Journal of Scientific & Industrial Research
Vol. 81, September 2022, pp. 952-962
DOI: 10.56042/jsir.v81i09.53185

Unsupervised Extractive News Articles Summarization leveraging Statistical, Topic-Modelling and Graph-based Approaches

Utpal Barman[1]*, Vishal Barman[2], Nawaz Khan Choudhury[2], Mustafizur Rahman[2] & Shikhar Kumar Sarma[3]

[1]The Assam Kaziranga University, Jorhat 785 006, Assam

[2]Girijananda Chowdhury Institute of Management and Technology, Guwahati 781017, Assam

[3]Gauhati University, Guwahati 781014, Assam

Received 03 August 2021; revised 24 August 2022; accepted 25 August 2022

Due to the presence of large amounts of data and its exponential level generation, the manual approach of summarization takes more time, is biased, and needs linguistic professional experts. To avoid these substantial issues or to generate a succinct summary report, automatic text summarization is very much important. Three different approaches namely the statistical approach such as Term Frequency Inverse Document Frequency(TF-IDF), the topic modeling approach such as Latent Semantic Analysis (LSA), and graph-based approaches such as TextRank were applied to generate a concise summary for the benchmark the British Broadcasting Corporation (BBC) news articles summarization dataset. The domain-specific implementations of each approach in the five domains of the dataset and domain-agnostic prospects were explored in the paper while drawing various insights. The generated summaries were evaluated using the Recall-Oriented Understudy for Gisting Evaluation (ROUGE) framework, leveraging precision, recall, and f-measure metrics. The approaches were not only able to achieve a commendable ROUGE score but also outperform the previous works on the dataset.

Keywords: LSA, NLP, ROUGE, TextRank, TF-IDF

Introduction

Generation of data occurs at an exponential level on regular basis, mostly comprising of unstructured data in textual form. Estimation measures that up to 463 exabytes of data to be generated by 2025. Therefore, an efficient and accurate data handling strategy is important to control the data being generated everyday. Automatic Summarization of Text (AST) is a solution that refers to the methodologies which automatically generate concise summaries with succinct and salient information from long texts.[1] Removal or minimization of redundant information is another objective of ATS.[2] Extractive and Abstractive are the two main categories of text summarization.[3] Extractive Text Summarization (ETS) involves identifying and directly extracting the most relevant information in succinct excerpts[1] to produce a succinct summary.[4] Abstractive Text Summarization (ATS) involves understanding the significant perspective of a text to produce a summary.[5] Unlike ETS, here the salient features or excerpts are not directly selected rather they are generated in clear natural language.[6] ATS is comparatively more complicated than ETS as it generates novel sentences using sentence rephrasing and suggesting new words in context to the source word and sentence.[1] Coherency in the generated summary is expected for abstractive summarization to enhance the readability and grammatical accuracy. In this paper, we centered our focus on ETS of news articles utilizing a statistical approach namely Term Frequency-Inverse Document Frequency (TF-IDF), a topic modelling approach namely Latent Semantic Analysis (LSA), and a graph-based algorithm namely TextRank.[7]

The TF-IDF is mainly a frequency-driven statistical approach where the score of each document or in this paper, a sentence was calculated to generate a consolidated score and further sorted to rank the sentences.[1] The TF-IDF is a numerical approach that signifies the relevancy of a particular word or term in a document based on the appearance frequency of a word in a sentence.[8] The intuition behind the algorithm is that the weight is calculated based on the frequency of the term in a document. The more score in weight means the more important the term.

The LSA is a derivative of topic modelling approaches where latent denotes hidden that is the hidden topic or encoded topics are selected through a

*Author for Correspondence
E-mail: utpalbelsor@gmail.com

specialized dimensionality reduction technique namely Singular Value Decomposition (SVD) after the given document is converted into the corresponding document-term matrix. LSA helps to analyze the relationship between terms and the associated documents.[9] The mathematical technique, SVD ensures the identification of the hidden pattern of the relationships between the terms and concepts underlying in a provided set of documents.

TextRank[10] was derived from PageRank[11] and it is a graph ranking-based algorithm that was applied by Google to rank websites. Both keyword and sentence extraction are done using the TextRank where the node or vertex of the graph represents the extracted word and sentence.[10] In an algorithm, a certain threshold is considered to make the connections among the vertices and a node with the highest number of connections is considered the best.[10]

All three algorithms were perfect for their usability in the domain and language-independent implementations. The domain-specific aspects of each of the algorithms were examined in the study where a benchmark dataset, the BBC summary dataset of news articles was leveraged for this research. Five domains are encompassed in the BBC dataset as Sports, Entertainment business, Technology, and Politics.[12]

Global Vectors (GloVe) were used as static word embeddings13 and a TF-IDF vectorizer was used for vectorization on the dataset. The model performance was estimated using ROUGE metrics,[14] leveraging recall, precision, and f-measure metrics, and further, illustrated graphically as well as analyzed to explore valuable insights to be gained.

