Wikipedia Current Events Summarization using Particle Swarm Optimization

Santosh Kumar Mishra [1], Darsh Kaushik [2], Sriparna Saha [1], and Pushpak Bhattacharyya [3]
[1] Department of Computer Science & Engineering, Indian Institute of Technology Patna, India
[2] Department of Computer Science & Engineering, National Institute of Technology Silchar, India
[3] Department of Computer Science & Engineering, Indian Institute of Technology Bombay, India
{santosh_1821cs03, sriparna}@iitp.ac.in [1], darsh_ug@cse.nits.ac.in [2], pb@cse.iitb.ac.in [3]

Abstract

This paper proposes a method to summarize news events from multiple sources. We pose event summarization as a clustering-based optimization problem and solve it using particle swarm optimization. The proposed methodology uses the search capability of particle swarm optimization, detecting the number of clusters automatically. Experiments are conducted with the Wikipedia Current Events Portal dataset and evaluated using the well-known ROUGE-1, ROUGE-2, and ROUGE-L scores. The obtained results show the efficacy of the proposed methodology over the state-of-the-art methods. It attained improvements of 33.42%, 81.75%, and 57.58% in terms of ROUGE-1, ROUGE-2, and ROUGE-L, respectively.

1 Introduction

The continuously rising amount of text data makes analyzing and comprehending textual files tiresome as technology progresses in a fast-changing fashion. Capturing important information from large documents is a time-consuming and labor-intensive job from the reader's perspective. A large number of documents must be handled quickly, and a large amount of text data necessitates the use of text and document summarization algorithms. However, the focus has been on single-document summarization both for extractive and abstractive variants with comparatively little advancements in multi-document summarization.

Multi-document summarization techniques are becoming paramount in recent years. There are several real life applications of multi document summarization like : scientific summarization (Yasunaga et al., 2019) (Mishra et al., 2021d) (Mishra et al., 2020), news summarization (Fabbri et al., 2019), email thread summarization (Zhang et al., 2021), summarization of product reviews (Gerani et al., 2014), course feedback summarization (Luo et al., 2016), Wikipedia article generation (Liu et al., 2018), summarization of medical documents (Afantenos et al., 2005).

Deep learning has gained a huge amount of attention in recent years as a result of its success in computer vision (Krizhevsky et al., 2012), natural language processing (Devlin et al., 2014) (Mishra et al., 2020), and multi-modal applications (Wang et al., 2020) (Mishra et al., 2021b) (Mishra et al., 2021a). Researchers use deep learning to solve challenging problems because of its capacity to capture highly nonlinear data relationships. Deep learning-based models have recently been used in multi-document summarization (Zhang et al., 2021) (Fabbri et al., 2019) (Yasunaga et al., 2017), advancing the field of text summarization and allowing models to improve their performances. Attempts to utilise big deep learning models, which have considerably improved the state-of-the-art for different supervised natural language processing tasks, however, are hampered by a shortage of large datasets, making a comprehensive evaluation impossible. Even with larger datasets, compute resources and the corresponding training time might also pose a challenge in case of MDS (multi-document summarization) since several documents have to be processed. Moreover, for single and multi-document summarization, meta-heuristics algorithms have shown good results in previous studies (Mishra et al., 2021d) (Saini et al., 2019a) (Saini et al., 2019b) (Mishra et al., 2021c).

In this work, we have proposed a meta-heuristic optimization technique-based multi-document summarization methodology using Wikipedia Current Events Portal (WCEP) dataset introduced in Association for Computational Linguistics (ACL), 2020 (Ghalandari et al., 2020). Major contributions of this work are as follows:

- Employment of word mover's distance (WMD) (Kusner et al., 2015) to find the doc-

Proceedings of the 18th International Conference on Natural Language Processing, pages 447–455
Silchar, India. December 16 - 19, 2021.

ument center, it captures the semantic similarity. The proposed approach(s) utilize the word move distance (WMD) to capture the semantic similarity of sentences. It's worth noting that WMD doesn't require sentences to be represented as vectors. It employs word embeddings for various terms derived from a word2vec model trained on the Google news corpus, which comprises of 3 billion words and each word vector has 300 dimensions. If two phrases are similar, the WMD for each will be 0.

