A New Automatic Multi-Document Text Summarization Technique using Topic-Modeling

Rajendra Kumar Roul[1], Samarth Mehrotra[1], Yash Pungaliya[1], Jajati Keshari Sahoo[2]

[1] Department of Computer Science, BITS-Pilani, K.K.Birla Goa Campus, Zuarinagar, Goa-403726, India, rkroul@goa.bits-pilani.ac.in, samarth.1397@gmail.com, yashpungaliya@gmail.com
[2] Department of Mathematics, BITS-Pilani, K.K.Birla Goa Campus, Zuarinagar, Goa-403726, India, jksahoo@goa.bits-pilani.ac.in

Abstract. This paper proposes a novel methodology to generate an extractive text summary from a corpus of documents. Unlike most existing methods, our approach is designed in such a way that the final generated summary covers all the important topics from the corpus of documents. We propose a heuristic method which uses the Latent Dirichlet Allocation technique to identify the optimum number of independent topics present in a corpus of documents. Some of the sentences are identified as the important sentences from each independent topic using a set of word and sentence level features. In order to ensure that the final summary is coherent, we suggest a novel technique to reorder the sentences based on sentence similarity. The use of topic modeling ensures that all the important content from the corpus of documents is captured in the extracted summary which in turn strengthen the summary. Experimental results show that the proposed approach is promising.

Keywords: Extractive, Multi-document, ROUGE, Summarization, LDA, Topic Modeling

1 Introduction

Generating a brief summary from a large amount of textual data is an important research area in the field of computer science. A summary can be defined as ‘some brief statements that present the main aspects of a subject (i.e., a corpus of documents) in a concise manner’. The tools and techniques used for automatic text summarization help to convert the raw textual information into a summary without any human intervention.

Based on the nature of the summary generated from text documents, summarization can be of two types: *abstractive* and *extractive* [1]. Abstractive summaries are generated by interpreting the raw text and generating the same information in a different and concise form. Modern day abstractive summarization systems uses complex neural network based architectures such as RNNs and LSTMs [**?**]. On the other hand, extractive summarization is implemented by identifying the important sections of the text, processing and combining them to form a meaningful summary [2]. Text summarization can be further divided into two categories: *single* and *multi-text* summarization. In single text summarization, text is summarized from one document where as Multi-document text summarization systems are able to generate reports that are rich in important information, and concisely present varying views that span multiple documents. Many researchers have worked in the field of text summarization [**?**] [3] [4] [5] [6], however research work in the domain of multi-document text summarization using topic modeling is limited.

The paper proposes a four-step procedure to generate the final summary from the corpus. In the first step, Latent Dirichlet Allocation (LDA) is used to select only a subset of sentences per topic. The second step calculates fifteen important features for each sentence. In the third step, an aggregate score for each sentence is calculated using those fifteen features. This aggregate score is used to select the important sentences which are to be the part of the summary. In the fourth and final step, the important sentences are ordered to form a coherent summary. Empirical results on different DUC datasets justify the suitability and importance of the proposed approach for multi-document text summarization.

Rest of the paper is organized as follows: Section 2 describes the detailed methodology to summarize the corpus of documents. Results and analysis of the experimentation have been carried out in Section 3, which is followed by the conclusion of the work in Section 4.

2 Proposed Approach

Consider a corpus C of documents d = $\{d_1, d_2, ..., d_i\}$. Initially, all these documents are merged into a large document called D_{large}. Then D_{large} is split into n sentences, i.e., $\{s_1, s_2..., s_n\}$. The proposed approach reduces these n sentences to n/X (i.e., length of the final summary), where X is a user defined value and greater than 2 [3]. The experiments are carried out by using X=13[4], as n is a reasonably large value in most collection of documents in the DUC dataset. Final summary is generated using the following steps.