The commencement of the domain of ATS was initiated by Lunh[15] where auto abstracts of magazine articles and technical papers were extracted.[15] Other approaches were introduced in the later stages. Graph-based methods were introduced in the spectrum due to efficient approaches to the representation of the structure of a document.[16] TextRank was implemented as a derivative from PageRank where sentences were considered as the graph nodes and the similarity scores as the edges.[7] LexRank,[17] is reported based on the concept of sentence salience determination, and that is derived from PageRank.[16]

Semantic-based methods were also implemented where LSA was most commonly used by El-K assas *et al*.[1] In this algorithm, an unsupervised approach was taken to represent text-based semantics relying on the observable co-occurrence of words. SVD which was implemented in the input matrix contributed to the identification of relationships among terms and hidden topics, resulting in eventual sentence ranking.[1]

The Recurrent Neural Network (RNN) was proposed by Chen and Nguyen[18] where the selective encoding of the relevant features at the sentence level took place and was later extracted.[18] Another approach accounts for a reinforcement learning-based sentence ranking approach for extractive summarization where it was observed that cross-entropy training was inappropriate for the summarization task.[4]

Transformer model implementation for ETS was recently introduced through BERTSUM[3], which was a simple variant of Bidirectional Encoder Representations (BERT).[19] BERT is a neural network-based method popularly implemented for its remarkable achievements in the NLP scenario.

The summary dataset of BBC news articles is leveraged in this study and was also employed in the previous work[5] where algorithms including Lexical Chains and WordNet were utilized to extract the summary. The words and sentences with higher positions were selected to form the final summary. Another such approach was devised by Ahmad *et al*.[6] using the same dataset and another dataset namely CNN News Dataset.

This research forwards the below-mentioned contributions toward the news article summarization.

i) The study was performed in a well-defined benchmark dataset called the BBC News Article dataset where the news of 5 different domains was summarized and analyzed.
ii) Unsupervised statistical, topic-Modelling, and graph-based approaches were considered for the study.
iii) Domain-specific and domain-agnostic prospects were explored in the paper while drawing various insights.
iv) Lexical Chains, TextRank, and TF-IDF algorithms were undertaken in which lexical chains outperformed the other two approaches in the dataset.

Materials and Method

About the Approach

Three unsupervised approaches namely TextRank, LSA, and TF-IDF were reported in this study. The TextRank algorithm, known as the graph ranking algorithm identifies the importance of the graph

vertices through globalized information in a recursive manner. The algorithm was derived from Google's PageRank Algorithm.[11] Initially, PageRank was used to compute the weights of the web pages which enhance the performance of the web search engine where the web pages were presented as a graph, and vertices of the graph denoted a web page. A web page was directed webpage if the page was linked to another web page. Assignments of weights were done using the following Eq. 1.

$$S(V_i) = (1-d) + d * \sum_{j \in In(V_i)} \frac{1}{|Out(V_j)|} S(V_j) \quad ...(1)$$

where, $S(V_i)$ represents the weights of webpages i and the damping factor is d and it is set to 0.85.(11) The Vi represents the inbound connection or link of the i and the Vj denotes the total number of outgoing connections or links.

Like above, the TextRank is considered as the upgrade version of the PageRank where sentences are considered instead of web pages. The similarity score between the sentences is computed recursively (Eq. 2) and that is represented as the edge of the graph after finding the similarity matrix from the vectors that are formed from the sentence similarity.

$$WS(V_i) = (1-d) + d * \sum_{V_j \in In(V_i)} \frac{w_{ji}}{\sum_{V_k \in Out(V_j)} w_{jk}} WS(V_j) \quad ...(2)$$

Here, w represents the edge weight and W denotes the vertex score of the graph. Vi and Vj are similar notes as PageRank.

The Term Frequency (TF) is referred to a weight that is determined by the frequency of incidence of a term in a document (Eq. 3).

$$tf(t,d) = \frac{f_d(t)}{\max_{w \in d} f_d(w)} \quad \ldots(3)$$

Here, tf, t, d, $f_d(t)$, and $\max_{w \in d} f_d(w)$ denotes the term frequency, document, occurrence of the term, and maximum occurrence of the document.

The IDF refers to the quantification of the inverse function of the number of documents in which a term or word occurs. It signifies the specificity of the term. To reduce the risk of unnecessary bias which is introduced through the usage of terms like 'a' an', the' etc., which are used more frequently but has an insignificant contribution to the meaning of the textual data, IDF is leveraged. It can be represented by the following formula.

$$idf(t,D) = \ln\left(\frac{|D|}{|\{d \in D: t \in d\}|}\right) \quad ...(4)$$

where, idf, t, d, D, and $\{d \in D: t \in d\}$ denotes the inverse document frequency, term, document, total documents, and document need to occur.

TF-IDF is the product of this tf and idf function of a particular term. It can be denoted by the following formula:

$$tfidf(t,d,D) = tf(t,d) \cdot idf(t,D) \quad \ldots(5)$$

Here, tfidf, t, d, D, tf(t,d), and idf(t, D) signifies the frequency-inverse document frequency function, term, document, total documents, term frequency and inverse document frequency.

The logarithmic term ensures the value of idf tends to zero for a term occurring frequently in a considerable number of documents. TF-IDF is a statistical approach ensuring TF-IDF score for each term which is eventually consolidated to find the score of each sentence of an article which helps in further ranking to perform extractive summarization.

LSA also known as Latent Semantic Indexing (LSI) is a topic modelling approach that ensures efficient extraction of hidden semantic structures of words and sentences for a given textual data. The features may not be the original features of the dataset but rather derived or encoded but it contributes essentially to the data. A latent word signifies hidden so the algorithm tries to extract hidden features through an algebraic-statistical approach. SVD is a mathematical concept that plays a key role in the working of this algorithm. It is a dimensionality reduction technique that performs the factorization of the matrix into three matrices. It is given by the formula:

$$a = U\Sigma V^T \quad \ldots(6)$$

where, A, U, $\sum$, and V denote the matrix, orthogonal matrix (Light Singular Value, diagonal matrix (Singular Value), and another orthogonal matrix (Right Singular Value)

LSA confirms the semantic relationship between the hidden components and each term to generate a score which is consolidated to find the sentence score to rank the sentences and perform extractive summarization. The methodology undertaken is depicted in Fig. 1.