- We have used the particle swarm optimization-based clustering (PSO) (Kennedy and Eberhart, 1995) technique to cluster these news event sentences efficiently. It decides the number of clusters automatically within documents. This is the first effort to summarize Wikipedia's current event documents using PSO-based clustering to the best of our knowledge.
- To generate the summary, meaningful sentences from different clusters are selected using sentence scoring features like sentence's position, sentence length, similarity with paper's title, and similarity with the document center.

2 Related Work

To solve multi-document summarization, non-neural and neural network-based methods have been used in the literature.

Non-neural approaches have been widely used in the literature for multi-document summarization. In (Carbonell and Goldstein, 1998), authors have used query relevance and maximum marginal relevance to accomplish text summarization. They utilized the maximum marginal relevance to maintain anti-redundancy in generated summary. Authors of (Radev et al., 2004) proposed a clustering-based approach in which summary is generated using cluster centroid. Apart from this, they proposed evaluation techniques using subsumption and sentence utility for single and multi-document summarization. In (Erkan and Radev, 2004), an unsupervised graphical method, LexRank, has been proposed for text summarization. Here, the proposed method accomplishes sentence scoring using the graph-based method. LexRank finds the important sentences utilizing eigenvector centrality of graph representation denoting sentences. Authors of (Mihalcea and Tarau, 2004) have proposed the TextRank method for text summarization; this is based on page ranking methodology. In (Haghighi and Vanderwende, 2009), a generative probabilistic methodology to summarize multiple documents is proposed. Here, the authors have proposed a way of constructing a sequence of models using a frequency-based model. In (Radev and McKeown, 1998), authors have developed a method 'SUMMONS' that combines information from various news articles and converts it into a summary. In (Barzilay et al., 1999), multi-document summarization is accomplished by finding similar elements across texts from different documents. A graph-based summarization technique, namely 'Opinosis' introduced in (Ganesan et al., 2010), generates a precise abstractive summary from the redundant opinion. A word-level and sentence-level ranking based on various indicators of importance, keyword extraction, and phrase-level salience (Hong, 2005) (Cao et al., 2015), greedy heuristics on relation graphs and embedding (Yasunaga et al., 2017) has been used to solve the multi-document summarization.

Nowadays, supervised learning is used to solve extractive and abstractive summarization problems. But, limitation of supervised approaches is that it requires a huge amount of data for training. An attention with encoder-decoder based recurrent neural network is introduced in (Nallapati et al., 2016a). Here, authors have carried out abstractive summarization over DUC and CNN/Daily mail datasets. In (Cheng and Lapata, 2016), authors have proposed a data-driven approach with a deep neural network that incorporates the continuous sentence features. They developed an architecture consisting of a hierarchical document encoder and an attention-based extractor. A sentence ranking-based approach for single document summarization is explored in (Narayan et al., 2018). Here, authors have proposed a training methodology to optimize the ROUGE evaluation metric using reinforcement learning. A conditional recurrent neural network has been employed for abstractive summarization in (Chopra et al., 2016). Here, the convolutional attention-based encoder ensures the conditioning of the input sequence that helps the decoder to focus on relevant input words at each time step of the summary generation. In (Nallapati et al., 2016b), an extractive summarization of the document has been carried out using contrasting recurrent neural network-based architecture. The proposed method classifies the sentences in a sequential way that decides whether a sentence should be accepted or rejected to be included in the summary. Further, a sentence selector selects a single sentence at a time in random order to form the summary. A sequence to sequence architecture for abstractive summarization has been introduced in (See et al., 2017). Authors have proposed a pointer generator-based sequence to sequence model that can copy a word from the source text that helps in generating an accurate summary. In (Paulus et al., 2017), an intra-attention mechanism has been introduced that attends the input sequence and generates the output separately in a continuous manner. The authors have also proposed a new training methodology that utilizes reinforcement learning and supervised word prediction. Standard word prediction is coupled with RL's global sequence prediction training, resulting in more comprehensible summaries. Author of (Cohan et al., 2018) proposed a architecture to learn discourse structure of the documents. Apart from these, they also employed an attentive discourse-aware decoder that can summarize single and multiple documents. In (Celikyilmaz et al., 2018), abstractive summarization has been accomplished using deep communicating agents in the encoder-decoder model. Here, the deep communicating agents divide the long documents into smaller parts and assign them to different collaborative agents. The collaborative agents work as agents connected through a single decoder which trains end-to-end using reinforcement learning to generate a coherent and accurate summary. In (Gehrmann et al., 2018), authors have introduced a data-efficient content selector that finds the phrase in the input document that is important for the summary. This selector is employed as bottom-up attention to constraining the model to similar phrases.

