Multi-document Summarization by Ensembling of Scoring and Topic Modeling Techniques

Rajendra Kumar Roul
raj.roul@thapar.edu
DCSE, TIET, Patiala, Punjab

Navpreet
navpreet705@gmail.com
DCSE, TIET, Patiala, Punjab

Saif Nalband
saif.nalband@thapar.edu
DCSE, TIET, Patiala, Punjab

Abstract

With the growing volume of text, finding relevant information is increasingly difficult. Automatic Text Summarization (ATS) addresses this by efficiently extracting relevant content from large document collections. Despite progress, ATS faces challenges like managing long, repetitive sentences, preserving coherence, and maintaining semantic alignment. This work introduces an extractive summarization approach based on topic modeling to address these issues. The proposed method produces summaries with representative sentences, reduced redundancy, concise content, and strong semantic consistency. Its effectiveness, demonstrated through experiments on DUC datasets, outperforms state-of-the-art techniques.

1 Introduction

Automatic Text Summarization (ATS) condenses data into concise summaries (Shakil et al., 2024), with research focusing on improved methodologies. Key ATS tasks include text understanding, content compression, summary representation, and relevant information identification (Alami Merrouni et al., 2023). ATS involves three stages: interpretation, transformation, and generation. Challenges remain in identifying implicit information, avoiding redundancy, and maintaining readability. ATS can be abstractive or extractive (Khurana and Bhatnagar, 2022). Abstractive summarization uses natural language generation, while extractive summarization involves text representation, relevance scoring, and key sentence selection. Summaries can be generic or query-focused (Tawong et al., 2024; Roul et al., 2019), and indicative or informative. Topic modeling, a statistical method, enhances extractive summarization by clustering semantically similar words into topics, ensuring relevance and coherence (Rani and Lobiyal, 2022; Roul and Arora, 2019).

Numerous studies have explored text summarization via topic modeling (Verma et al., 2022; Roul, 2021; Jiang et al., 2024). However, challenges persist, including an optimal topic number for LDA, limited diversity, sentence overlap, low representativeness, readability issues, long sentences, poor semantic relationship accounting, and key sentence omission. This study proposes a novel multi-document extractive summarization technique integrating topic modeling with unified scoring mechanisms and semantic sentence similarity, improving summary quality by addressing these challenges. The contributions of this paper are as follows:

- **Retention of Stopwords:** The summarization method retains all stopwords, ensuring that the syntactic structure of sentences is preserved. This preservation is vital for maintaining the integrity of sentences, regardless of their length.
- **Automatic Topic Determination:** A heuristic approach automatically identifies the optimal number of topics for Latent Dirichlet Allocation (LDA). This technique effectively captures relevant topic terms while promoting diversity among the topics in the generated summary.
- **Consistent Summary Generation:** Topic modeling with LDA is applied to produce a coherent summary. The summary is composed of succinct, representative sentences that minimize repetition and convey meaningful semantic information.
- **Sentence Organization:** The importance of each sentence is assessed by combining the KL-divergence measure with the silhouette coefficient. Based on these importance scores, the selected sentences are systematically ordered within the final summary.

2 Methodology

2.1 Preprocessing of documents

Let C be a corpus containing documents $D = \{d_1, d_2, \ldots, d_x\}$. The documents are merged into a unified set $D_{\text{large}} \subseteq C$, independent of order. From D_{large}, a set S with n sentences is formed by extracting all sentences. Words and documents are represented as a matrix, where each document d_i is a vector using Term Frequency-Inverse Document Frequency (TF-IDF). The weight of the j^{th} word in the i^{th} document is w_{ji}.

