IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-7, Issue-5, May 2019 E-ISSN: 2347-2693

Towards Better Single Document Summarization using Multi-Document Summarization Approach

Sandhya Singh[1*], Kevin Patel[2], Krishnanjan Bhattacharjee[3], Hemant Darbari[4], Seema Verma[5]

[1,5] Banasthali University, Rajasthan, India
[2] CFILT, IIT Bombay, Mumbai, India
[3,4] AAI, C-DAC, Pune, India

[*]*Corresponding Author: sandhya.singh@gmail.com*

DOI: https://doi.org/10.26438/ijcse/v7i5.695703 | Available online at: www.ijcseonline.org

Accepted: 26/May/2019, Published: 31/May/2019

Abstract— Extractive Single Document Summarization (SDS) is the task of summarizing a single document via extracting importance sentences verbatim and arranging them in a cohesive manner. It is different from Multi-Document Summarization (MDS) where multiple source documents are processed to generate a single summary. This paper proposes a two-stage mechanism to perform single document summarization via multi-document summarization technique. The approach involves the use of popular extractive summarization algorithms to generate summaries which are then further processed as multi-document summarization instance. The MDS approach used is based on word graph based sentence fusion followed by concept-based Integer Linear Programming (ILP) method for maximizing the coverage in sentence selection. The proposed system outperforms each of the single document summarizers by at least 2.6 percent point ROUGE scores, thereby indicating that performing single document summarization via multi-document summarization is a promising venue for further research in summarization.

Keywords— Text Summarization, Single Document Summarization (SDS), Multi-Document Summarization (MDS), Extractive Summarization, Integer Linear Programming (ILP)

I. Introduction

Automatic Text Summarization (ATS) involves automatically condensing the text document(s) while retaining the crux. In this age of *information overload*, automatic text summarization is a vital requirement for consuming information. The challenge comes with an increase in the number of documents for the same topic and the length of the document for creating a summary. Thus, this has been an active area of research in the field of Natural Language Processing (NLP) [1, 2, 3].

Based on the number of documents fed as an input to the system, ATS approaches can be classified into single document summarization (SDS) and multi-document summarization (MDS). In single document summarization, the output summary is created from a single source document. The summarization system relies on a cohesive piece of text with very little repetition of facts. In multi-document summarization, more than one source document is used for summarization. Multiple documents lead to an increase in redundancy which needs to be minimized by the summarization system while maximizing the important information.

Based on the type of output, ATS approaches can be classified into extractive summarization and abstractive summarization. In extractive summarization, the important sentences are taken verbatim from the source document and arranged in a cohesive manner. In abstractive summarization, important concepts from the source document are understood and re-phrased using natural language. It requires deep linguistic knowledge to generate grammatically correct language constructs while reconstructing the sentences. Abstractive summarization approach correlates to the humans approach of summarizing the articles.

Besides these, there are other variations of summarization based on the user need like indicative summarization [4, 5], informative summarization [6, 7], query-based summarization [8] *etc.*

In this work, the focus is on extractive single document summarization. The main challenges in extractive summarization are:

1) Finding the most salient sentences

2) Minimizing the entity references not mentioned in the selected sentences

3) Arranging the chosen sentences in a cohesive manner.

These challenges are further accentuated in case of extractive multi-document summarization, where the salient sentences may come from multiple sources, thereby making reference resolving and sentence ordering more challenging. It is intuited that an extractive MDS system is bound to work well on the SDS case, at least for the second and third challenge.

Through this paper, the following question is raised:

"Can extractive SDS be improved using extractive MDS?"

Many extractive SDS systems have been proposed over the years. They propose different mechanisms for salient sentence detection. However, none of them is a complete mechanism by themselves.

So this paper investigates the above question in the following manner: an extractive SDS is performed using MDS in a cascading manner. The approach involves passing the source document through a set of SDS algorithms as stage 1, which will reveal different sets of salient sentences. Then the original source document, along with the outputs of these SDS algorithms is given as input to a MDS algorithm as stage 2, which will generate the final output summary. As a result, the final summary will have sentences whose saliency has been determined via multiple SDS approaches, and that the sentence ordering will be taken care of by MDS algorithm. Thereby, it is expected that the final MDS generated summary will be better as compared to individual SDS summaries. Our preliminary investigation reveals that this is indeed the case, with the MDS summary having an average of 2.6 percent point better ROUGE score as compared to the SDS summaries. This work justifies further investigating SDS through MDS approaches.

The rest of the paper is organized as follows: section II discusses some of the approaches used for single document summarization, multi-document summarization and some ensemble approaches experimented in the past for summarization. Section III describes the experimental setup including the data used and the method followed for the experiment. Section IV discusses the result and its analysis followed by conclusion in section V.

II. RELATED WORK

Automatic text summarization has been actively pursued over the years, with the main focus on detecting important sentences. Some relevant works have been discussed here to understand the approaches experimented so far.