The limitation of the supervised approach (deep learning model) is that it needs a huge amount of data for learning. We often don't have enough data to train a supervised model in many instances. Motivated by this, we present an unsupervised method for summarizing events in an extractive way from recent news, which we evaluate on the WCEP dataset (Ghalandari et al., 2020). It contains daily news events and their corresponding summaries. The proposed approach does not require massive data, and it has consistent performance irrespective of dataset size.

3 Proposed Methodology

This section has an overview of the proposed methodology. Fig 1 and Algorithm 1 illustrate the steps and pseudo-code, respectively. The notations used in this section are defined in Table 1.

The proposed methodology is based on a natural phenomenon; at the end of its execution, it generates a set of solutions. We get a set of optimal solutions at the end. Here, a solution is made up of a group of sentence clusters that have been optimized (particle).

Particle swarm optimization (Kennedy and Eberhart, 1995) is a famous nature-inspired method that was designed inspired by the social behavior of bird flocks. It's a population-based method of searching. The method maintains a population of particles. Every particle in this diagram represents a viable optimization solution. A swarm comprises numerous alternative solutions to an optimization issue known as particles in the PSO framework. The PSO algorithm's goal, in this case, is to find the optimal particle position that produces the best fitness value in terms of the objective function. We used a PSO-based clustering approach with K-means clustering to seed the original swarm. It entails the following procedures:

1. Particle representation: Each particle chooses K different sentence vectors as initial cluster centroid vectors in the first step.

2.
   - Points are assigned to various clusters as follows: Each phrase vector is allocated to the centroid vector that is closest to it, and then the fitness value is calculated using Equation 5.
   - Updation of position and velocity: In order to create the new solution, the particle's velocity and position are changed using Equations 1 and 2.

3. Step 2 should be repeated until the termination condition is met:
   - The total number of iterations has been achieved.
   - There is a little change in the centroid vector.

Each particle in N_d dimensional space represents a position, and it moves throughout multi-dimensional search space, changing its location in reference to both:

- Particle's best position found.
- Best position in the neighborhood of that particle.

The following information is maintained by every particle:

- y_i The particle's personal best position.
- x_i: The particle's current position.
- v_i The particle's current velocity.

Using the notations above, the particle's position is modified according to

$$\begin{aligned} v_{i,k}(t+1) &= wv_{i,k}(t) + c_1 r_{1,k}(t)(y_{i,k}(t) \\ &-x_{i,k}(t)) + c_2 r_{2,k}(t)(\hat{y_k}(t) - x_{i,k}(t)) \end{aligned} \tag{1}$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{2}$$

Here, c_1 and c_2 is denotes the acceleration constant, inertia weight is w, $r_{1,j}(t)$, $r_{2,j}(t)$ denote the random number between 0 and 1 where $k = 1, ..., N_d$. Velocity is computed using three components: (1) component denoting function of particle's distance from the personal best position, (2) fraction of the previous velocity, (3) social component representing the distance between particle and best particle.

The particle's personal best position is measured as follows:

$$\begin{aligned} y_i(t+1) = y_i(t) \;\; if \;\; f(x_i(t+1)) \geq f(y_i(t)) \\ x_i(t+1) \;\; if \;\; f(x_i(t+1)) < f(y_i(t)) \end{aligned} \tag{3}$$

Equation 1 denotes the global best version of PSO, where at end global best solution is taken into consideration, where i^{th} particle's neighborhood has best particle $\hat{y_k}$.

