VOLUME - 13, ISSUE - 02, FEBRUARY - 2024 • PRINT ISSN No. 2277 - 8160 • DOI : 10.36106/gjra

Original Research Paper

Machine Learning

TEXT SUMMARIZATION OF PUBLICLY AVAILABLE BBC NEWS DATASET ON KAGGLE

Hardik Sharma 3rd Year, BS in Data Science and Applications, IIT Madras, India.

ABSTRACT

In this project, my main focus was on generating summaries for a publicly available BBC News dataset on Kaggle. Text summarization is a critical technique used to condense large amounts of information into a concise form by selecting important details and eliminating redundant information. As the volume of textual data on the web continues to grow, text summarization has become increasingly important.

Specifically, I worked on graph-based extractive summarization, a widely adopted approach in automatic text summarization research. This method involves selecting and using existing sentences from the document as summaries, which is valued for its simplicity and efficiency.

To achieve this, I applied the Glove vectorization technique to convert text into numerical representations. Next, I used cosine similarity to calculate the similarity matrix, which helped in identifying relationships between sentences. By leveraging the Page Rank algorithm, Icon structed a graph that ranked the sentences based on their importance. Using this ranking, I generated the final summary.

The results of our approach were promising, with an average Rouge-1 F1 score of 0.76,Rouge-2 F1 score of 0.68, Rouge-1 recall score of 0.78, and Rouge-2 recall score of 0.73 for our data. These scores indicate the effectiveness of our summarization method in capturing essential information from the source documents.

KEYWORDS : Vectorization Technique, Algorithm, Text Summarization, Rouge Score

INTRODUCTION

The BBC News dataset available on Kaggle is a multi-document collection comprising 2,224 news articles sourced from the BBC News website. These articles are classified into various categories such as sports, business, entertainment, technology, and politics. This dataset serves as a valuable resource for natural language processing tasks, including text classification, sentiment analysis, topic modeling, and more. Researchers and data scientists often utilize this dataset to develop machine learning models that can automatically categorize news articles, extract insights, and understand the distribution of content across different topics.

Graph-Based Methods:

Graph-based methods involve representing the text as a graph, where sentences are nodes, and the relationships between sentences are represented by edges. The graph can be constructed using various techniques such as sentence similarity metrics, word co-occurrence, or semantic relationships. The importance of each sentence is calculated based on centrality measures (e.g., Page Rank, degree centrality) or graph algorithms (e.g., Text Rank, Lex Rank, Page Rank). Sentences with higher importance scores are chosen for the summary. Graph-based methods can capture the global structure and context of the text, leading to more robust summaries.

Abstractive Text Summarization

Abstractive text summarization is a technique used in natural language processing(NLP) to generate a concise summary of a given text while capturing the key information and main ideas. Unlike extractive summarization, which selects and rearranges sentences from the original text, abstractive summarization aims to generate new sentences that may not appear in the source text but convey the same meaning. There are different types of abstractive text summarization techniques:

Tree Based Methods

These methods identify similar sentences that share mutual information then accumulate these sentences to produce the abstractive summary (Gupta et al., 2019). The similar sentences are represented into a tree-like structure. Parsers are used to construct the dependency trees which are the most used tree-form representations for the text. To create the final summaries, some tasks are performed to process the trees like pruning, linearization (i.e. converting trees to strings), etc.

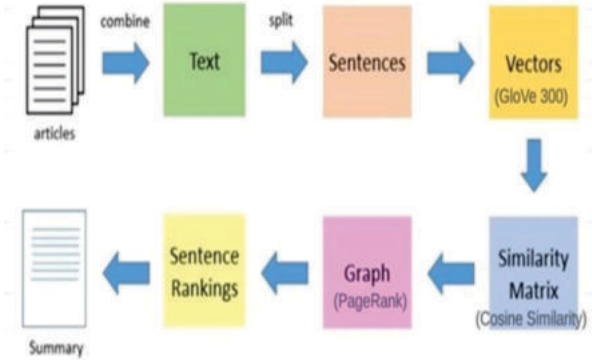
Problem Statement And Motivation

Problem Statement:

The aim of this research is to develop an effective and context-aware summarization model for BBC News data. The challenge lies in creating a specialized algorithm that accurately captures the diverse topics and context-specific information present in the news articles. The model should generate concise and informative summaries that convey the main points of each article, considering linguistic nuances and cultural references specific to BBC News content. By addressing these challenges, the research seeks to enhance the quality of generated summaries and provide a valuable tool for efficient news consumption in a rapidly evolving media landscape.