For extractive single document summarization, Silber and McCoy experimented with creating lexical chains and scoring them to select the sentences for extractive text summarization [9, 10]. Gong and Liu used information retrieval method to rank the relevance of sentences followed by latent semantic analysis to select the important sentences for extractive summary creation [11, 12]. Similarly, Harabagiu and Lacatusu developed a framework for a single document and multi-document summaries based on information Extraction techniques [13]. Erkan and Radev introduced a probabilistic graph-based ranking method known as LexRank for finding the important sentence from the document [14]. Sornil and Gree-Ut combined content-based and graph-based techniques for sentence extraction using Hopfield Network algorithm for ranking the text segments [15, 16]. Smith et al. experimented with coreference links to create more cohesive summaries [17]. The rank of the sentences is based on the number of out-links and in-links of coreference. Cheng and Lapata presented a neural framework using hierarchical document encoder and an attention-based extractor [18]. Nallapati et al. also proposed SummaRuNNer, a Recurrent Neural Network (RNN) based model for extractive summarization [19]. Garcia et al. experimented with coherent extractive single document summaries using Integer Linear Programming (ILP) approach with competitive results [40].

For extractive multi-document summarization, Radev et al. developed a comprehensive framework for summarization which combined various summarization algorithms like position based, centroid-based, largest common subsequence *etc.* [20]. Celikyilmaz and Hakkani-Tur proposed a hybrid model for multi-document summarization [21]. The hybrid model combined a generative model for pattern discovery and a regression model for inference. Fattah also suggested a machine learning based hybrid model for sentence extraction [22]. Cao et al. proposed a sentence-ranking framework based on a recurrent neural network model [23]. Hirao et al. used Integer Linear Programming (ILP) formulation to obtain a compressive summary for a multi-document scenario with promising results [24]. Gillick and Favre used ILP for creating a maximum coverage model by minimizing the redundancy and compressing sentences based on parse trees for summarization [37].

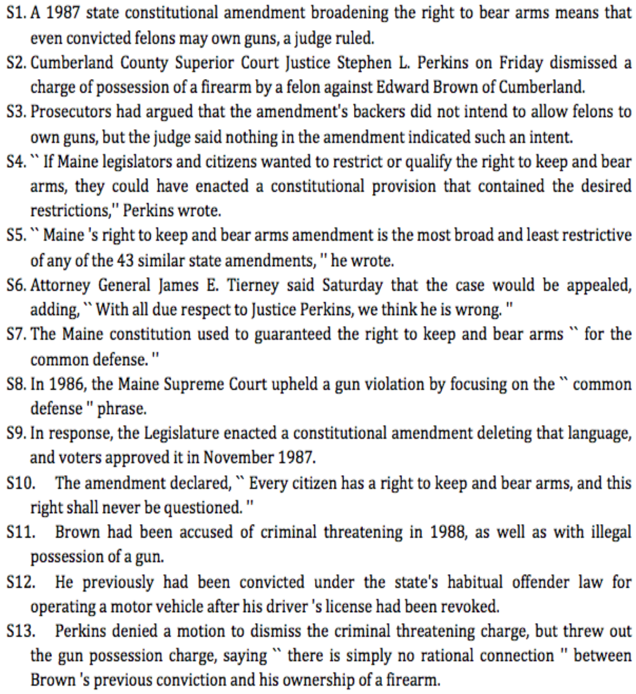
Besides the above discussed approaches, some work was also found dealing with the ensemble or combining techniques. Galgani et al. experimented with a rule-based hybrid approach for combining legal documents and using a supervised learning approach for catchphrase extraction from legal domain documents [25]. Hong et al. proposed a pipeline approach for multi-document summarization by combining the summaries from four portable unsupervised summarizers [26]. A supervised model is further used to select the resulting summary from the four candidates. Recently, Dutta et al. experimented with supervised and unsupervised ensemble approach using off the shelf algorithms for summarizing the contents of micro-blogging sites like Twitter with promising results [27]. So Galgani et al. and Dutta et al. approaches are designed for small text scenarios (catchphrase extraction and micro-blogs) and are not directly applicable to larger texts [25, 27]. While Hong et al. [26] do use cascading, they have done it directly for multi-document summarization [26].

Applying the cascading approach for SDS using the MDS technique on larger texts is the precise research gap that this work is attempting to fill. The intuition behind the idea is that *”Repetition of a concept adds emphasis and strengthens a point in human writing”.* Here MDS is expected to induce emphasis on concepts to be covered.

III. EXPERIMENTAL SETUP

A. Dataset

For the experimentation, the DUC 2001 dataset[1] released as the part of Document Understanding Conference 2001 [2] shared task on document summarization is used. The dataset consists of the summary dataset for both single document and multi-document summarization task. For the experiment here, only the single document dataset of 303 articles and their corresponding human-written summaries that are abstractive in nature is used. The dataset is from generic news domain. A sample of sentence tokenized input article and its corresponding reference summary is shown in *Figure 1 and 2* correspondingly.



S1. A 1987 state constitutional amendment broadening the right to bear arms means that even convicted felons may own guns, a judge ruled.

S2. Cumberland County Superior Court Justice Stephen L. Perkins on Friday dismissed a charge of possession of a firearm by a felon against Edward Brown of Cumberland.

S3. Prosecutors had argued that the amendment's backers did not intend to allow felons to own guns, but the judge said nothing in the amendment indicated such an intent.