2.2 Topic modeling and Heuristic method

i. T_{initial} is initialized as $T_{\text{initial}} = 2 \cdot d$, where d is the total number of documents in $D_{\text{large}} \subseteq C$.

ii. Topic modeling is performed on corpus C using Latent Dirichlet Allocation (LDA) to generate T_{initial} topics. The Gensim library[1] in Python is used.

iii. Topic similarity (Topic_i, Topic_j) is computed using Jensen-Shannon Divergence, Kullback-Leibler Divergence, and Hellinger Distance. A similarity matrix records average scores, with $k = T_{\text{initial}}$ initialized and $\text{Topic}_{ii} = 1$ for diagonal elements.

iv. A similarity threshold of 0.45, optimal through systematic testing from 0.25 to 0.95 in 0.05 increments[2], decrements T_{initial} if any off-diagonal entry exceeds 0.45:

$$T_{\text{initial}} \leftarrow T_{\text{initial}} - 1.$$

v. Steps (ii)–(iv) are repeated until all off-diagonal similarities fall below 0.4. The final topic count is assigned as:

$$T_{\text{req}} \leftarrow T_{\text{initial}}.$$

2.3 Selection of candidate sentences from topic clusters

Topic modeling with LDA generates T_{req} topics from the input corpus C. Due to the corpus's high density (excess sentences relative to the ideal summary size), candidate sentences are selected from each topic T based on the following criteria:

i. Score based on Representativeness: The representativeness (rpe) of a sentence s_i measures its alignment with the document d's central theme. It is calculated using the cosine similarity ($cos\text{-}sim$) between s_i and the document centroid s_c, as shown in Equation 1:

$$rpe(s_i) = cosine\text{-}similarity(s_i, s_c) = \frac{s_i \cdot s_c}{\|s_i\| \, \|s_c\|} \quad (1)$$

Here, the centroid s_c is computed by summing the vector representations of all sentences within d, as shown in Equation 2, where n is the total number of sentences in the document:

$$s_c = \sum_{i=1}^{n} s_i \quad (2)$$

ii. Score based on Diversity: To prioritize uniqueness and reduce redundancy, the diversity score (div) is calculated for each sentence s_i. It is defined as the minimum similarity of s_i with all other sentences s_j in the document ($j \neq i$), as expressed in Equation 3

$$div(s_i) = \min\left(sim(s_i, s_j)\right), \quad \forall j \neq i \quad (3)$$

iii. Score based on Length: Preference is given to shorter sentences in the summary. The length score (len) of a sentence depends on its real length (rl, total words) and its effective length (el, distinct words). The formula is given in Equation 4 where n represents the total number of sentences:

$$len(s) = \frac{el(s)}{\max\left(el(s_j)\,\forall j\right)} \times \frac{\log\left(\max(rl(s_j)\,\forall j)\right)}{rl(s)} \quad (4)$$

iv. Score based on Word2Vec: Word2Vec (Levy et al., 2015) generates word embeddings using methods like skip-gram and continuous bag of words (CBOW). Here, the CBOW approach is employed with Gensim[3] to compute the Word2Vec score for a sentence s as follows:

1. Calculate the average length q of sentences in T (excluding stopwords):
2. Generate a query vector V_{query} using the q most frequent words from T.
3. Train a Word2Vec model on T, and obtain embedding vectors v_1 for sentence s and v_2 for V_{query}.

[1] https://radimrehurek.com/gensim/

[2] This threshold balances minimizing redundancy and maintaining diversity.

[3] https://radimrehurek.com/gensim/models/word2vec.html

4. Compute the Word2Vec score (s_{Word2Vec}) using Equation 6.

$$s_{\text{Word2Vec}} = \textit{cosine-similarity}(v_1, v_2) = \frac{v_1 \cdot v_2}{\|v_1\| \, \|v_2\|} \tag{6}$$

v. Score based on Doc2Vec: Similar to Word2Vec, Doc2Vec[4] generates document embeddings. The Doc2Vec score of s (Equation 7) is computed using vectors k_1 (Doc2Vec embedding s and k_2 (embedding of V_{query}):

$$s_{\text{Doc2Vec}} = \frac{k_1 \cdot k_2}{\|k_1\| \, \|k_2\|} \tag{7}$$

vi. Score based on LDA2Vec: LDA2Vec[5] enhances Word2Vec by combining word vectors with document vectors. A similar procedure to Word2Vec is applied to calculate the LDA2Vec score (Equation 8) using vectors t_1 (LDA2Vec embedding of s) and t_2 (embedding of V_{query})