A single particle describes the N_c centroid vectors in the sense of clustering. Here, every particle x_i is formed, as follows:

$$x_i = \{m_{i1}, m_{i2}, ...m_{ij}, ...m_{iN_c}\} \tag{4}$$

Here, m_{ij} corresponds to the centroid vector of the j^{th} cluster of the i^{th} particle of the C_{ij} cluster; therefore, for the existing data vectors, a swarm describes a set of candidate clusters. As a quantization error, the fitness of the particle is calculated as follows:

$$E = \frac{\sum_{j=1}^{N_c} [\sum_{\forall Z_p \in C_{ij}} d(z_p, m_j)/|C_{ij}|]}{N_c} \tag{5}$$

PSO begins with a swarm, which is a collection of possible solutions (particles). The particle X_i is made up of solutions $\{m_{i1}, m_{i2}, ...m_{ij}, ...m_{iN_c}\}$ with varying numbers of clusters. Our assumption is the solution, m^{ij}, would represent the centers of the sentence clusters. However, this is a challenging task to identify the number of clusters in a document automatically. Because of this complexity, each solution has a different number of clusters, ranging from $1 \leq K \leq M$. M signifies the number of sentences to be clustered, using K number of clusters.

K-mean algorithm, with the current number of cluster centers, is invoked for each solution. After each iteration of the K-means algorithm, cluster centroids/centers are modified, and this step is repeated until the centroids are converged. particles change the velocity and position to obtain the best fitness values. In the end, it automatically decides the number of clusters as the algorithm terminates.

3.1 Summary Generation

The summary generation procedure is as follows:

- **Document's center identification:** The sentence with the lowest average WMD distance is considered as the document center with respect to all other sentences. The M number of sentences are taken into account to determine the average WMD for a sentence.

$$t = argmin_i \sum_{i=1}^{M} \sum_{j=1, i \neq j}^{M} \frac{wmd(s_i, s_j)}{A} \tag{6}$$

Where, representative sentence or document centre is t, M denotes the number of news sentences , s_i, s_j denote document's i^{th} and j^{th} sentence, respectively, A represents the number of sentence pairs.

- **Cluster's ranking in i^{th} particle:** The Cosine similarity computed between the cluster centers and the document center are used to rank clusters inside a particle in decreasing order. In other words, cluster with the shortest distance to the document center will be given higher priority(higher rank) than others.

In order to generate the summary, sentences belonging to different clusters (high to low) must be extracted as per their ranks.

Sentence scores are therefore assessed in each cluster on the basis of four features. Those are the similarity of sentence to the paper's title, length of the sentence, the position of the sentence, sentences close to the document center, Descriptions of these features are given below:

- Similarity with the paper's title (F1): The sentences which are semantically close to the document's title have given high scores (Saini et al., 2019a). Firstly, these sentences are considered summary generation. This is defined as follows:

$$F1 = wmd(s_i^k, title) \tag{7}$$

where, s_i^k represents i^{th} sentence of the k^{th} cluster, document's title is represented by $title$ and $dist_{wmd}$ is WMD between sentence and document's title.

- Position of the sentence (F2:) Essential sentences can be found at the start of most paragraphs/documents. These sentences can be helpful to generate a good quality summary (Saini et al., 2019a).

$$F2 = \frac{1}{\sqrt{r}} \tag{8}$$

- Length of sentence (F3:) The length of sentence has been used as a selecting criteria. Here the sentence which are longer in the length given higher priority over others (Saini et al., 2019b) (Mishra et al., 2021d).

- Sentences close to the document center in terms of WMD (F4): Sentences in each cluster identical to the document center in terms of WMD have been included first in summary (Saini et al., 2019b) (Saini et al., 2019a).

Symbol	Meaning
N_c	Number of cluster centroids
x_i	Particle's current position
t	Time steps
v_i	Particle's current velocity
N_c	Cluster centroid vector
y_i	Particle's best position
N_d	Input dimension
$v_{i,k}(t+1)$	Updated velocity in k dimension
w	weight of inertia
r	Random number between 0 and 1
$\hat{y}$	Best particle
m	Centroid Vector
c_1, c_2	Acceleration constant
WMD	Word mover's distance