1. *LDA is used to reduce number of sentences from n to 2n/X*:
   Assume that the set of sentences $\{s_1, s_2, ..., s_n\}$ consists of k independent topics. Although the problem of identification of the exact number of topics in a corpus is an unsolved problem, we propose a heuristic method which help us to decide the value of k. This heuristic method is discussed later in this step. After identifying a reasonably good k, we perform topic modeling on the set of n-sentences (each sentence is consider as an individual document) using LDA [7]. Gensim[5] a python library is used for this purpose. Based on these k-topics that are generated from the n-sentences, a sentence-topic matrix is created as shown in Table 1 where, each entry w_{ij} denotes the weight j^{th} topic of i^{th} sentence. To ensure that the final summary covers all the topics, we select a total of $2n/(Xk)$ sentences from each topic (i.e., from each column of the Table 1) having maximum weight. Since this procedure is carried for k topics, we will get a total of $2n/X$ sentences. However, some

[3] since reduction is being performed, $2/X < 1$

[4] experimental results generate a good summary for X=13

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

Table 1: Sentence-Topic Matrix

	$Topic_1$	$Topic_2$	...	$Topic_k$
s_1	w_{11}	w_{12}	...	w_{1k}
s_2	w_{21}	w_{22}	...	w_{2k}
s_3	w_{31}	w_{32}	...	w_{3k}
.	.	.	.	...
.	.	.	.	...
.	.	.	.	...
s_n	w_{n1}	w_{n2}	...	w_{nk}

sentences might be chosen from multiple columns, therefore, we will always have less than or equal to $2n/X$ sentences. Algorithm 1 illustrates the implementation.

Algorithm 1: Selection of $2n/Xk$ sentences from each topic

```
Input: $n \times k$ sentence-topic matrix
Output: $2n/Xk$ sentences
$reduced\_matrix[iterations][k] \leftarrow \phi$
for all $t \in (0, k-1)$ do
    for all $i \in (0, iterations-1)$ do
        $max\_elem \leftarrow max(sentence\_topic\_matrix[][t])$
        $max\_index \leftarrow index\ of\ max\_elem\ in\ matrix$
        $reduced\_matrix[i][t] \leftarrow max\_index$
        $sentence\_topic\_matrix[max\_index][t] \leftarrow 0$
    end for
end for
return $reduced\_matrix$
```

Heuristic method to decide the number of independent topics (k):

To decide an appropriate number of independent topics[6] (k), the following steps are used and is illustrated in Algorithm 2.

i. Initially the number of topics i.e., $k_{initial}$ is set to a large value, say M (M is set to $2i$ where i is the number of documents of the corpus C).

ii. Topic modeling is performed on n sentences as discussed in step 1 assuming that the number of topics as $k_{initial}$ and then the similarities between topics are calculated.

iii. Since each topic is a probability distribution over the vocabularies of C, the topic similarity is computed using *KL*-Divergence [8], Hellinger's Distance[7], and Jensen Shannon Divergence [9].

iv. A topic-topic similarity matrix is generated as shown in Table 2, where $Topic_{ij}$ denotes the aggregate similarity between $Topic_i$ and $Topic_j$, and $Topic_{ii}$ =1.

v. In this entire topic-topic similarity matrix, apart from the diagonal elements (as diagonal elements are 1), if any pair of topics which have a similarity greater than 0.40 (user defined threshold) is found out then that is an indication that the

[6] by independent means low similarity between topics

[7] www.encyclopediaofmath.org/index.php?title=Hellinger_distance& oldid=16453

initial estimate of the number of topics $k_{initial}$ is higher than the appropriate number of independent topics (k). So, we reduce $k_{initial}$ by 1, i.e., $k_{initial}$ = $k_{initial}$-1 and repeat the Steps (ii-v) again.

vi. Eventually a stage will come where we would have a number of topics k i.e., in the topic-topic similarity matrix between any pair of topics, the similarity score is less than 0.4. This will ensure that we will reach a suitable number of independent topics which serves the purpose of summarization as it covers the breadth of topics of the original corpus.