$$s_{\text{LDA2Vec}} = \frac{t_1 \cdot t_2}{\|t_1\| \, \|t_2\|} \tag{8}$$

2.4 Calculation of unified score

To select candidate sentences that exhibit strong representativeness (cohesiveness), high diversity, minimal length, and significant semantic similarity (assessed using Word2Vec, Doc2Vec, and LDA2Vec scores), a unified score (unf) is calculated as presented in Equation 9. Based on the

$$\textit{unf}(s) = \textit{rpe}(s) + \textit{div}(s) + \textit{len}(s) + s_{\textit{Word2vec}} + s_{\textit{Doc2vec}} + s_{\textit{LDA2vec}} \tag{9}$$

unified score, the top $m\%$ of sentences from each topic are selected and stored in a list L_{new}, forming the initial summary.

2.5 Structuring sentences in the preliminary summary

The sentences in L_{new} are organized according to their importance using two approaches: a local score (based on entropy) and a global score (based on the silhouette coefficient). A combined score incorporating these measures ensures that sentences are prioritized effectively.

i. Entropy-Based Technique: This approach calculates a sentence's local importance within its document d using Kullback-Leibler Divergence (KLD) as shown in Equation 10.

$$\textit{KLD}(s, d) = \sum_{w} P(w \mid s) \log \left(\frac{P(w \mid d)}{P(w \mid s)} \right) \tag{10}$$

Here:

$$P(w \mid s) = \frac{\textit{term-frequency}(w, s)}{\|s\|} \tag{11}$$

$$P(w \mid d) = \frac{\textit{term-frequency}(w, d)}{\|d\|} \tag{12}$$

$$s_{\textit{Weight}} = \frac{1}{\text{KLD}(s, d)} \tag{13}$$

The weight of a sentence is inversely proportional to its KLD value as shown in Equation 14.

$$s_{\textit{Silhouette}} = \frac{\textit{separation} - \textit{cohesion}}{\max(\textit{cohesion}, \textit{separation})} \tag{14}$$

ii. Silhouette Coefficient: The silhouette coefficient assesses a sentence's global importance within its topic T. It is computed using Equation 15.

$$s_{\textit{total}} = s_{\textit{Weight}} + s_{\textit{Silhouette}} \tag{15}$$

Here, *cohesion* measures the similarity of s to the centroid of its topic T, and *separation* evaluates its dissimilarity to centroids of neighboring topics.

iii. Combined Score: The total importance score for a sentence is computed by combining the local (entropy-based) and global (silhouette-based) scores as shown below.

$$s_{\textit{total}} = s_{\textit{Weight}} + s_{\textit{Silhouette}}$$

Sentences in L_{new} are arranged in descending order of $s_{\textit{total}}$, producing the final summary referred to as the 'system-generated summary'.

2.6 Generation of extractive gold summaries

The process for generating extractive summaries from the DUC dataset, which includes four human-crafted gold summaries (C_{duc}), is described as follows:

[4] https://tedboy.github.io/nlps/generated/generated/gensim.models.Doc2Vec.score.html

[5] https://towardsdatascience.com/lda2vec-word-embeddings-in-topic-models-4ee3f0246243

i. Utilize the NLP Toolkit[6] to parse all sentences from each document $d \in C_{duc}$.

ii. Construct a list L comprising terms from the four gold summaries. For each sentence $s \in L_{\text{new}}$, compute its score based on the count of overlapping words with L.

iii. Rank the sentences in L_{new} based on their computed scores.

iv. The top m sentences[7] are extracted to form the extractive gold summary. Each document $d \in C_{duc}$ is associated with an extractive gold summary of ten sentences.

3 Evaluation of Experimental Findings

3.1 Corpus used

The Document Understanding Conference (DUC)[8] dataset was utilized for experimental analysis, with its detailed description provided in Table 1. The datasets for DUC-2006 and DUC-2007 are derived from the AQUAINT corpus, while those for DUC-2001 and DUC-2002 originate from TREC-9. A topic modeling approach was applied to each of these datasets, specifically using Latent Dirichlet Allocation (LDA). The optimal number of topics for each dataset, determined using the proposed heuristic method, is illustrated in Figures 1 to 4.