S4. `` If Maine legislators and citizens wanted to restrict or qualify the right to keep and bear arms, they could have enacted a constitutional provision that contained the desired restrictions,'' Perkins wrote.

S5. `` Maine 's right to keep and bear arms amendment is the most broad and least restrictive of any of the 43 similar state amendments, '' he wrote.

S6. Attorney General James E. Tierney said Saturday that the case would be appealed, adding, `` With all due respect to Justice Perkins, we think he is wrong. ''

S7. The Maine constitution used to guaranteed the right to keep and bear arms `` for the common defense. ''

S8. In 1986, the Maine Supreme Court upheld a gun violation by focusing on the `` common defense '' phrase.

S9. In response, the Legislature enacted a constitutional amendment deleting that language, and voters approved it in November 1987.

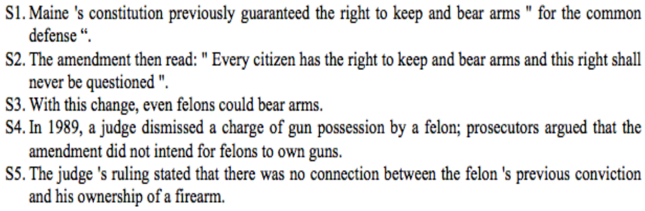
S10. The amendment declared, `` Every citizen has a right to keep and bear arms, and this right shall never be questioned. ''

S11. Brown had been accused of criminal threatening in 1988, as well as with illegal possession of a gun.

S12. He previously had been convicted under the state's habitual offender law for operating a motor vehicle after his driver 's license had been revoked.

S13. Perkins denied a motion to dismiss the criminal threatening charge, but threw out the gun possession charge, saying `` there is simply no rational connection '' between Brown 's previous conviction and his ownership of a firearm.

Figure 1. AP890722-0081 article from DUC 2001 dataset



S1. Maine 's constitution previously guaranteed the right to keep and bear arms " for the common defense “.

S2. The amendment then read: " Every citizen has the right to keep and bear arms and this right shall never be questioned ".

S3. With this change, even felons could bear arms.

S4. In 1989, a judge dismissed a charge of gun possession by a felon; prosecutors argued that the amendment did not intend for felons to own guns.

S5. The judge 's ruling stated that there was no connection between the felon 's previous conviction and his ownership of a firearm.

Figure 2. AP890722-0081 reference summary from DUC 2001 dataset

B. Method

For creating an extractive single document summary, the steps followed are shown through program flow in *Figure 3*. The input document for summarization was preprocessed for sentence and word tokenization. The tokenized input was then stemmed using the NLTK's porter stemmer and stop-word removed. The preprocessed input is given as input to the following algorithms implemented using various python libraries:

[1] https://www-nlpir.nist.gov/projcects/du/data/2001/

[2] https://duc.nist.gov/

1. **Frequency Summarizer (FB):** A simple word frequency based summarizer was coded using NLTK library of python. Based on the word frequency sentences are ranked after adding the frequency of each word in the sentence. The process involved creating a frequency-table individually for each document. A sentence score is calculated by the summation of each word frequency in the sentence and normalized with the length of the sentence. Once the sentences are scored, all the sentences above a threshold level are included as the part of the summary in ranking order.

2. **Kullback–Leibler divergence algorithm (KLD):** The Kullback–Leibler divergence is a statistical measure of the difference between two probability distributions [28]. It is calculated as in given below in (1).

$$D_{KL}(P \| Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)} \qquad (1)$$

With reference to extractive summarization, the algorithm attempts to find summary sentences that match the input document unigram distribution. The algorithm greedily adds sentences to a summary until the KL divergence value decreases from a threshold value [29]. This approach is also known as relative entropy algorithm.

3. **LexRank algorithm (LR):** LexRank algorithm is a graph based unsupervised approach inspired by the PageRank algorithm [14]. A graph is created based on sentences as nodes and edges as similarity measure calculated using IDF-modified cosine similarity as given in (2) for each document.

$$idf\text{-}\mathrm{mod}\text{-}\cos ine(x,y) = \frac{\sum_{w \in x,y} tf_{(w,x)} tf_{(w,y)} (idf_w)^2}{\sqrt{\sum_{x_i \in x} (tf_{(x_i,x)} idf_{x_i})^2} \times \sqrt{\sum_{y_i \in y} (tf_{(y_i,y)} idf_{y_i})^2}} \qquad (2)$$

From the similarity matrix obtained above, the sentences above the threshold are taken as summary sentences in the ranking order. In order to avoid highly similar sentences to be chosen for the summary, the sentences with similar scores to already included sentences were discarded.

4. **Latent Semantic Analysis algorithm (LSA):** LSA works at the semantic level by representing the document as a bag of words and clustering them to find the underlying concept of the document [11, 30].