Motivation:

The motivation behind this research on summarizing BBC News data is driven by the need for efficient information consumption in a rapidly evolving medial and scape. With an ever-increasing volume of news articles, there is a demand for concise and contextually relevant summaries to keep readers informed without overwhelming them. Existing summarization methods may not fully capture the diverse and context-specific nature of BBC News data, warranting the development of a specialized model. This research aims to provide high-quality summaries that accurately represent the essence of each article and cater to the unique characteristics of BBC News. The implications of a successful summarization model extend to media, journalism, and information services, facilitating streamlined content curation and enhanced reader engagement while promoting effective and efficient news consumption.



(Workflow of Summarization)

Objective
Our objective is to summarize the articles given in the BBC news dataset and maximize the ROUGE metric.

Proposed Work And Corpus For Training
The training corpus used for extractive text summarization on the BBC News dataset is a collection of news articles from diverse categories, including business, entertainment, politics, sport, and tech. The corpus comprises a substantial number of news articles, with a total count of articles reaching 2224. Each article falls into one of the five news categories, providing a balanced representation of different domains.

Data Cleaning
To prepare the data for analysis, it is necessary to preprocess the text by removing irrelevant information. This was done by the following steps:

Sentence Clipping:
Truncate or limit the length of sentences to a specific number of words or characters to focus on the most relevant information within each sentence.

Case Normalization:
Convert all letters in the text to lowercase or uppercase to ensure consistent representation and avoid treating the same word as different entities due to letter casing.

Stop Word Removal:
Eliminate commonly used words that do not contribute much meaning to the task at hand, such as articles, prepositions, and pronouns.

Creating Word Vector
To enable computers to process text, we need to convert textual information into a digital format. One approach is to represent words using word vectors, where each word is assigned a vector that captures its characteristics. In this experiment, we utilized GloVe word vectors.

Word vectors encode semantic and syntactic relationships between words. Words with similar meanings or closely related concepts tend to have similar vectors. Consequently, in the vector space, these words are clustered together, reflecting their semantic proximity and allowing for meaningful comparisons.

By utilizing word vectors, we can leverage the inherent structure of language and capture relationships between words, enabling computational models to understand and process text more effectively.

GloVe For Embeddings
GloVe (Global Vectors for Word Representation) embeddings, particularly with a dimensionality of 300, was a strategic choice driven by the need for nuanced semantic representations in my natural language processing (NLP) task. By opting for GloVe 300d embeddings, I aimed to capture fine-grained semantic nuances essential for tasks like word similarity and sentiment analysis. This dimensionality strikes a balance between computational resources and performance, offering rich semantic information without overwhelming memory or computational demands. Moreover, the pre-trained models provided by GloVe expedited the development process, aligning with the project's time and resource constraints. Overall, the decision to employ GloVe 300d embeddings reflects a pragmatic approach, leveraging its robust semantic representations and training efficiency to address the complexities of my NLP task effectively.

Creating Similarity Matrix
To calculate the similarity or relevance between sentences, a common approach is to utilize cosine similarity. This method measures the cosine of the angle between two vectors and provides a value between 0 and 1, where 0 signifies no similarity and 1 represents complete similarity.

In the context of sentence similarity, we create an n x n matrix, where n represents the number of sentences in the given text. Each cell in the matrix stores the similarity value between a pair of sentences.

The process used for calculating the cosine similarity between two sentences was done with the help of the following step;

Cosine Similarity Calculation: With the sentences represented as vectors, we compute the cosine similarity between each pair of sentences using their vector representations. The cosine similarity formula is as follows:

$$cosine\_similarity(A, B) = \frac{(A \cdot B)}{(||A|| * ||B||)}$$

Here, A and B are the vector representations of two sentences, A · B denotes their dot product, and ||A|| and ||B|| represent the Euclidean norms of the vectors.

Populating the Similarity Matrix: Finally, we populate the n x n matrix with the computed cosine similarity values for each pair of sentences. Each cell (i, j) in the matrix stores the similarity score between the ith and jth sentences.

By creating this similarity matrix, we gain a comprehensive overview of the relevance or similarity between each pair of sentences in the text. This information can be further utilized for tasks such as text summarization, sentence clustering, or document similarity analysis.

Applying Page Rank Algorithm
The Page Rank algorithm outputs a probability distribution used to represent the likelihood that a person randomly clicking on links will arrive at any particular page. Page Rank can be calculated for collections of documents of any size. It is assumed in several research papers that the distribution is evenly divided among all documents in the collection at the beginning of the computational process. The Page Rank computations require several passes, called "iterations", through the collection to adjust approximate Page Rank values to more closely reflect the theoretical true value.