Table 1: List of abbreviations

Algorithm 1 WCEP_EventSumm-PSO

1: **Input:** News event from Wikipedia Current Event Portal
2: **Output:** Summary of the news events
3: Initialize each particle with N_c randomly selected centroids.
4: **for** $i \leftarrow 1$ to t_{max} **do**
5:     **for** each particle i **do**
6:         **for** each data vector z_p **do**
7:             Find the Euclidean distances $d(z_p, m_{ij})$ to all cluster centroids C_{ij}
8:             z_p assigned to cluster C_{ij} such that $d(z_p, m_{ij}) = min_{\forall c=1,\ldots\ldots,N_c}\{d(z_p, m_{ic})\}$
9:             Calculate the fitness using Equation 5
10:     Global and local best positions are being updated.
11:     Update the centroids of the clusters using Equations 1 and 2.
12:     Summary generation corresponding to Global best solutions as discussed in section 3.1

Methods	ROUGE-1	ROUGE-2	ROUGE-L
Similarity with the title (F1)	0.459	0.239	0.395
Position of the sentence (F2)	0.471	0.249	0.405
Length of the sentence (F3)	0.425	0.203	0.355
Similarity with the document center (F4)	0.442	0.221	0.376

Table 2: Score obtained with different features

4 Experimental Setup

This section has a detailed discussion on dataset and evaluation metrics used.

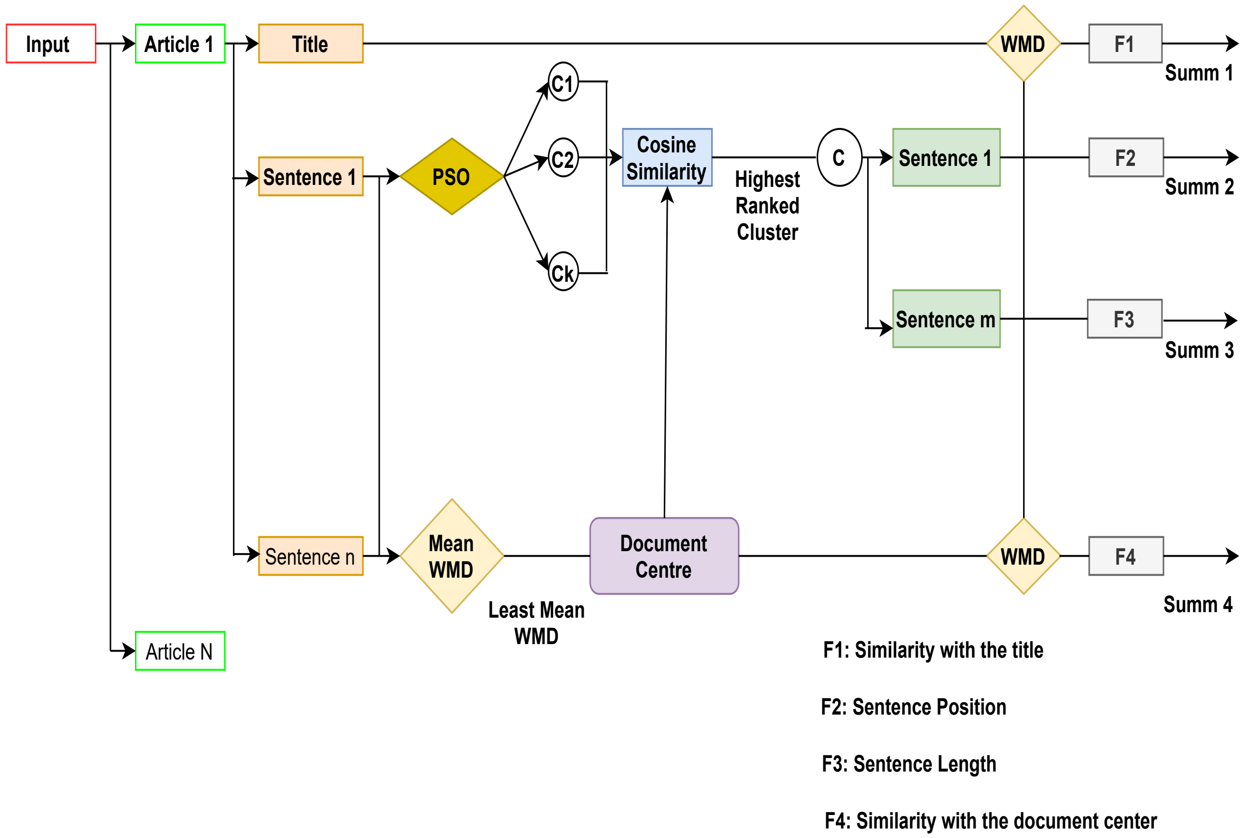

Figure 1: The process flow chart of the proposed method.