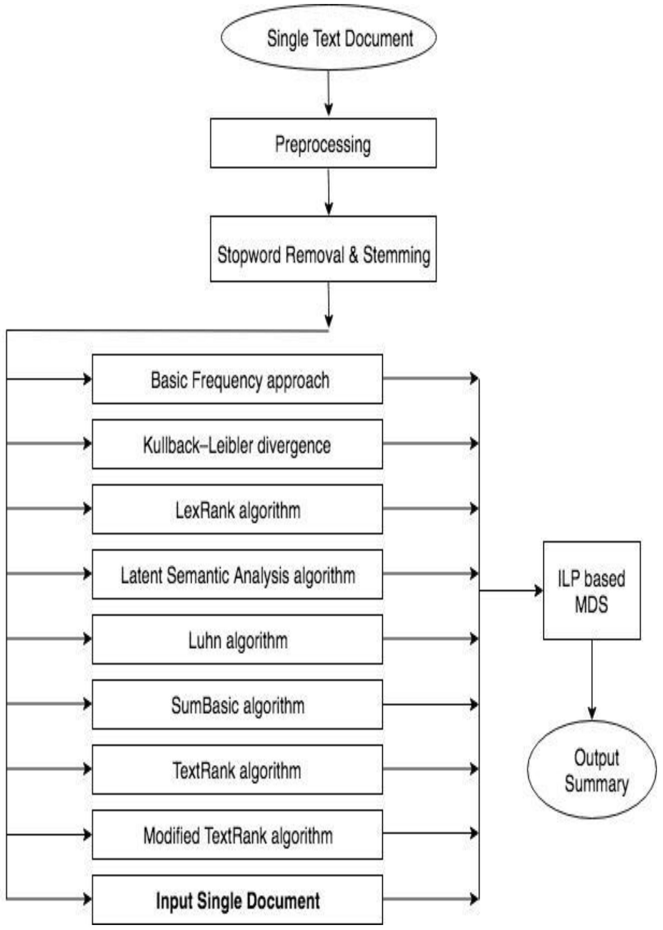


Figure 3. Steps followed for SDS using MDS

A *term* × *sentence* matrix is created for each sentence in the document. The matrix is normalized using tf-idf method to give more weight to less frequent terms. A data reduction technique, Singular Value Decomposition (SVD) is used to reduce the dimensionality and identify the important concepts in the text. The SVD is calculated as in (3).

$$A = U \Sigma V^T \qquad (3)$$

where A is an $m \times n$ matrix of *term×sentence*. U is an $m \times n$ column-orthonormal matrix whose columns are called left singular vectors, Σ is an $n \times n$ diagonal matrix whose diagonal elements are non-negative singular values sorted in descending order, and V is an $n \times n$ orthonormal matrix, whose columns are called right singular vectors [11].

From the right singular vector from matrix V^T, the sentence with higher variance (index value) is included as part of the summary. The process is iterated until the length of the required summary is reached.

5. **Luhn algorithm (Luhn):** Luhn algorithm is a heuristic-based approach to summarization that calculates the significance factor of each word in the text [31]. The significance factor is computed by counting the frequency of not so common words from the document and arranging them in decreasing order. The co-occurrence of significant words in a window of 4-5 words is used to score the sentences. The score is calculated by squaring the number of significant words and then dividing by the total number of words in the sentence. The sentences with higher significant scores are taken as the summary sentences.

6. **SumBasic algorithm (SB):** An algorithm based on the probability distribution of the words appearing in the input document [32]. The probability of each word is calculated as in (4).

$$p(w_i) = \frac{n_i}{N} \qquad (4)$$

where n_i is the number of times the word w_i appeared in the input document, and N is the total number of content word tokens in the input document. The sentence score is calculated as the average probability of the words in the sentence as in (5).

$$weight(S_j) = \sum_{w_i \ in \ S_j} \frac{p(w_i)}{\left|\{w_i \mid w_i \in S_j\}\right|} \qquad (5)$$

The highest scoring sentence having the highest probability word is chosen as the part of summary. Once selected, the probability of the words in chosen sentence is updated to get higher impact on the choice of subsequent sentences as in (6).

$$p_{new}(w_i) = p_{old}(w_i) * p_{old}(w_i) \qquad (6)$$

This updating probabilities of words helps in reducing the redundancy in sentence selection. This process is continued untill the required summary length is achieved.

7. **TextRank algorithm (TR):** TextRank is a graph-based ranking algorithm inspired by the PageRank algorithm used for web page ranking [33]. It is based on the understanding that the most important sentence will be the one that is most similar to every other sentence in the document. A cosine similarity matrix is calculated from the vector representation of sentences as in (7).

$$\cos ine\ similarity(A,B) = \frac{\sum_{i=1}^{n} A_i * B_i}{\sqrt{\sum_{i=1}^{n} A_i^2 * \sum_{i=1}^{n} B_i^2}} \qquad (7)$$

and converted to a graph with sentences as nodes and similarity score between two sentences as the edge.

