

Table 1 Result of evaluation metrics on number of summary sentences samples

Algorithms & Number of Summary Sentences	Precision	Recall
LSA with num_sum_sentences = 3	0.03	0.05
LSA with num_sum_sentences = 10	0.12	0.26
Indicator representation with num_sum_sentences = 3	0.06	0.83
Indicator representation with num_sum_sentences = 10	0.11	0.31

Furthermore, the use of a Random Forest classifier dramatically improves summarization accuracy. The resulting ROUGE score of 0.34 and accuracy of 80% (measured using the ROC curve in Figure 4) show a significant improvement in summarization quality. The confusion matrix in Figure 3 gives a thorough evaluation of the classifier's performance, demonstrating a balanced prediction of summary and non-summary texts. This integration demonstrates the efficiency of machine learning approaches in enhancing the summarizing process, allowing for more accurate and fast extraction of crucial information from research publications.

4 Conclusion

In this paper, the practicality of the proposed methodology is demonstrated for text summarization techniques that highlights the nuanced interplay between precision, recall, and the number of summary sentences generated. Latent Semantic Analysis (LSA) and indicator representation have significant benefits and disadvantages, with LSA performing better with a larger number of summary sentences and indicator representation excelling in recall, particularly with fewer summary phrases. Integrating a Random Forest classifier considerably improves summarization accuracy, as seen by the increased ROUGE score and 80% accuracy estimated using the ROC curve. This integration demonstrates the ability of machine learning technologies to modify and optimize summarization procedures. Moving forward, these findings can inform the development of more effective summarization algorithms, allowing for improved extraction and synthesis of critical information from research articles and other textual sources.

Declarations

- The authors received no specific funding for this study.
- The authors declare that they have no conflicts of interest to report regarding the present study.
- No Human subject or animals are involved in the research.
- All authors have mutually consented to participate.
- All the authors have consented the Journal to publish this paper.
- Authors declare that all the data being used in the design and production cum layout of the manuscript is declared in the manuscript.

References

[1] Gelbukh, A.: Natural language processing. In: Fifth International Conference on Hybrid Intelligent Systems (HIS'05), Rio de Janeiro, Brazil (2015)

[2] Patil, A.P., Dalmia, S., Ansari, S.A.A., Aul, T., Bhatnagar, V.: Automatic text summarizer. In: 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI), New Delhi (2020)

[3] Moratanch, N., Gopalan, C.: A survey on extractive text summarization. Unpublished (2017)

[4] Albalawi, R., Yeap, T., Benyoucef, M.: Using topic modeling methods for short-text data: A comparative analysis. Frontiers in Artificial Intelligence (2020)

[5] Rajasundari, T., Palaniappan, S., Kumar, P.: Performance analysis of topic modeling algorithms for news articles. Journal of Advanced Research in Dynamical and Control Systems (2017)

[6] Barde, B.V., Bainwad, A.M.: An overview of topic modeling methods and tools. In: 2017 International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai (2017)

[7] Nokkaew, K., Kongkachandra, R.: Keyphrase extraction as topic identification using term frequency and synonymous term grouping. In: 2018 International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI NLP), Pattaya, Thailand (2018)

[8] Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. Journal of Machine Learning Research (2003)

[9] Gong, Y., Liu, X.: Generic text summarization using relevance measure and latent semantic analysis. In: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (2011)

[10] Wang, D., Zhu, S., Li, T., Gong, Y.: Multi-document summarization using sentence-based topic models. Unpublished (2009)

[11] Cherukuri, A.K., Srinivas, S.: Latent semantic indexing using eigenvalue analysis for efficient information retrieval. International Journal of Applied Mathematics and Computer Science (2020)

[12] Li, J., Fan, Q., Zhang, K.: Keyword extraction based on tf/idf for chinese news document. Wuhan University Journal of Natural Sciences (2007)

[13] Alghamdi, R., Alfalqi, K.: A survey of topic modeling in text mining. International Journal of Advanced Computer Science and Applications (2015)