Table 2: Topic-Topic matrix

	$Topic_1$	$Topic_2$	...	$Topic_k$
$Topic_1$	$Topic_{11}$	$Topic_{12}$	...	$Topic_{1k}$
$Topic_2$	$Topic_{21}$	$Topic_{22}$	...	$Topic_{2k}$
$Topic_3$	$Topic_{31}$	$Topic_{32}$	...	$Topic_{3k}$
.	.	.	...	.
.	.	.	...	.
.	.	.	...	.
$Topic_k$	$Topic_{k1}$	$Topic_{k2}$	...	$Topic_{kk}$

Algorithm 2: Selection of appropriate number of independent topics

```
Input: the set of sentences ${s_1, s_2, ..., s_n}$
Output:$num\_topics$ //number of topics
$num\_topics \leftarrow 2 * num\_docs$// number of documents
while num_topics >0 do
    $ldamodel \leftarrow$ generate an LDA model using num_topics
    $similairity\_matrix[num\_topics][num\_topics] \leftarrow \phi$
    $similarity\_matrix \leftarrow$ generated topic-topic similarity matrix
    if (element $\in$ similarity_matrix) >0.4 and element is not a diagonal element then
        $num\_topics \leftarrow num\_topics - 1$
    else
        exit
    end if
end while
return num_topics
```

2. *Identifying important word and sentence level features*:
Fifteen important features are identified to generate an aggregate score for each sentence s of a document d. Those features are length, weight, density, title words, upper-case words, quoted text, numerical words, alphanumeric words, cue-phrase, similarity among sentences, similarity among paragraphs, LSI-based scores, concept-based scores, Doc2Vec scores, and Word2Vec scores of each s.
3. *Reducing 2n/X sentences to n/X sentences*:
After performing the topic modeling with the appropriate number of topics k, and selecting $\frac{2n}{Xk}$ sentences from each topic, we are left with a total of $2n/X$ sentences. For each topic, the following steps are used:

i. All fifteen word and sentence level features for each sentence (as discussed in Step 2) are extracted in the set of $2n/Xk$ sentences.

ii. An aggregate score for each sentence is calculated using Equation 1.

$$\frac{\text{sum of all normalized feature scores}}{\text{total number of features}} \tag{1}$$

iii. All the sentences are ranked based on their aggregate scores and the top n/Xk sentences are selected among them.
iv. Steps (i-iii) are carried out for all k topics, hence we now have a total of n/X sentences. However, since some of the sentences might be chosen for more than one time as discussed in Step 1, therefore we will always have less than or equal to n/X sentences.

4. *Ordering of the sentences*:
Now we have a list of n/X sentences, but these sentences have not yet been ordered. To ensure that the generated summaries are coherent, the following steps are followed:
i. Since the corpus C contains a total of i documents, hence we have a total of i opening lines (opening line indicates the first sentence of a document), i.e., one line for each document.
ii. After traversing through the list of n/X sentences, if we encounter an opening line OL during this traversal, then OL will be chosen as the summary's first sentence. Unlikely, when none of the n/X sentences contain any opening line then a sentence is selected randomly from this list of n/X sentences as the opening line.
iii. A sentence-sentence similarity matrix is generated using WordNet based similarity. The order of the sentences is stored in a list. The first entry in this list is the starting sentence. We then go to the r^{th} row of the sentence-sentence similarity matrix where r is the starting index. We traverse this row and find the maximum element in the row apart from the diagonal elements. Consider that this element is found in the j^{th} column. We now add j to our list of ordered sentences and then jump to the j^{th} row of the matrix. Then the element having the maximum score from the j^{th} row excluding the previous selected sentence and the j^{th} column is selected. This process is repeated till all the sentences are added to the list. This ensures that consecutive sentences are similar and so the summary will be coherent.