About Dataset

The dataset reported in this study is the News Summary Dataset of BBC and which is collected from Kaggle. It is a publicly accessible standard dataset generally used for summarization, and the dataset was inherited from another dataset used by Greene and Cunningham.[12] From the year 2004 to 2005, around

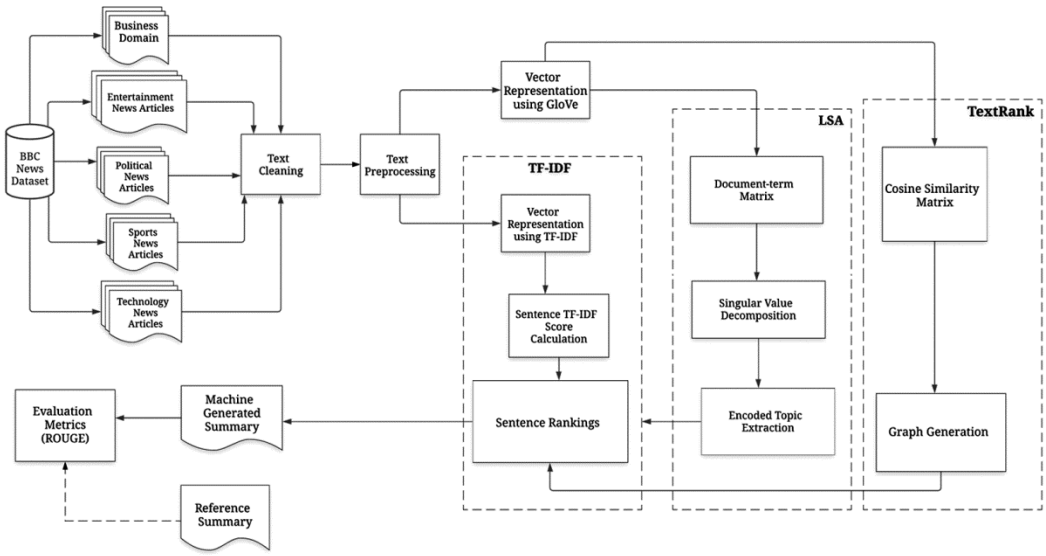

Fig. 1 — The flow chart of the adopted methodology

2225 documents were reported in the dataset in 5 domains, i.e, sports, technology, business, entertainment, and politics. The dataset contains the news article and the corresponding human-generated summaries. We undertook two different approaches: Domain Agnostic or Domain Independent and Domain Specific to gain valuable insights regarding the performance of each of the algorithms in five specific domains and a generic BBC news articles domain.

Data Cleaning and Preprocessing

Initially, sentence segmentation was performed in both the news articles and their corresponding summaries of the dataset where each article was converted into tokens where each token contained respective sentences of the article. Pointless noises are removed from the dataset by cleaning the data such as line breaks, hyphen encodings, quotation marks, special characters, punctuations, and bracketed texts. The news was then lowercased and conversion of possessive noun stakes place where for instance phrases like "puppy's collar" and "Rahul's books" were converted to "puppy collar" and "Rahul books", respectively as they signify the same entity. After this, the sentences were further tokenized to words as tokens where contraction mapping conversion takes place in which for instance, "aren't" is changed to "are not", "could've" to "could have" and so forth. The tokenized terms were checked for the presence of stop words or non-semantic irrelevant words that didn't provide any significant contribution to the meaning of the textual data like 'for', 'the', etc. where they were compared with a predefined stop word list and subsequently removed.[21] Then lemmatized was done for the tokenized words to use of predefined dictionary where all the inflected words of a root word like 'unhappy', 'happiness', 'happily', etc. are grouped under the root word 'happy'. All these data cleaning and preprocessing steps ensured the removal of any bias and further reduced the complexity of the process. Then the preprocessed words were joined to form the respective sentences.

Vector Representation of Words using GloVe

For further processing, the text was vectorized and the GloVE is reported in this paper for high dimensional representation of textual data. The similar mean words were clubbed together for similar representation[13] in the vector representation of the GloVe. A pre-trained vector of Wikipedia 2014 with a 100-dimension size is used in this study. The preprocessed word was depicted as the corresponding vector using the pre-trained GloVE and the final vector of each sentence is calculated by considering the average of these vectors.

TextRank Algorithm

Graph Generation and Similarity Matrix

The vector similarity refers to the similarity quantification of a vector and it is represented using cosine similarity using the following formula.[22]

$$\text{Cos } \alpha = \frac{A \times B}{|A| \times |B|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}} \quad \ldots(7)$$

where, A and B are the two vectors and α is the angle between them. Based on the similarity matrix a graph

was created with the vertices or nodes of the graph representing each sentence of the article and the edges as connections as depicted in Fig. 2.

0. Musical treatment for Capra film The classic film It's A Wonderful Life is to be turned into a musical by the producer of the controversial hit show Jerry Springer - The Opera.
1. Frank Capra's 1946 movie starring James Stewart, is being turned into a £7m musical by producer Jon Thoday.
2. He is working with Steve Brown, who wrote the award-winning musical Spend Spend Spend.
3. A spokeswoman said the plans were in the "very early stages" with no cast, opening date or theatre announced.
4. A series of workshops have been held in London and on Wednesday a cast of singers unveiled the musical to a select group of potential investors.
5. Mr Thoday said the idea of turning the film into a musical had been an ambition of his for almost 20 years.
6. It's a Wonderful Life was based on a short story, The Greatest Gift, by Philip van Doren Stern.
7. Mr Thoday managed to buy the rights to the story from Van Doren Stern'sfamily in 1999, following Mr Brown's success with Spend Spend Spend.
8. He later secured the film rights from Paramount, enabling them to use the title It's A Wonderful Life.