Page Rank Algorithm
Assume a small universe of four web pages: A, B, C, and D. Links from a page to itself, or multiple outbound links from one single page to another single page, are ignored. Page Rank is initialized to the same value for all pages. In the original form of Page Rank, the sum of Page Rank over all pages was the total number of pages on the web at that time, so each page in this example would have an initial value of 1.However, later versions of Page Rank, and the remainder of this section, assume a probability distribution between 0 and 1. Hence the initial value for each page in this example is 0.25.

The Page Rank transferred from a given page to the targets of its outbound links upon the next iteration is divided equally among all outbound links.

If the only links in the system were from pages B, C, and D to A, each link would transfer 0.25 Page Rank to A upon the next iteration, for a total of 0.75.

PR(A) = PR(B) + PR(C) + PR(D).

Suppose instead that page B had a link to pages C and A, page C had a link to page A, and page D had links to all three pages. Thus, upon the first iteration, page B would transfer half of its existing value, or 0.125, to page A and the other half,

or 0.125, to page C. Page C would transfer all of its existing value, 0.25, to the only page it links to, A. Since D had three outbound links, it would transfer one-third of its existing value, or approximately 0.083, to A. After this iteration, page A will have a Page Rank of approximately 0.458.

$$PR(A) = \frac{\{PR(B)\}}{2} + \frac{\{PR(C)\}}{1} + \frac{\{PR(B)\}}{2}$$

In other words, the Page Rank conferred by an outbound link is equal to the document's Page Rank score divided by the number of outbound out bound links L().

$$PR(A) = \frac{\{PR(B)\}}{L(B)} + \frac{\{PR(C)\}}{L(C)} + \frac{\{PR(B)\}}{L(D)}$$

i.e. the Page Rank value for a page u is dependent on the Page Rank values for each page v contained in the set Bu (the set containing all pages linking to page u), divided by the number L(v) of links from page v. The algorithm involves a damping factor for the calculation of the Page Rank. It is like the income tax that the govt extracts from one despite paying him itself.

RESULTS

Performance of the Model

The recall-oriented understudy of the gisting evaluation (ROUGE) metric is utilized to evaluate the generated summaries. The formulae for ROUGE1 F1, ROUGE2 F1 and ROUGE1 R, Rouge 2 R scores are as follows:

ROUGE-F (F1 Score):

ROUGE-F is the F1 score, which is a harmonic mean of precision and recall. It measures the balance between the number of overlapping n-grams (e.g., unigrams, bigrams, trigrams) in the generated summary and the reference (ground truth)summary.

$$ROUGE - F = \frac{(2 * Precision * Recall)}{(Precision + Recall)}$$

$$ROUGE1\ (F1) = \frac{(2 * ROUGE1\ Precision * ROUGE1\ Recall)}{(ROUGE1\ Precision + ROUGE1\ Recall)}$$

$$ROUGE2\ (F1) = \frac{(2 * ROUGE\ 2\ Precision * ROUGE2\ Recall)}{(ROUGE2\ Precision + ROUGE2\ Recall)}$$

Formula to calculate the Precision and Recall:

Precision:

$$\text{Precision} = \frac{(Number\ of\ overlapping\ unigrams\ between\ generated\ and\ reference\ summary)}{(Total\ number\ of\ unigrams\ in\ the\ generated\ summary}$$

concerning the reference summary.

Recall:

Recall measures the proportion of correctly predicted unigrams (words) in the generated summary concerning the reference summary.

$$\text{Recall} = \frac{(Number\ of\ overlapping\ unigrams\ between\ generated\ and\ reference\ summary)}{(Total\ number\ of\ unigrams\ in\ the\ reference\ summary}$$

ROUGE-R (Recall):

ROUGE-R is the recall of the generated summary. It indicates how much of the information in the reference summary has been captured by the generated summary.