2.1 Generating Extractive gold summaries from Human written summaries

Python Natural Language Toolkit[8] has been used to tokenize the documents into sentences. The way extractive gold summaries are created from the human written gold summaries (available in DUC datasets) are explained below.

i) Each document d of C is parsed sentence by sentence. For each sentence $s \in d$, calculate the number of keywords it has in common with each of the four human written gold summaries of the corpus C. The number of common words between each s and four human written gold summaries gives the score for the sentence s. This way the scores for all the sentences of a document are calculated.
ii) Now rank all the sentences of a document d based on their scores computed in Step i). Top-m important sentences are selected from the ranked sentences in order to form the extractive gold summary of d. For experimental purpose, we have taken m = 5, i.e., each document has an extractive gold summary of five sentences. Repeat steps i) and ii) for all documents of the corpus C.

[8] http://www.nltk.org/

3 Experimental Analysis

For experimental purpose, Document Understanding Conference (DUC)[9] datasets are used, where each dataset having four human written summaries. DUC-2005 has 50 Document sets and DUC-2007 has 45 document sets.

ROUGE or Recall-Oriented Understudy for Gisting Evaluation score [10] is generally used to measure the performance of text summarization. ROUGE-N measures the unigram, bigram, trigram, and higher order n-gram overlap between the summaries generated by the system (i.e., by the proposed approach) and the gold summary (either human written or extractive). ROUGE-1 and ROUGE-2 are used to compute the F-measure value for unigram and bigram matching respectively.

3.1 Discussion

For experimental purpose, the stop-words are removed from all DUC-datasets while computing the ROUGE score. Tables 3 - 6 show the performance of ROUGE-1 score of extractive and human written gold summary on different DUC datasets. Similarly, Tables 7 - 10 show the performance of ROUGE-2 score of extractive and human written gold summary on different DUC datasets. From the results of these tables, the following points can be observed

i. The ROUGE-N scores of human written summaries are less than the extractive summaries and this is because as in the original document (i.e., the document of the DUC dataset), people try to use different words and paraphrase the documents in their own way, hence most of the words of the human written summaries do not match with the original documents. However, the extractive gold summaries takes care of the words common between the sentence of the original document and the four human gold summaries as discussed in Section 2.1.

ii. If the stop-words are removed from the corpus, then for every concise summary, ROUGE-1 score alone may suffice [10] which are reflected when we compared the obtained ROUGE-1 with the ROUGE-2 scores.

4 Conclusion

The paper developed an approach that generates a concise summary using multi-document text summarization. Initially, the topic modeling using LDA is run on each single sentence of a corpus and a subset of sentences are selected per topic. Next, fifteen important word and sentence level features are identified and using these features, the aggregate score of each sentence is calculated. Important sentences are selected based on the aggregate scores and finally sentences are ordered to form a coherent summary. Unlike most of the other techniques, the generated summaries will not only contain sentences that have the best structural features, but also cover all the key topics of the corpus. Future work includes further improvement of the two heuristic methods i.e., identifying a suitable number of topics and reordering of the sentences.

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

Table 3: ROUGE-1 Extractive (DUC-2005)

Doc-Id	Recall	Precision	F-Score
D301	0.43273	0.74537	0.54757
D307	0.44990	0.77817	0.57016
D311	0.37160	0.86547	0.51995
D313	0.47493	0.79914	0.59578
D321	0.55001	0.75214	0.63538
D324	0.20625	0.89593	0.33531
D331	0.71323	0.72054	0.71687
D332	0.74888	0.68653	0.71635
D343	0.40890	0.85114	0.55242
D345	0.33215	0.86331	0.47973
D346	0.47539	0.74505	0.58043
D347	0.78038	0.71554	0.74656
D350	0.45230	0.75985	0.56706
D354	0.47950	0.78655	0.59579
D391	0.68774	0.68914	0.68844
Average	0.50425	0.77692	0.58985

Table 4: ROUGE-1 Human written (DUC-2005)