Sentence Ranking

Based on the inbound connections and influence of the generated graph, the TextRank algorithm[7] was implemented on a general graph and iteratively there is an addition to the weight of the node based on Eq. (2). Then the scores were normalized in a range of 1 and 0 where 1 denotes the highest and 0 denotes the lowest score. Sorting of sentences is done based on the score and rank.

LSA Algorithm

Document Term Matrix Generation

Like TextRank, the vectors generated by the GloVe static word embeddings were utilized even in the LSA algorithm.[7,13] The input article document was represented as a matrix to perform calculations in a subsequent process. This matrix was initialized as a document term matrix with the sentence being represented as a document as well as the words in that respective sentence as the terms in the matrix. The vectors of each word populated the cells of the matrix. The cells were ensured to denote the value of each word in the sentences.

Singular Value Decomposition (SVD)

SVD was an algebraic method that not only can reduce the dimensionality of a matrix but was also able to model the relationships between words or phrases and sentences. It was performed on the generated matrix where factorization of the matrix

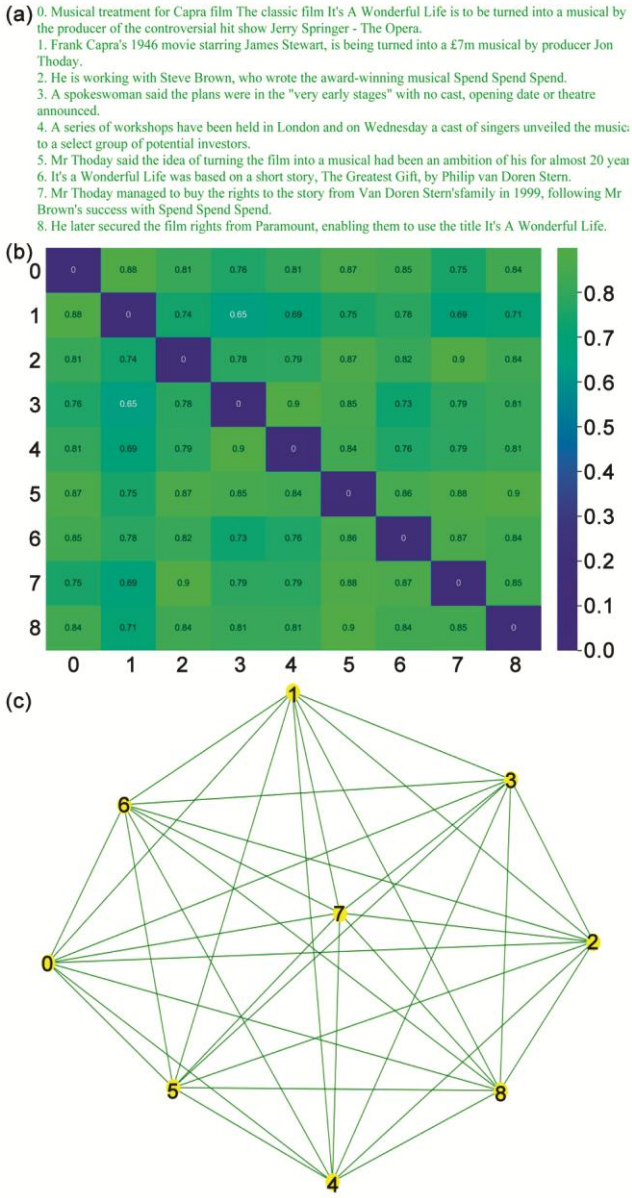

Fig. 2 — (a) Example Article from the dataset, (b) its corresponding similarity matrix, and (c) corresponding Graph Representation for TextRank

takes place based on Eq. (6). It was observed that most of the articles were well represented by the first hidden topic or latent component as shown in Fig. 3. The relatively shorter length of the news articles may account for this outcome.

Sentence Ranking

After performing SVD, the relative score between each term and the first latent topic was generated and consolidated to result in a sentence score, and eventually, the sentences of an article were ranked based on the sentence score. LSA was a topic

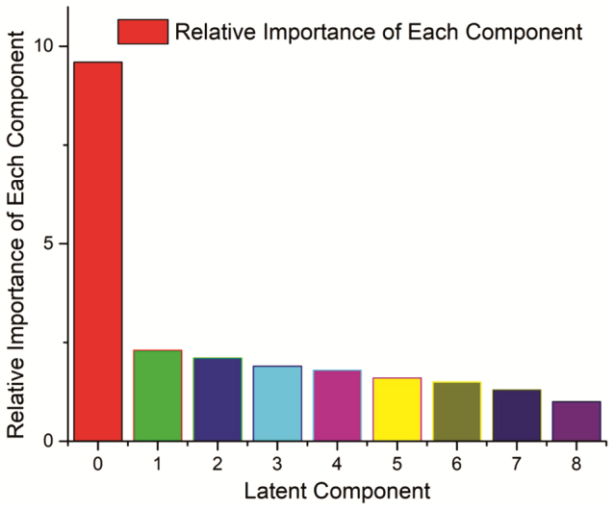

Fig. 3 — Relative Importance of hidden components generated by SVD for the example article in Fig. 2(a)

modeling approach, so the high number of common words among sentences confirmed the semantic relationships among sentences. It indicated that the meaning of the sentence was decided using the word it contained and the meaning of the words can be decided using the sentence that contains the word.