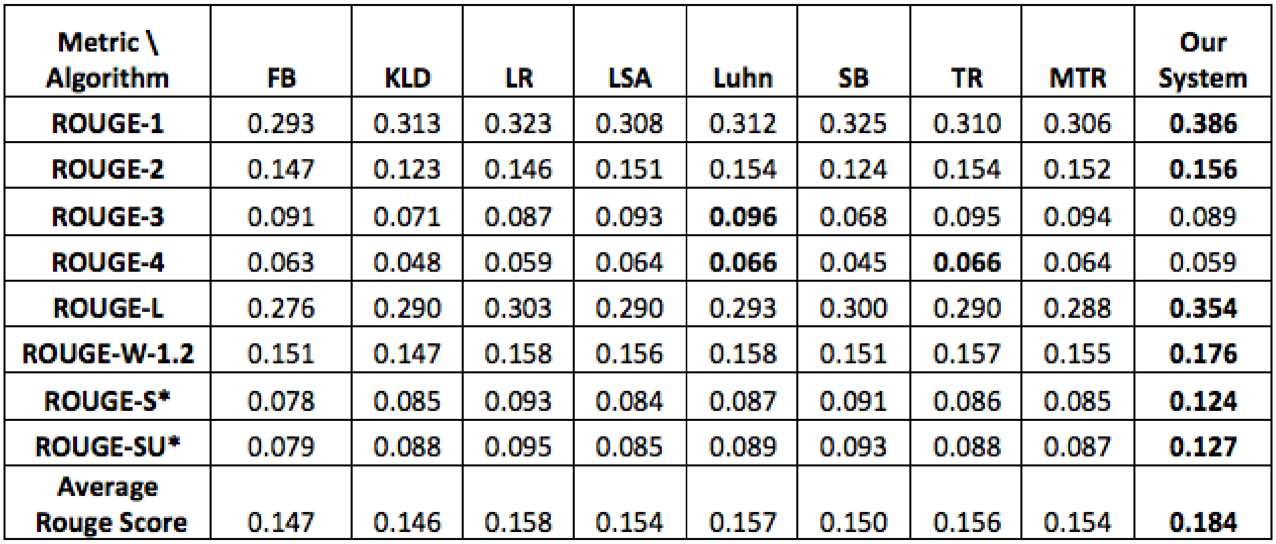
Graph nodes are ranked using pagerank and high rank sentences are assumed as the summary sentences.

8. **Modified TextRank algorithm (MTR):** A variation of TextRank algorithm based on Okapi BM25 similarity ranking function used for calculating the similarity between two sentences [34, 35]. The sentences are then ranked based on the similarity scores using the PageRank algorithm for summarization.

Each of these algorithms results in an extractive summary output of the input text and propagates its own unique strength in their resulting summary. All these output summaries taken together can be assumed as a multi-document summarization problem instance and is tackled through a multi-document summarization approach. The extractive summary outputs along with the original source document are cascaded to a multi-document summarization algorithm.

The MDS algorithm used here is based on sentence fusion approach presented by Filippova and Integer Linear Programming technique presented by Gillick and Favre [36, 37]. The method involves creating clusters of all the sentences from multiple documents based on the cosine similarity scores. From each cluster, iteratively adding the sentences to start and end nodes, a word graph is created. The linking or creation of nodes in the word graph is done using directed edges based on the part of speech of the word. From this directed graph, K shortest path is considered between the start and end node with the constraint that the no. of words in the path is not less than eight and must contain a verb. The path with minimum total weight is selected as the compressed sentence to be considered for summary. For sentence selection for the output summary, a concept based maximum coverage model using Integer Linear Programming (ILP) is used. The concepts are assumed as word bigrams from the source document weighted with no. of stage 1 output summaries they appear. The weighted concept is used as the objective function of the ILP problem. The occurence of concept in the sentence is used as a constraint to ILP. Thus, the concept-based ILP statement used to select the output summary sentence is as in (8).

Table 1: The performance of each individual extractive single document summarization algorithm along with our proposed system. (Multiply each value with 100 to get percent points. Rouge scores calculated with 95% confidence)



Metric \ Algorithm	**FB**	**KLD**	**LR**	**LSA**	**Luhn**	**SB**	**TR**	**MTR**	**Our System**
ROUGE-1	0.293	0.313	0.323	0.308	0.312	0.325	0.310	0.306	**0.386**
ROUGE-2	0.147	0.123	0.146	0.151	0.154	0.124	0.154	0.152	**0.156**
ROUGE-3	0.091	0.071	0.087	0.093	**0.096**	0.068	0.095	0.094	0.089
ROUGE-4	0.063	0.048	0.059	0.064	**0.066**	0.045	**0.066**	0.064	0.059
ROUGE-L	0.276	0.290	0.303	0.290	0.293	0.300	0.290	0.288	**0.354**
ROUGE-W-1.2	0.151	0.147	0.158	0.156	0.158	0.151	0.157	0.155	**0.176**
ROUGE-S*	0.078	0.085	0.093	0.084	0.087	0.091	0.086	0.085	**0.124**
ROUGE-SU*	0.079	0.088	0.095	0.085	0.089	0.093	0.088	0.087	**0.127**
Average Rouge Score	0.147	0.146	0.158	0.154	0.157	0.150	0.156	0.154	**0.184**