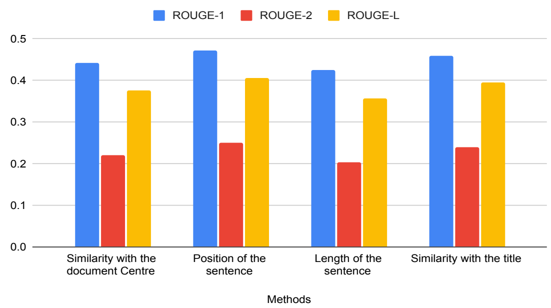

Figure 2: The bar graph of obtained score with the different features

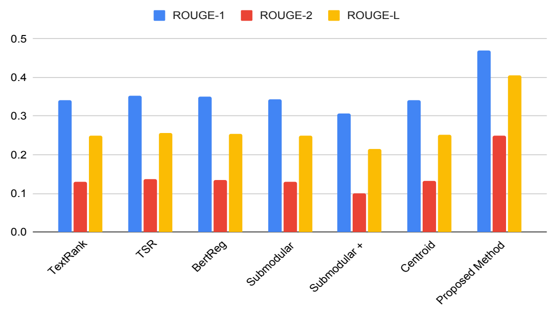

Figure 3: The bar graph showing comparison with the state-of-the-art methods

4.1 Dataset

We use WCEP (Ghalandari et al., 2020) dataset for the experimentation. The dataset contains 10,200 items from recent news events, as well as their summaries. Train set, validation set and test set consist of 8158, 1020 and 1022 respectively.

4.2 Evaluation metrics

The proposed approach is evaluated using the popular evaluation metrics ROUGE scores (Lin, 2004) used for text and document summarization. This score computes the overlapping n-grams between the generated summary and the ground truth summary. F1-score, precision, and recall are commonly utilized in the literature to do quantitative analysis. The Rouge-F1 scores are shown in the Tables 2 and Table 3 .

5 Result and Discussions

This section has a detailed discussion on results obtained and their analysis. We have shown the obtained score in Table 2 and comparison with the

Methods	ROUGE-1	ROUGE-2	ROUGE-L
TextRank	0.341	0.131	0.25
TSR	0.353	0.137	0.257
BertReg	0.35	0.135	0.255
Submodular	0.344	0.131	0.25
Submodular + Abs	0.306	0.101	0.214
Centroid	0.341	0.133	0.251
Proposed Method	**0.471**	**0.249**	**0.405**

Table 3: Comparison of the proposed method with the state-of-the-art methods

state-of-the-art methods in Table 3.

5.1 State-of-the-art comparative baselines

We have accomplished the comparison with the following state-of-the-art methods:

- *TextRank:* This is an unsupervised method of text summarization (Mihalcea and Tarau, 2004). It is based on a graph-based ranking model that perceives the most important sentence and the keyword for the summary.
- *Centroid:* This methodology generates the summary utilizing the cluster centroid generated by a topic detection algorithm (Radev et al., 2004).
- *TSR:* This approach is based on sentence ranking based on statistical feature an average of the word embedding vectors (Ren et al., 2016).
- *BERTREG:* This is similar to TSR methodology, but it uses the sentence embedding produced by pre-trained BERT (Devlin et al., 2019).
- *SUBMODULAR:* This method is based on the submodular function that integrates coverage and non-redundancy to find the important sentence within the document to form the summary (Chali et al., 2017).
- *SUBMODULAR + Abs:* Abstractive based approach sentence compression and metging is incorporated in SUBMODULAR approach (Chali et al., 2017).

5.2 Analysis of the Results:

We have shown the obtained score with different features in Table 2 and comparison with state-of-the-art methods in 3. It can be concluded from Table 3 that the proposed methodology outperforms the state-of-the-method. It can be seen from Table 3 that TSR (Ren et al., 2016) has the highest score among all methods. The bar graph of scores obtained with different features and comparison with the state-of-the-art is shown in Fig 2 and Fig 3 respectively. If, we compare with the TSR method, the proposed method has the improvement of 33.42%, 81.75%, and 57.58% considering ROUGE-1, ROUGE-2, and ROUGE-L, respectively.