TF-IDF Algorithm

Vector Representation using TF-IDF

The TF-IDF scores of each term of an article were calculated based on Eqs (3)–(5) and a two-dimensional data matrix was generated numerically also known as a bag of words where each term or word of the article is represented with its corresponding TF-IDF score.

The weight in the TF-IDF is calculated based on the frequency of the term in a document. The more score in weight the more important term. Let's have the following Document of the BBC dataset from the politics category (Document no 20).

Document 1: Final appeals are being made for the government not to ditch the reform plan for England's secondary schools put forward by the Tomlinson report.

Document 2: The government's response to the plan for a four-tier diploma to replace all existing 14-19 qualifications is expected next week.

Document 3: His main concern was the reports that there would be a diploma - but only to replace existing vocational qualifications.

Document 4: The chief inspector of schools, David Bell, also said recently that GCSEs and A-levels should go.

The following Table 1 shows the TF-IDF example of the above documents where the number of words does not include stop words.

Sentence Scoring

The generated bag of words with higher TF-IDF value indicated that the word occurs more frequently in the sentence but less frequently in the whole document or the article. A consolidated sentence score or average score was evaluated for each sentence as shown in Fig. 4. The sentence scores were sorted and ranked. As observed in Fig. 4, the first sentence was the most important followed by the fifth sentence of the example article in Fig. 2(a) and the third sentence was the least important, based on the TF-IDF score of each sentence.

Summary Generation

Based on the sorted sentences from all three approaches, the summary was generated on six different retention rates. Retention rate signifies the total sentences present in the output summary.[6] It is given by the following Eq. 8.[(6)]

$$n = \left(\frac{A+B}{100}\right) \quad ...(8)$$

where, A is the percentage of retention rate, which in our case is 30, 40, 50, 60, 70, and 80, B is the total sentences in the article and n is the total sentences in the summary. ROUGE metrics are leveraged to appraise the performance of the three respective algorithms on each summary with these six different retention rates both in domain agnostic and domain-specific approaches.[14]

Results and Discussion

ROUGE[14] is a scoring algorithm mostly used for evaluation in summarization tasks where the summaries are compared using a similarity score. In this study, n-gram co-occurrence statistics are used for the performance evaluation of ROUGE-N. The statistics are used between the reference and machine-generated summary. Unigram and bigram of ROUGE are used to find the one-word co-occurrence and consecutive word occurrence. The ROUGE-N metric can be computed using the following formula.

$$\text{ROUGE} - \text{N} = \frac{\sum_{S \in \{\text{ Ref Summaries }\}} \sum_{n-gram \in S} \text{Count}_{\text{match}}\ (n-\text{ gram })}{\sum_{S \in \{\text{ Ref Summaries }\}} \sum_{n-gram \in S} \text{Count}\ (n-\text{ gram })} \quad ...(9)$$

where, N represents the n-gram length and the Count denotes the n-grams (maximum number) in the summary.

Table 1 — Execution of TF-IDF in terms of weight and frequency

No.	Token	Term count				Document count	IDF	TF × IDF			
		Doc 1	Doc 2	Doc 3	Doc 4			Doc 1	Doc 2	Doc 3	Doc 4
1	final	0.041	0	0	0	1	0.602	0.025	0	0	0
2	appeals	0.041	0	0	0	1	0.602	0.025	0	0	0
3	made	0.041	0	0	0	1	0.602	0.025	0	0	0
4	government	0.041	0	0	0	1	0.602	0.025	0	0	0
5	ditch	0.041	0	0	0	1	0.602	0.025	0	0	0
6	reform	0.041	0	0	0	1	0.602	0.025	0	0	0
7	plan	0.041	0.05	0	0	2	0.301	0.013	0.015	0	0
8	england_s	0.041	0	0	0	1	0.602	0.025	0	0	0
9	secondary	0.041	0	0	0	1	0.602	0.025	0	0	0
10	schools	0.041	0	0	0.0625	2	0.301	0.013	0	0	0.019
11	put	0.041	0	0	0	1	0.602	0.025	0	0	0
12	forward	0.041	0	0	0	1	0.602	0.025	0	0	0
13	Tomlinson	0.041	0	0	0	1	0.602	0.025	0	0	0
14	report	0.041	0	0	0	1	0.602	0.025	0	0	0
15	government_s	0	0.05	0	0	1	0.602	0	0.03	0	0
16	response	0	0.05	0	0	1	0.602	0	0.03	0	0
17	four-tier	0	0.05	0	0	1	0.602	0	0.03	0	0
18	diploma	0	0.05	0.05	0	2	0.301	0	0.015	0.015	0
19	replace	0	0.05	0.05	0	2	0.301	0	0.015	0.015	0
20	existing	0	0.05	0.05	0	2	0.301	0	0.015	0.015	0
21	14-19	0	0.05	0	0	1	0.602	0	0.03	0	0
22	qualifications	0	0.05	0.05	0	2	0.301	0	0.015	0.015	0
23	expected	0	0.05	0	0	1	0.602	0	0.03	0	0
24	next	0	0.05	0	0	1	0.602	0	0.03	0	0
25	week	0	0.05	0	0	1	0.602	0	0.03	0	0
26	main	0	0	0.05	0	1	0.602	0	0	0.03	0
27	concern	0	0	0.05	0	1	0.602	0	0	0.03	0
28	reports	0	0	0.05	0	1	0.602	0	0	0.03	0
29	would	0	0	0.05	0	1	0.602	0	0	0.03	0
30	—	0	0	0.05	0	1	0.602	0	0	0.03	0
31	vocational	0	0	0.05	0	1	0.602	0	0	0.03	0
32	Chief	0	0	0	0.0625	1	0.602	0	0	0	0.038
33	inspector	0	0	0	0.0625	1	0.602	0	0	0	0.038
34	David	0	0	0	0.0625	1	0.602	0	0	0	0.038
35	bell	0	0	0	0.0625	1	0.602	0	0	0	0.038
36	also	0	0	0	0.0625	1	0.602	0	0	0	0.038
37	said	0	0	0	0.0625	1	0.602	0	0	0	0.038
38	recently	0	0	0	0.0625	1	0.602	0	0	0	0.038
39	gcses	0	0	0	0.0625	1	0.602	0	0	0	0.038
40	a-levels	0	0	0	0.0625	1	0.602	0	0	0	0.038
41	go	0	0	0	0.0625	1	0.602	0	0	0	0.038