$$\text{Maximize:} \quad \sum_i w_i c_i \qquad (8)$$

$$\text{Subject to:} \quad \sum_j l_j s_j \leq L$$

$$s_j Occ_{ij} \leq c_i, \ \forall i, j \qquad (9)$$

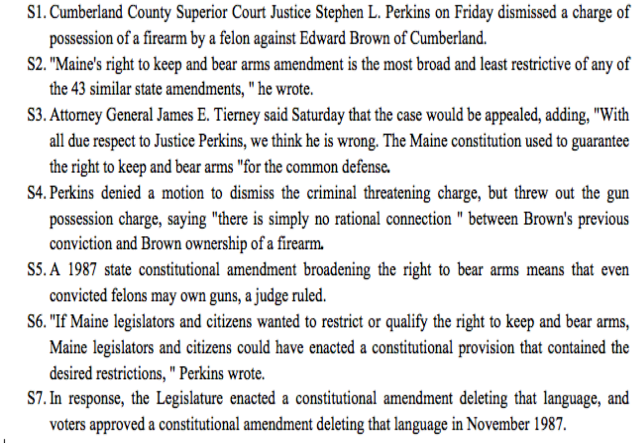
$$\sum_j s_j Occ_{ij} \geq c_i, \ \forall i \qquad (10)$$

$$c_i \in \{0,1\} \ \forall i$$

$$s_j \in \{0,1\} \ \forall j$$

where c_i is an indicator for the presence of concept i in the summary, w_i is its weight, s_j is an indicator for the presence of sentence j in the summary. Occ_{ij} is included to indicate the occurrence of concept i in sentence j. Constraints (9) and (10) check that selecting a sentence leads to selecting all the concepts it contains and selecting a concept is done only if it is present in at least one selected sentence.

Through this ILP approach, the selection of sentences for MDS output leads to the generation of output summary with maximum concept coverage. The sentences are taken verbatim or fused wherever possible. This resulting summary is the extractive summary generated for the input single document using the MDS technique. *Figure 4* shows the output obtained from our cascading technique for the input article given in *figure 1.*



S1. Cumberland County Superior Court Justice Stephen L. Perkins on Friday dismissed a charge of possession of a firearm by a felon against Edward Brown of Cumberland.

S2. "Maine's right to keep and bear arms amendment is the most broad and least restrictive of any of the 43 similar state amendments, " he wrote.

S3. Attorney General James E. Tierney said Saturday that the case would be appealed, adding, "With all due respect to Justice Perkins, we think he is wrong. The Maine constitution used to guarantee the right to keep and bear arms "for the common defense.

S4. Perkins denied a motion to dismiss the criminal threatening charge, but threw out the gun possession charge, saying "there is simply no rational connection " between Brown's previous conviction and Brown ownership of a firearm.

S5. A 1987 state constitutional amendment broadening the right to bear arms means that even convicted felons may own guns, a judge ruled.

S6. "If Maine legislators and citizens wanted to restrict or qualify the right to keep and bear arms, Maine legislators and citizens could have enacted a constitutional provision that contained the desired restrictions, " Perkins wrote.

S7. In response, the Legislature enacted a constitutional amendment deleting that language, and voters approved a constitutional amendment deleting that language in November 1987.

Figure 4: Output obtained from MDS approach for the input article.

IV. Result and Discussion

Table 1 shows the consolidated results obtained from each of the algorithms from stage 1 as well as from stage 2 after cascading through the multi-document summarization algorithm. The scores are obtained from the automatic evaluation of output summaries using ROUGE metrics [39]. ROUGE is a collection of metrics used to evaluate various measures of output summaries. The ROUGE metric collection consists of ROUGE-N measures from 1-gram to 4-gram, ROUGE-L measures the longest common subsequence (LCS), ROUGE-W measures weighted-LCS favouring consecutive LCS, ROUGE-S measures skip-bigram co-occurrences and ROUGE-SU measures skip-unigram co-occurrences. It gives precision and recall values as output by comparing an automatically produced summary against a reference or a set of references (human-produced) summary.

From the collection, ROUGE-1 and ROUGE-2 have been found reported in most summary related literature. Here, in *table 1* all the metric scores from the collection have been reported for better analysis. For each metric, F-scores have been reported that combines both precision and recall scores for better accuracy.

It can be observed from *Table 1* that barring a few cases (ROUGE-3 and ROUGE-4), the average ROUGE scores of the proposed system performs better than each of the single document summarization approaches for each metric. Also, the average F-score of the proposed system outperforms each of the individual single document summarizers by at least 2.6 percent points. The ILP based multi-document summarization approach is able to capture more concepts and information as compared to the single document summarization algorithms and is evident from *figure 4.*

V. CONCLUSION AND FUTURE SCOPE

In this paper, a novel idea of performing single document summarization using multi-document summarization via a two-step process has been proposed. Given a source document to be summarized, first, a set of single document summarization algorithm used on the source document. Next, the output of these summarizers along with the original document is passed as a multi-document summarization problem instance to an ILP based multi-document summarizer. The saliency of the approach lies in the fact that the multi-document summarizer is able to exploit the strengths of each of the single document summarizers (through the corresponding generated summary). The cascaded approach experimented performs better than each of the single document summarizers by at least 2.6 percent points, thereby indicating that posing single document summarization as a multi-document summarization problem in this manner is a noteworthy approach. Since the domain of the dataset is generic news article, this approach can be tested on other domain-specific, single document summarization datasets in future. And also the claim can be used on other popular datasets used for single document summarization.