Table 1: Datasets Used

Dataset		Number of Sets	Number of Documents	Avg. Sentences per per Document	Summaries (Length)
DUC	2007	45	1123	37.59	200
	2006	50	1248	30.25	200
	2002	59	567	32.56	100
	2001	30	299	28.30	100

3.2 Discussion on performances using DUC-2002 and DUC-2006 datasets

The experimental performance of five traditional summarization methods was evaluated using the DUC-2002 dataset, as shown in Figure 5. The proposed model outperformed all baselines. Using the DUC-2006 dataset (1,250 documents), Figure 6 presents ROUGE scores comparing the proposed model with six baselines. The proposed model achieved the highest ROUGE-1 score and competitive ROUGE-2 scores, demonstrating superior content relevance and summarization quality.

[6] https://www.nltk.org/

[7] In this work, $m = 10$.

[8] http://www.duc.nist.gov

3.3 Analysis of performance metrics using statistical methods

Summary quality depends on sentence density, length, key terms, and linguistic components. Readability was assessed using the Coleman-Liau (CoL) index, Flesch-Kincaid Grade Level (FKG), and Automated Readability Index (ARIn) (Table 2). This method, tested on DUC2006 and DUC2002 datasets (Tables 3 and 4), outperformed existing techniques, improving readability and clarity.

Table 2: Readability Evaluation Techniques

Metric	Formula
CL	$6.88 \times \frac{\text{words}}{\text{characters}} - 0.34 \times \frac{\text{sentences}}{\text{words}} - 14.84$
FKG	$0.37 \times \frac{\text{sentences}}{\text{words}} + 10.7 \times \frac{\text{words}}{\text{syllables}} - 14.58$
ARIn	$5.70 \times \frac{\text{words}}{\text{sentences}} + 0.54 \times \frac{\text{sentences}}{\text{words}} - 20.48$

Table 3: Summary Readability(DUC-2002)

Model	CL	FKGL	ARI
URANK	74.2	85.4	71.3
TEXTRANK	71.5	80.3	**86.9**
TGRAPH	67.3	75.3	78.9
ILP	78.5	82.4	67.3
NN-SE	67.8	78.2	70.3
Proposed Model	**79.3**	**86.7**	84.3

Table 4: Summary Readability (DUC-2006)

Model	CL	FKGL	ARI
CTMSUM	73.80	75.30	77.60
IIITH-sum	68.90	73.20	75.80
TopicalN	81.40	79.9	71.60
OnModer	84.90	81.20	79.60
SFU_v36	78.90	87.10	80.70
RMSUM	83.67	80.40	85.80
Proposed Model	**85.37**	**85.35**	**89.30**

3.4 Analysis of performance metrics: Impact of including and excluding Stop Words

The effect of stop-words on text summarization was analyzed using the DUC-2002 dataset. Results (Tables 5 and 6) show that models like Sumbasic, KL-LDA, and Doc-LDA, which retain stop-words, perform better. This improvement, attributed to ROUGE metrics, highlights the importance of structural elements preserved by stop-words in enhancing ROUGE scores.

4 Conclusion and Future Work

This study proposes a method combining LDA-based topic modeling with a unified scoring system for extractive multi-document summarization. It ensures concise summaries with high representativeness, minimal redundancy, and strong semantic similarity, evaluated using ROUGE scores on DUC datasets. Experimental results show its superiority

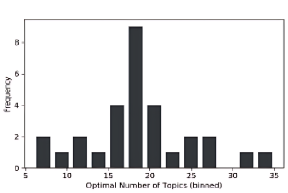

Figure 1: Required no. of topics (DUC-2007)

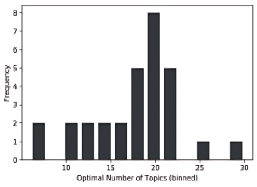

Figure 2: Required no. of topics (DUC-2006)

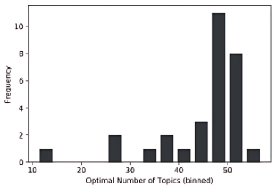

Figure 3: Required no. of topics (DUC-2002)