The ROUGE-L computes the longest common matching sequence of words. It requires in-sequence matches rather than consecutive matches that can reflect the sentence-level word order. Since we used six retention rates, ROUGE values for all the retention rates were taken and the average of the ROUGE scores counts was taken for both domain agnostic and domain-specific approaches. The precision, recall, and f-measure metrics were also utilized where recall was used to quantify the summary content of the reference which the system-generated summary was able to capture. The precision quantifies the trueness of the system-generated summary where the F-measure denotes the harmonic mean of both.

We tried to perform the evaluation on domain agnostic and domain-specific approaches in each of the domains of the Summary Dataset and gain valuable insights through the graphical representations, which are tabulated in the following Tables 2 & 3.

We considered the ROUGE-N metric for our graphical representation where, N was taken as 2, as it is stricter than ROUGE-1 and also checks the semantic structure more efficiently. It is observed that a commendable ROUGE score was achieved by all the algorithms. However, only TextRank was seen to be consistent in all five domains. The checking of semantic relationships between sentences in the TextRank algorithm may account for its consistent performance. However, TF-IDF was able to provide outstanding performance in three domains except for the Business and Politics domain where it underperformed. Possible reasons revolve around the intuition on which the TF-IDF algorithm was based which signifies that a higher TF-IDF value of an algorithm was related to the rare occurrence of the word in the document, which might not have been the case in the Business and Politics domain where redundant use of words was highly plausible. In the topic modeling approach, LSA comparatively underperformed the other two algorithms but was still able to achieve a commendable ROUGE score which even outperforms the state-of-the-art results of previous works.

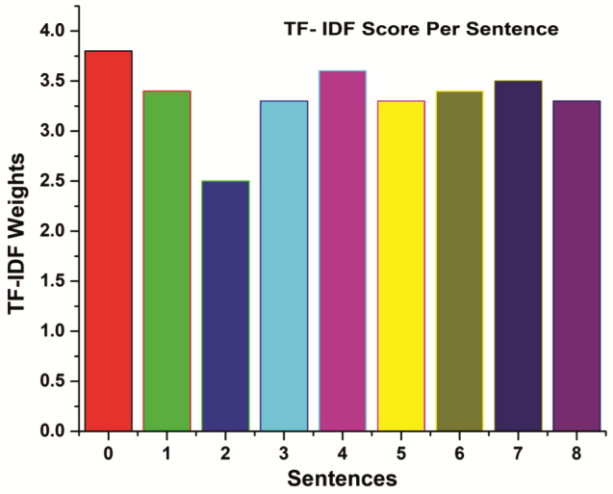

Fig. 4 — TF-IDF score of each sentence of the example article in Fig. 2(a)

In Tables 2 & 3, the performance of the algorithms was compared and analyzed using ROUGE-2 metrics through both domain agnostic approaches based on

Table 2 — Performance Measurement using ROUGE-L and ROUGE-N based evaluation of Domain Agnostic Approach

Algorithm	ROUGE- 1			ROUGE-2			ROUGE-L		
	Precision	Recall	F-measure	Precision	Recall	F-measure	Precision	Recall	F-measure
TextRank	0.68	0.869	0.75	0.59	0.76	0.65	0.68	0.83	0.74
LSA	0.71	0.76	0.71	0.58	0.63	0.58	0.68	0.70	0.68
TF-IDF	0.65	0.90	0.75	0.57	0.80	0.66	0.65	0.88	0.74

Table 3 — Performance Measurement using ROUGE-L and ROUGE-N for Business Domain

Domain	Algorithm	ROUGE- 1			ROUGE-2			ROUGE-L		
		Precision	Recall	F-measure	Precision	Recall	F-measure	Precision	Recall	F-measure
Business	TextRank	0.61	0.86	0.69	0.56	0.78	0.63	0.70	0.91	0.77
	LSA	0.30	0.50	0.37	0.14	0.27	0.19	0.35	0.53	0.42
	TF-IDF	0.28	0.50	0.35	0.11	0.24	0.15	0.30	0.51	0.37
Entertainment	TextRank	0.70	0.80	0.74	0.61	0.70	0.65	0.71	0.83	0.76
	LSA	0.64	0.60	0.60	0.50	0.50	0.48	0.61	0.61	0.60
	TF-IDF	0.72	0.90	0.79	0.66	0.83	0.73	0.73	0.93	0.81
Politics Domain	TextRank	0.54	0.76	0.62	0.47	0.66	0.57	0.51	0.73	0.59
	LSA	0.54	0.65	0.57	0.43	0.51	0.44	0.47	0.59	0.51
	TF-IDF	0.46	0.74	0.56	0.37	0.61	0.46	0.43	0.72	0.54
Sports Domai	TextRank	0.49	0.55	0.51	0.43	0.48	0.44	0.50	0.58	0.53
	LSA	0.44	0.45	0.43	0.35	0.37	0.35	0.47	0.47	0.46
	TF-IDF	0.73	0.93	0.80	0.67	0.86	0.74	0.74	0.92	0.81
Technology	TextRank	0.68	0.86	0.75	0.59	0.76	0.65	0.68	0.83	0.74
	LSA	0.71	0.76	0.71	0.58	0.63	0.58	0.68	0.70	0.68
	TF-IDF	0.65	0.90	0.75	0.57	0.80	0.66	0.65	0.88	0.74