REFERENCES

[1] I. Mani*, "Advances in automatic text summarization.* "MIT press, **1999.**

[2] D.R. Radev, E. Hovy, K. McKeown, *"Introduction to the special issue on summarization."* Computational linguistics, 28(4), pp.399-408, **2002.**

[3] H. Saggion, T. Poibeau, *"Automatic text summarization: Past, present and future."* In Multi-source, multilingual information extraction and summarization (pp. 3-21). Springer, Berlin, Heidelberg, **2013.**

[4] M.Y. Kan, K.R. McKeown, J.L. Klavans, *"Applying natural language generation to indicative summarization."* In Proceedings of the 8th European workshop on Natural Language Generation-Volume 8 (pp. 1-9). Association for Computational Linguistics, **2001.**

[5] H. Saggion, G. Lapalme, *"Generating indicative-informative summaries with sumUM."* Computational linguistics, 28(4), pp.497-526, **2002.**

[6] J.L. Klavans, M.Y. Kan, K. McKeown, *"Domain-specific informative and indicative summarization for information retrieval."*, In proceedings of the Workshop on text summarization (DUC 2001), **2001.**

[7] T. Hirao, M. Nishino, M. Nagata, *"Oracle Summaries of Compressive Summarization."* In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers) (Vol. 2, pp. 275-280), **2017.**

[8] W. Bosma, *"Query-based summarization using rhetorical structure theory."* LOT Occasional Series, 4, pp.29-44, **2005.**

[9] H.G. Silber, K.F. McCoy, *"An efficient text summarizer using lexical chains."* In Proceedings of the first international conference on Natural language generation-Volume 14 (pp. 268-271). Association for Computational Linguistics, **2000.**

[10] R. Barzilay, M. Elhadad*, "Using lexical chains for text summarization."* Advances in automatic text summarization, pp.111-121, **1999.**

[11] Y. Gong, X. Liu, *"Generic text summarization using relevance measure and latent semantic analysis."* In Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 19-25). ACM, **2001.**

[12] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, R. Harshman, *"Indexing by latent semantic analysis."* Journal of the American society for information science, 41(6), pp.391-407, **1990**.

[13] S.M. Harabagiu, F. Lacatusu, *"Generating single and multi-document summaries with gistexter."* In Document Understanding Conferences (pp. 11-12), **2002**.

[14] G. Erkan, D.R. Radev, *"Lexrank: Graph-based lexical centrality as salience in text summarization."* Journal of artificial intelligence research, 22, pp.457-479, **2004.**

[15] O. Sornil, K. Gree-Ut, *"An automatic text summarization approach using content-based and graph-based characteristics."* In 2006 IEEE Conference on Cybernetics and Intelligent Systems (pp. 1-6). IEEE, **2006.**

[16] H. Chen, T. Ng, *"An algorithmic approach to concept exploration in a large knowledge network (automatic thesaurus consultation): Symbolic branch and bound search vs. connectionist Hopfield net activation."* Journal of the American Society for Information Science, 46(5), pp.348-369, **1995.**

[17] C. Smith, H. Danielsson, A. Jönsson, *"A more cohesive summarizer. "* Proceedings of COLING 2012: Posters, pp.1161-1170, **2012.**

[18] J. Cheng, M. Lapata, *"Neural summarization by extracting sentences and words. "* In proc. Of 54th Annual Meeting of the Association for Computational Linguistics, **2016.**

[19] R. Nallapati, F. Zhai, B. Zhou, *"Summarunner: A recurrent neural network based sequence model for extractive summarization of documents."* In Thirty-First AAAI Conference on Artificial Intelligence, **2017.**

[20] D.R. Timothy, T. Allison, S. Blair-goldensohn, J. Blitzer, A. Elebi, S. Dimitrov, E. Drabek, A. Hakim, W. Lam, D. Liu, J. Otterbacher, *"MEAD a platform for multidocument multilingual text summarization."* In International Conference on Language Resources and Evaluation, **2004.**

[21] A. Celikyilmaz, D. Hakkani-Tur*, "A hybrid hierarchical model for multi-document summarization."* In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (pp. 815-824). Association for Computational Linguistics, **2010.**

[22] M.A. Fattah, *"A hybrid machine learning model for multi-document summarization."* Applied intelligence, 40(4), pp.592-600, **2014.**

[23] Z. Cao, F. Wei, L. Dong, S. Li, M. Zhou, *"Ranking with recursive neural networks and its application to multi-document summarization. "* In Twenty-ninth AAAI conference on artificial intelligence**, 2015.**

[24] T. Hirao, M. Nishino, M. Nagata, *"Oracle Summaries of Compressive Summarization."* In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers) (Vol. 2, pp. 275-280), **2017.**

[25] F. Galgani, P. Compton, A. Hoffmann, *"Combining different summarization techniques for legal text."* In Proceedings of the Workshop on Innovative Hybrid Approaches to the Processing of Textual Data (pp. 115-123). Association for Computational Linguistics, **2012.**

[26] K. Hong, M. Marcus, A. Nenkova, *"System combination for multi-document summarization."* In Proceedings of the 2015 conference on empirical methods in natural language processing (pp. 107-117), **2015.**

[27] S. Dutta, V. Chandra, K. Mehra, A.K. Das, T. Chakraborty, S. Ghosh, *"Ensemble Algorithms for Microblog Summarization."* IEEE Intelligent Systems, 33(3), pp.4-14, 2018.