6 Conclusion and Future Works

This paper presents a method of Wikipedia current event summarization using a particle swarm optimization-based clustering methodology. We utilized the search capability of particle swarm optimization as an underlying optimization strategy, an evolutionary algorithm. The proposed method detects the number of clusters automatically. The different feature has been employed for sentence scoring within-cluster and to form the final summary. The efficacy of the proposed method has been tested on the WCEP dataset. The obtained results show the effectiveness of the proposed method over state-of-the-art methods. Compared to the best method among the state-of-the-art, the proposed method has the improvement of 33.42%, 81.75%, and 57.58% in terms of ROUGE-1, ROUGE-2, and ROUGE-L, respectively.

In the future, this work can be extended using the ensembling of the clustering technique. Apart from that, more sophisticated feature word mover's distance, BERT similarity, and textual entailment can be utilized for a summary generation.

Acknowledgments

Dr. Sriparna Saha gratefully acknowledges the Young Faculty Research Fellowship (YFRF) Award, supported by Visvesvaraya Ph.D. Scheme for Electronics and IT, Ministry of Electronics and Information Technology (MeitY), Government of India, being implemented by Digital India Corporation (formerly Media Lab Asia) for carrying out this research.

References

Stergos Afantenos, Vangelis Karkaletsis, and Panagiotis Stamatopoulos. 2005. Summarization from medical documents: a survey. *Artificial intelligence in medicine*, 33(2):157–177.

Regina Barzilay, Kathleen McKeown, and Michael Elhadad. 1999. Information fusion in the context of multi-document summarization. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics*, pages 550–557.

Ziqiang Cao, Furu Wei, Li Dong, Sujian Li, and Ming Zhou. 2015. Ranking with recursive neural networks and its application to multi-document summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29.

Jaime Carbonell and Jade Goldstein. 1998. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336.

Asli Celikyilmaz, Antoine Bosselut, Xiaodong He, and Yejin Choi. 2018. Deep communicating agents for abstractive summarization. *arXiv preprint arXiv:1803.10357*.

Yllias Chali, Moin Tanvee, and Mir Tafseer Nayeem. 2017. Towards abstractive multi-document summarization using submodular function-based framework, sentence compression and merging. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 418–424.

Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. *arXiv preprint arXiv:1603.07252*.

Sumit Chopra, Michael Auli, and Alexander M Rush. 2016. Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 93–98.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. *arXiv preprint arXiv:1804.05685*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard Schwartz, and John Makhoul. 2014. Fast and robust neural network joint models for statistical machine translation. In *proceedings of the 52nd annual meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1370–1380.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479.

Alexander R Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir R Radev. 2019. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. *arXiv preprint arXiv:1906.01749*.

Kavita Ganesan, ChengXiang Zhai, and Jiawei Han. 2010. Opinosis: A graph based approach to abstractive summarization of highly redundant opinions.

Sebastian Gehrmann, Yuntian Deng, and Alexander M Rush. 2018. Bottom-up abstractive summarization. *arXiv preprint arXiv:1808.10792*.

Shima Gerani, Yashar Mehdad, Giuseppe Carenini, Raymond Ng, and Bita Nejat. 2014. Abstractive summarization of product reviews using discourse structure. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1602–1613.

Demian Gholipour Ghalandari, Chris Hokamp, Nghia The Pham, John Glover, and Georgiana Ifrim. 2020. A large-scale multi-document summarization dataset from the wikipedia current events portal. *arXiv preprint arXiv:2005.10070*.

Aria Haghighi and Lucy Vanderwende. 2009. Exploring content models for multi-document summarization. In *Proceedings of human language technologies: The 2009 annual conference of the North American Chapter of the Association for Computational Linguistics*, pages 362–370.

Gumwon Hong. 2005. Relation extraction using support vector machine. In *Second International Joint Conference on Natural Language Processing: Full Papers*.

James Kennedy and Russell Eberhart. 1995. Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks*, volume 4, pages 1942–1948. IEEE.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105.