Table 2 and domain-specific approaches based on Table 3. TF-IDF and TextRank both showcased similar performances followed by LSA which was nearly lagging. However, in the domain-specific approach, if we consider the Business domain, both LSA and TF-IDF underperformed miserably showing their dependency on the dataset for performance. TextRank was able to showcase commendable performance in the Business domain. Similarly, in Entertainment, TF-IDF was able to outperform LSA and TextRank. In the Sports domain, TF-IDF was able to outperform the other algorithms by a great margin. In the Politics domain, TextRank was able to achieve optimal results and finally, in the Technology domain, TF-IDF and TextRank achieved almost similar scores but TF-IDF exceeds by a small margin. LSA wasn't able to perform comparatively best in both domain agnostic and domain-specific approaches possibly due to lack of homogenous data and use of high complexity SVD technique which reduces performance in the long run. Though TF-IDF showcased outstanding performance in both domain agnostic and Entertainment, Sports, and Technology domain, it underperformed greatly in Business Domain. Plausible reasons for the underperformance include its basic intuition of rare occurrences being of most importance and lack of analysis of the semantic relationship between sentences. Finally, TextRank was able to report the uniformity in its performance in both domain agnostic and domain-specific approaches. Its graph-based modeling and sentence semantic relationship checking account for consistency. Hence, it can be considered the most optimal algorithm in the paper for the BBC News Articles Summary dataset.

Only two commendable works were observably performed in the dataset in the past as the dataset is fairly new. Janaki Raman and Meenakshi[5] focused on leveraging Lexical Chain and WordNet to perform extractive summarization. They considered the first article of Business Doman of the dataset entitled "Ad sales boost Time Warner profit" for the demonstration of their algorithm. They didn't mention the retention rate that they considered for the output, however, based on the output result they showed, we adjusted our retention rate. We have undertaken a similar approach but with the implementation of our algorithms. The performance comparison based on ROUGE-2 scores in the percent form, of their approach and our approach is depicted in Table 4.

Similarly, Ahmad *et al.*[6], focused on implementing three algorithms Lexical Chain (LC), TF-IDF, and TextRank on all the articles of the dataset. The approach they undertook involved keyword extraction in each of the algorithms so they considered the ROUGE-N metric with N=1 for the evaluation as they were concerned with the single words the algorithms would have extracted.[6] They considered three retention rates 30, 40, and 50. The Lexical Chain outperformed the other two algorithms. We considered the same retention rates and by application of our algorithms, we compared the performance of average ROUGE-1 scores in the following Table 5, and the best algorithm in their work, LC was considered for the graphical representation of the comparison in the following Table 5. Our approaches were able to out-perform the state-of-the-art results showcased in both works through a considerable margin as shown in Table 5.

It is observed from Table 5 that both TF-IDF and TextRank outperformed the LC and WordNet algorithm and LSA was able to outperform in Recall and F-measure metrics of ROUGE-2 scores. TextRank performed the best among all the algorithms. Similarly, in Table 5, the best algorithm in the previous work, Lexical chain was outperformed by both TF-IDF and TextRank. LSA was able to outperform in Precision and F-measure metrics of the ROUGE-1 scores.TF-IDF gave the best performance among the algorithms.

In the case of LSA, the SVD and dimension reduction techniques help in giving appropriate weight to the different terms present in the document. In the tf-idf approach, the term-frequency of many terms remains 0. In the TextRank algorithm, a graph-based algorithm is used to represent the sentences. Initially,

Table 4 — Comparison with previous work implementing Lexical Chain and WordNet

Algorithm	ROUGE-1			ROUGE-2			ROUGE-L		
	Precision	Recall	F-measure	Precision	Recall	F-measure	Precision	Recall	F-measure
LC & WordNet	72.00	38.60	50.25	45.99	24.61	32.06	60.68	36.09	45.26
TextRank	60.69	71.42	65.62	51.16	60.27	55.34	68.33	75.92	71.92
LSA	59.12	55.10	57.04	47.05	43.85	45.39	63.00	58.33	60.57
TF-IDF	42.38	60.54	49.85	27.75	39.72	32.67	60.18	42.76	49.99

Table 5 — Comparison with previous work implementing Lexical Chain, TF-IDF and TextRank

Algorithm	Average recall	Average precision	Average F-measure
Lexical Chain1	0.69	0.68	0.67
TF-IDF1	0.46	0.54	0.49
TextRank1	0.65	0.68	0.65
TextRank	0.77	0.76	0.76
LSA	0.59	0.79	0.67
TF-IDF	0.83	0.73	0.77

we are creating a sentence similarity matrix between the sentences by using the cosine similarity. The weights or the probability of each text document are the same. After many iterations, the new probabilities become stable and obtain the final values. Probabilities are nothing but the rank values of the text documents. From the explanation, we get to know that all three approaches fully depend on the frequency of appearance of the sentences. All approaches use frequency count while finding the summary