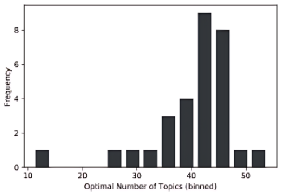

Figure 4: Required no. of topics (DUC-2001)

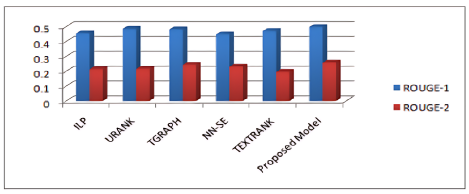

Figure 5: ROUGE score comparison (DUC-2002)

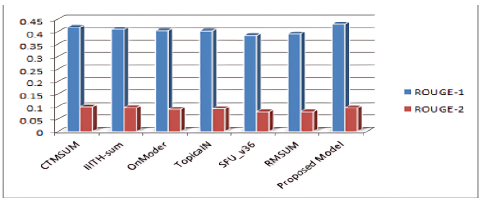

Figure 6: ROUGE score comparison (DUC-2006)

Table 5: Performances using stopwords

Algorithm	R-1	R-2	R-L	R-SU4
Doc-LDA	0.434	0.153	0.391	**0.194**
Sumbasic	0.409	0.098	0.375	0.149
KL-LDA	0.408	0.133	0.374	0.176
Proposed Approach	**0.465**	**0.165**	**0.399**	0.185

Table 6: Performances Without using stopwords

Algorithm	R-1	R-2	R-L	R-SU4
Sumbasic	0.309	0.074	0.296	0.102
KL-LDA	0.289	0.107	0.274	0.124
Doc-LDA	0.322	0.133	0.300	0.146
Proposed Approach	**0.396**	**0.224**	**0.333**	**0.193**

over existing methods. Future work may explore integrating abstractive summarization for enhanced coherence and meaning.

References

Zakariae Alami Merrouni, Bouchra Frikh, and Brahim Ouhbi. 2023. Exabsum: a new text summarization approach for generating extractive and abstractive summaries. *Journal of Big Data*, 10(1):163.

Shiwei Jiang, Qingxiao Zheng, Taiyong Li, and Shuanghong Luo. 2024. Clinical research text summarization method based on fusion of domain knowledge. *Journal of Biomedical Informatics*, page 104668.

Alka Khurana and Vasudha Bhatnagar. 2022. Investigating entropy for extractive document summarization. *Expert Systems with Applications*, 187:115820.

Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the association for computational linguistics*, 3:211–225.

Ruby Rani and DK Lobiyal. 2022. Document vector embedding based extractive text summarization system for hindi and english text. *Applied Intelligence*, 52(8):9353–9372.

Rajendra Kumar Roul. 2021. Topic modeling combined with classification technique for extractive multi-document text summarization. *Soft computing*, 25:1113–1127.

Rajendra Kumar Roul and Kushagr Arora. 2019. A nifty review to text summarization-based recommendation system for electronic products. *Soft Computing*, 23(24):13183–13204.

Rajendra Kumar Roul, Samarth Mehrotra, Yash Pungaliya, and Jajati Keshari Sahoo. 2019. A new automatic multi-document text summarization using topic modeling. In *Distributed Computing and Internet Technology: 15th International Conference, ICDCIT 2019, Bhubaneswar, India, January 10–13, 2019, Proceedings 15*, pages 212–221. Springer.

Hassan Shakil, Ahmad Farooq, and Jugal Kalita. 2024. Abstractive text summarization: State of the art, challenges, and improvements. *Neurocomputing*, page 128255.

Kamonwan Tawong, Pichayapas Pholsukkarn, Pakanun Noawaroongroj, and Thitirat Siriborvornratanakul. 2024. Economic news using lstm and gru models for text summarization in deep learning. *Journal of Data, Information and Management*, 6(1):29–39.

Pradeepika Verma, Anshul Verma, and Sukomal Pal. 2022. An approach for extractive text summarization using fuzzy evolutionary and clustering algorithms. *Applied Soft Computing*, 120:108670.