[28] S. Kullback, "Information theory and statistics." Courier Corporation, **1997.**

[29] A. Haghighi, L, Vanderwende, *"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 (pp. 362-370). Association for Computational Linguistics, **2009.**

[30] J. Steinberger, K. Jezek*, "Using latent semantic analysis in text summarization and summary evaluation."* Proc. ISIM, 4, pp.93-100, **2004.**

[31] H.P. Luhn, *"The automatic creation of literature abstracts."* IBM Journal of research and development, 2(2), pp.159-165, **1958.**

[32] A. Nenkova, L. Vanderwende, *"The impact of frequency on summarization."* Microsoft Research, Redmond, Washington, Tech. Rep. MSR-TR-2005, 101, **2005.**

[33] R. Mihalcea, P. Tarau, *"Textrank: Bringing order into text."* In Proceedings of the 2004 conference on empirical methods in natural language processing, **2004.**

[34] S. Robertson, H. Zaragoza, *"The probabilistic relevance framework: BM25 and beyond. "* Foundations and Trends® in Information Retrieval, 3(4), pp.333, **2009.**

[35] F. Barrios, F. López, L. Argerich, R. Wachenchauzer, *"Variations of the similarity function of textrank for automated summarization.",* In Proc. Argentine Symposium on Artificial Intelligence, ASAI, **2016.**

[36] K. Filippova, *"Multi-sentence compression: Finding shortest paths in word graphs."* In Proceedings of the 23rd International Conference on Computational Linguistics(pp. 322-330). Association for Computational Linguistics, **2010.**

[37] D. Gillick, B. Favre, *"A scalable global model for summarization."* In Proceedings of the Workshop on Integer Linear Programming for Natural Language Processing (pp. 10-18). Association for Computational Linguistics, **2009.**

[38] C. Li, X. Qian, Y. Liu, *"Using supervised bigram-based ILP for extractive summarization."* In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (Vol. 1, pp. 1004-1013), **2013.**

[39] C.Y. Lin, *"Rouge: A package for automatic evaluation of summaries."* Text Summarization Branches Out, **2004.**

[40] R. Garcia, R. Lima, B. Espinasse, H. Oliveira*, "Towards coherent single-document summarization: an integer linear programming-based approach."* In Proceedings of the 33rd Annual ACM Symposium on Applied Computing (pp. 712-719). ACM, **2018.**

[41] K. Filippova, M. Strube, "*Sentence fusion via dependency graph compression."* In Proceedings of the Conference on Empirical Methods in Natural Language Processing (pp. 177-185). Association for Computational Linguistics, **2008.**

[42] K.S. Kumar, S. Prasad, S. Banwral, V.B. Semwal, V.B., *"Sports video summarization using priority curve algorithm."* International Journal on Computer Science & Engineering, 2(9), pp.2996-3002, **2010.**

[43] S. Saraswathi, M. Hemamalini, S. Janani, V. Priyadharshini, *"Multi-document Summarization for Query Answering E-learning System."* International Journal on Computer Science and Engineering (IJCSE), 3(3), pp.1147-1154, **2011.**

Authors Profile

Ms. Sandhya Singh pursued her Bachelor of Science (CS) from Banasthali University, Rajasthan and Master of Science (CS) from Banaras Hindu University, UP. She is currently pursuing her Ph.D. from Banasthali University, Rajasthan, India in the field of NLG. Her research interest includes the field of Machine Translation, Natural Language Generation and Text Summarization.

Mr. Kevin Patel is B.Tech. (CSE) from NIT, Surat and M.Tech. (CSE) from IISC, Banglore. He is currently pursuing his PhD from IIT Bombay, Mumbai, India in the area of Explainabilty. His research interest includes the field of Natural Language Processing, Machine Learning, Deep Learning and Explainable Artificial Intelligence.

Mr. Krishnanjan Bhattacharjee is a Principal Technical Officer at C-DAC, Pune, India. He obtained his Ph.D. from Pune University in the field of Discourse Analysis. His research interest includes the areas of Big Data Analytics, Machine Learning in terms of Language model creations for training, Q-A systems, Decision Support Systems and Sentiment Analysis.

Mr. Hemant Darbari is the Director General of C-DAC, India, a premier R&D organization in Information Technologies and Electronics in the country. He has been working in the area of Artificial Intelligence since 1985. His research interests includes the area of Natural Language Processing (NLP), Machine assisted Translation (MT), Information Extraction and Information Retrieval (IE/IR), Intelligent Language Tutoring systems, Speech Technology (Automatic Speech Recognition & Text to Speech System) and Mobile computing and Simulations.

Ms. Seema Verma received her PhD at Banasthali University. She is a Professor at the Department of Electronics, Banasthali University, Rajasthan, India. Her research interest includes issues related to communication System, wireless communication, VLSI Design, MIMO of DM, cryptography & networks security, turbo codes, LDPC codes. She is a Fellow of IETE and member of Indian Science Congress, ISTE.