Conclusions

The undertaken approaches in the paper were not only able to obtain commendable ROUGE scores but also outperform the state-of-the-art results of previous works that were performed on the benchmark dataset. Valuable insights are also obtained from the comparative analysis of the three algorithms based on their performance. The TextRank algorithm reported and showcased consistent performance in all the mentioned domains and it can be considered the most optimal algorithm for the work. The statistical TF-IDF algorithm though performed outstandingly in Entertainment, Sports, and Technology domains even overtaking the optimal algorithm, TextRank of this study but underperformed in the Business domain and comparatively low in the Politics domain. Lack of semantic analysis and also the intuition that a rare occurrence of a word signifies the importance of the word, might not be applicable in the case of the articles in the Business and Politics domain. LSA underperformed in the Business, Politics, and Sports domains. Possible reasons for the underperformance include the high complexity of the SVD technique and homogeneity in the data of the domains, which might have considerably reduced the performance. However, it was observed that the algorithm was able to perform well in Entertainment and Technology domains. The three algorithms were able to attain commendable milestones in the performance which was clearly expressed through the paper, showcasing their credibility to extract salient information from the textual data.

References

1 El-Kassas W S, Salama C R, Rafea A A & Mohamed H K, Automatic text summarization: A comprehensive survey, *Expert Syst Appl*, **165** (2021) 113679, https://doi.org/10.1016/j.eswa.2020.113679.

2 Moratanch N & Chitrakala S, A survey on abstractive text summarization, *Int Conf Circuit, Power and Comput Technol (ICCPCT)*, 2016, 1–7, doi: 10.1109/ICCPCT.2016.7530193.

3 Liu Y, Fine-tune BERT for extractive summarization, *arXiv preprint arXiv:1903.10318*, (2019).

4 Narayan S, Cohen S B & Lapata M, Ranking sentences for extractive summarization with reinforcement learning, *arXiv preprint arXiv:1802.08636*, (2018). *Proc 2018 Conf North American Chapt Associat Comput Linguist: Human Language Technologies*, 2018, 1, 1747–1759.

5 Janaki Raman K & Meenakshi K, Automatic text summarization of article (NEWS) using lexical chains and wordnet—A review, *Artif Intell Tech Adv Comput Appl*, **130** (2021) 271–282.

6 Ahmad T, Ahmed S U, Ahmad N, Aziz A & Mukul L, News article summarization: Analysis and experiments on basic extractive algorithms, *Int J Grid Distrib Comput*, **13(2)** (2020) 2366–2379.

7 Mihalcea R & Tarau P, Textrank: bringing order into text, *Proc 2004 Conf Empiric Method Natur Language Process*, July 2004, 404–411.

8 Khan R, Qian Y & Naeem S, Extractive based text summarization using k-means and TF-IDF, *Int J Electron Bus*, **11(3)** (2019) 33.

9 Alami N, En-nahnahi N, Ouatik S A & Meknassi M, Using unsupervised deep learning for automatic summarization of Arabic documents, *Arab J Sci Eng*, **43(12)** (2018) 7803–7815.

10 Li A, Jiang T, Wang Q & Yu H, The mixture of textrank and lexrank techniques of single document automatic summarization research in Tibetan, in *8th Int Conf Intell Human-Machine Syst Cybernet (IHMSC)*, August 2016, **1**, 514–519.

11 Brin S & Page L, The anatomy of a large-scale hypertextual web search engine, *CNIS*, **30(1–7)** (1998) 107–117.

12 Greene D & Cunningham P, Practical solutions to the problem of diagonal dominance in kernel document clustering, *Proc 23rd Int Conf Machine Learn*, June 2006, 377–384.

13 Pennington J, Socher R & Mannin C D, Glove: Global vectors for word representation, *Proc 2014 Conf Empiric Method Natur Language Process (EMNLP)*, October 2014, 1532–1543.

14 Lin C Y, Rouge: A package for automatic evaluation of summaries, in *Text Summarization Branches Out*, July 2004, 74–81.

15 Luhn H P, The automatic creation of literature abstracts, *IBM J Res Dev*, **2(2)** (1958) 159–165.

16 El-Kassas W S, Salama C R, Rafea A A & Mohamed H K, EdgeSumm: Graph-based framework for automatic text summarization, *Inf Process Manag*, **57(6)** 2020 102264.

17 Erkan G & Radev D R, Lexrank: Graph-based lexical centrality as salience in text summarization, *Int J Artif Intell*, **22** (2004) 457–479.
18 Chen L & Le Nguyen M, Sentence selective neural extractive summarization with reinforcement learning, *11th Int Conf Knowl Syst Eng (KSE)*, October 2019, 1–5.
19 Devlin J, Chang M W, Lee K & Toutanova K, Bert: Pre-training of deep bidirectional transformers for language understanding, *arXiv preprint arXiv:1810.04805*, (2018).
20 Understand TextRank for Keyword Extraction by Python | by Xu LIANG | Towards Data Science. https://towardsdatascience.com/textrank-for-keyword-extraction-by-python-c0bae21bcec0 (accessed 2021-05-13).
21 Gupta V & Lehal G S, A survey of text summarization extractive techniques, *J Emerg Technol Web Intell*, **2(3)** (2010) 258–268.
22 Lahitani A R, Permanasari A E & Setiawan N A, Cosine similarity to determine similarity measure: Study case in online essay assessment, *4th Int Conf Cyber and IT Service Manag*, April 2016, 1–6.
23 Understanding TF IDF (term frequency - inverse document frequency). https://iq.opengenus.org/tf-idf/ (accessed 2021-08-01).
24 NLP — Text Summarization using NLTK: TF-IDF Algorithm | by Akash Panchal | Towards Data Science. https://towardsdatascience.com/text-summarization-using-tf-idf-e64a0644ace3 (accessed 2021-08-01).