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From word embeddings to document distances. In *International conference on machine learning*, pages 957–966. PMLR.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Peter J Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. 2018. Generating wikipedia by summarizing long sequences. *arXiv preprint arXiv:1801.10198*.

Wencan Luo, Fei Liu, Zitao Liu, and Diane Litman. 2016. Automatic summarization of student course feedback. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 80–85.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411.

Santosh Kumar Mishra, Rijul Dhir, Sriparna Saha, and Pushpak Bhattacharyya. 2021a. A hindi image caption generation framework using deep learning. *Transactions on Asian and Low-Resource Language Information Processing*, 20(2):1–19.

Santosh Kumar Mishra, Rijul Dhir, Sriparna Saha, Pushpak Bhattacharyya, and Amit Kumar Singh. 2021b. Image captioning in hindi language using transformer networks. *Computers & Electrical Engineering*, 92:107114.

Santosh Kumar Mishra, Harshavardhan Kundarapu, Naveen Saini, Sriparna Saha, and Pushpak Bhattacharyya. 2020. Iitp-ai-nlp-ml@ cl-scisumm 2020, cl-laysumm 2020, longsumm 2020. In *Proceedings of the First Workshop on Scholarly Document Processing*, pages 270–276.

Santosh Kumar Mishra, Naveen Saini, Sriparna Saha, and Pushpak Bhattacharyya. 2021c. Let's summarize scientific documents! a clustering-based approach via citation context. In *International Conference on Applications of Natural Language to Information Systems*, pages 330–339. Springer.

Santosh Kumar Mishra, Naveen Saini, Sriparna Saha, and Pushpak Bhattacharyya. 2021d. Scientific document summarization in multi-objective clustering framework. *Applied Intelligence*, pages 1–24.

Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. 2016a. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*.

Ramesh Nallapati, Bowen Zhou, and Mingbo Ma. 2016b. Classify or select: Neural architectures for extractive document summarization. *arXiv preprint arXiv:1611.04244*.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Ranking sentences for extractive summarization with reinforcement learning. *arXiv preprint arXiv:1802.08636*.

Romain Paulus, Caiming Xiong, and Richard Socher. 2017. A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304*.

Dragomir Radev and Kathleen McKeown. 1998. Generating natural language summaries from multiple online sources. *Computational Linguistics*, 24(3):469–500.

Dragomir R Radev, Hongyan Jing, Małgorzata Styś, and Daniel Tam. 2004. Centroid-based summarization of multiple documents. *Information Processing & Management*, 40(6):919–938.

Pengjie Ren, Furu Wei, Zhumin Chen, Jun Ma, and Ming Zhou. 2016. A redundancy-aware sentence regression framework for extractive summarization. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 33–43.

Naveen Saini, Sriparna Saha, Anubhav Jangra, and Pushpak Bhattacharyya. 2019a. Extractive single document summarization using multi-objective optimization: Exploring self-organized differential evolution, grey wolf optimizer and water cycle algorithm. *Knowledge-Based Systems*, 164:45–67.

Naveen Saini, Sriparna Saha, Anurag Kumar, and Pushpak Bhattacharyya. 2019b. Multi-document summarization using adaptive composite differential evolution. In *International Conference on Neural Information Processing*, pages 670–678. Springer.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.

Hu Wang, Qi Wu, and Chunhua Shen. 2020. Soft expert reward learning for vision-and-language navigation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX 16*, pages 126–141. Springer.

Michihiro Yasunaga, Jungo Kasai, Rui Zhang, Alexander R Fabbri, Irene Li, Dan Friedman, and Dragomir R Radev. 2019. Scisummnet: A large annotated corpus and content-impact models for scientific paper summarization with citation networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7386–7393.

Michihiro Yasunaga, Rui Zhang, Kshitijh Meelu, Ayush Pareek, Krishnan Srinivasan, and Dragomir Radev. 2017. Graph-based neural multi-document summarization. *arXiv preprint arXiv:1706.06681*.

Shiyue Zhang, Asli Celikyilmaz, Jianfeng Gao, and Mohit Bansal. 2021. Emailsum: Abstractive email thread summarization. *arXiv preprint arXiv:2107.14691*.