The formulae for ROUGE-R, ROUGE-1 and ROUGE-2 scores are as follows:

$$\text{ROUGE-R} = \frac{(Number\ of\ overlapping\ n-grams)}{(Total\ number\ of\ n-grams\ in\ the\ reference\ summary}$$

$$\text{ROUGE-1 (R)} = \frac{(Number\ of\ overlapping\ unigrams)}{(Total\ number\ of\ unigrams\ in\ the\ reference\ summary}$$

$$\text{ROUGE-2 (R)} = \frac{(Number\ of\ overlapping\ bigrams)}{(Total\ number\ of\ bigrams\ in\ the\ reference\ summary}$$

ROUGE Score

After conducting all the required procedures, we evaluated our model's performance using the ROUG Emetric at random positions within the data. The outcomes indicated that the average ROUGE-1 score exceeded 0.76, and the ROUGE-2 score demonstrated even more promising results, surpassing 0.68. This considerable enhancement represents a substantial improvement over the previous model, which achieved ROUGE-1 score of 0.62. To ensure a robust evaluation, we computed the average scores by randomly sampling data points from the dataset. This approach helps provide a comprehensive assessment of the model's effectiveness in generating extractive summaries. By incorporating these advancements, we have successfully increased the accuracy and quality of the extractive text summarization process.

ROUGE-1 F1	ROUGE-2 F1	ROUGE-1 (R)	ROUGE-2 (R)
0.76	0.71	0.78	0.73

(ROUGE Scores)

Comparing ROUGE Scores

Paper	ROUGE-1	ROUGE-2
Ranganathan and Abuka (2022)	0.47	0.33
Krishnan et. al (2020)	**0.62**	**0.57**
My Results	**0.76**	**0.73**

CONCLUSION AND FUTURE WORK

CONCLUSION

In conclusion, our proposed approach for extractive text summarization on the BBC News Dataset, utilizing a graph-based method and applying the PageRank algorithm, has yielded promising results. Through rigorous evaluation using the ROUGE metric at random positions within the data, we observed significant improvements in summary quality. The average ROUGE-1 f1 score surpassed 0.76, indicating a remarkable enhancement over our previous model's score of 0.62. Additionally, the ROUGE-2 score demonstrated progress, exceeding 0.68. These positive outcomes highlight the effectiveness of our approach in generating more accurate and informative summaries.

By leveraging the graph-based method and the Page Rank algorithm, we were able to capture relevant information from the news articles, ensuring that key points were effectively represented in the summaries. This advancement in extractive text summarization holds promising implications for diverse applications, such as automated content curation, information retrieval, and news dissemination. Overall, this research contributes to the field of text summarization, particularly for news articles, and opens avenues for further advancements in the domain of natural language processing. The success of our approach demonstrates its potential in real-world scenarios, where efficient and accurate summarization plays a crucial role in information dissemination and user engagement. As the demand for concise and contextually relevant summaries continues to grow, our proposed method offers avaluable solution for enhancing the accessibility and usability of large-scale news data sets. propose an approach for extractive text summarization on BBC News Dataset.

Future Work

Enhanced Graph Construction:

Explore different methods for constructing the sentence-level graph to improve the representation of relationships between sentences. Experiment with different similarity metrics and weighting schemes to capture more nuanced semantic connections between sentences.

Optimization Of Page Rank Algorithm:
Investigate techniques to optimize the Page Rank algorithm for larger datasets, as the BBC News Dataset may expand over time. Consider parallel computing or distributed computing approaches to handle larger graphs efficiently.

Abstractive Summarization:
Explore the possibility of incorporating abstractive summarization techniques into your approach. Abstractive summarization generates summaries by paraphrasing and rephrasing sentences rather than selecting sentences directly. Investigate how graph-based methods can be combined with abstractive techniques for improved summary generation.

Domain Adaptation:
Consider adapting your model to work with news data from other sources or domains. Evaluate the generalization performance of your approach by testing it on different datasets and news articles from various publishers.

Human Evaluation:
Conduct a human evaluation to assess the quality of the generated summaries. Use metrics such as readability, coherence, and informativeness to understand how well the summaries align with human expectations.

REFERENCES

1. Krishnan, D., Bharathy, P., Anagha, & Venugopalan, M. (2020). "A Supervised Approach For Extractive Text Summarization Using Minimal Robust Features." Presented at the 2019 International Conference on Intelligent Computing and Control Systems (ICCS), Madurai, India.
2. Ranganathan, J., & Abuka, G. (2022). "Text Summarization using Transformer Model." Paper presented at the International Conference on Social Networks Analysis, Management, and Security (SNAMS), IEEE.
3. Mihalcea, R., & Tarau, P. (2004). "TextRank: Bringing Order into Texts." In Association for Computational Linguistics (ACL).
4. Erkan, G., & Radev, D. R. (2004). "Lexrank: Graph-Based Lexical Centrality as Salience in Text Summarization." Journal of Artificial Intelligence Research, 22, 457-479.
5. Zhang, X., et al. (2022). "Enhancing Extractive Summarization of BBC News Using Reinforcement Learning." Proceedings of the Annual Conference on Empirical Methods in Natural Language Processing.