Doc-Id	Recall	Precision	F-Score
D301	0.46291	0.22785	0.30538
D307	0.51105	0.26882	0.35232
D311	0.47226	0.60594	0.53081
D313	0.68563	0.32934	0.44495
D321	0.59350	0.22408	0.32533
D324	0.19080	0.64404	0.29438
D331	0.76016	0.17862	0.28927
D332	0.58571	0.13851	0.22404
D343	0.54839	0.33601	0.41670
D345	0.35294	0.57365	0.43701
D346	0.51151	0.27370	0.35659
D347	0.61161	0.13117	0.21602
D350	0.54472	0.28926	0.37786
D354	0.52239	0.22173	0.31132
D391	0.57046	0.32261	0.41214
Average	0.52826	0.31768	0.35294

References

1. K. Ganesan, C. Zhai, and J. Han, “Opinosis: a graph-based approach to abstractive summarization of highly redundant opinions,” in *Proceedings of the 23rd International Conference on Computational Linguistics*. Association for Computational Linguistics, 2010, pp. 340–348.
2. N. Moratanch and S. Chitrakala, “A survey on extractive text summarization,” in *Computer, Communication and Signal Processing (ICCCSP), 2017 International Conference on*. IEEE, 2017, pp. 1–6.
3. R. K. Roul, J. K. Sahoo, and R. Goel, “Deep learning in the domain of multi-document text summarization,” in *International Conference on Pattern Recognition and Machine Intelligence*. Springer, 2017, pp. 575–581.
4. F. Liu, J. Flanigan, S. Thomson, N. Sadeh, and N. A. Smith, “Toward abstractive summarization using semantic representations,” *arXiv preprint arXiv:1805.10399*, 2018.
5. S. Narayan, S. B. Cohen, and M. Lapata, “Ranking sentences for extractive summarization with reinforcement learning,” *arXiv preprint arXiv:1802.08636*, 2018.
6. J. Zhang, J. Tan, and X. Wan, “Towards a neural network approach to abstractive multi-document summarization,” *arXiv preprint arXiv:1804.09010*, 2018.
7. D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent dirichlet allocation,” *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.
8. S. Kullback and R. A. Leibler, “On information and sufficiency,” *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.
9. B. Fuglede and F. Topsoe, “Jensen-shannon divergence and hilbert space embedding,” in *Information Theory, 2004. ISIT 2004. Proceedings. International Symposium on*. IEEE, 2004, p. 31.
10. C.-Y. Lin, “Rouge: A package for automatic evaluation of summaries,” in *Text summarization branches out: Proceedings of the ACL-04 workshop*, vol. 8, 2004, pp. 74–81.

Table 5: ROUGE-1 Extractive (DUC-2007)

Doc-Id	Recall	Precision	F-Score
D0701A	0.54533	0.11414	0.18877
D0702A	0.58369	0.20606	0.30459
D0703A	0.51667	0.23308	0.32124
D0705A	0.55703	0.21627	0.31157
D0706B	0.49161	0.17402	0.25705
D0707B	0.30196	0.24290	0.26923
D0708B	0.51429	0.25862	0.34417
D0709B	0.79960	0.15011	0.25277
D0710C	0.38978	0.29897	0.33839
D0711C	0.67520	0.29781	0.41332
D0712C	0.46970	0.26007	0.33477
D0713C	0.47486	0.20706	0.28838
D0714D	0.44698	0.24968	0.32039
D0715D	0.59544	0.27756	0.37862
D0716D	0.54277	0.22412	0.31724
Average	0.52699	0.22736	0.30936

Table 7: ROUGE-2 Extractive (DUC-2005)

Doc-Id	Recall	Precision	F-Score
D301	0.26548	0.45738	0.33596
D307	0.30684	0.53084	0.38889
D311	0.26914	0.62703	0.37662
D313	0.34112	0.57410	0.42796
D321	0.37891	0.51822	0.43775
D324	0.17576	0.76435	0.28579
D331	0.46161	0.46634	0.46396
D332	0.50026	0.45860	0.47853
D343	0.31701	0.66007	0.42831
D345	0.26956	0.70092	0.38937
D346	0.29128	0.45659	0.35567
D347	0.57447	0.52673	0.54957
D350	0.29049	0.48812	0.36422
D354	0.32333	0.53047	0.40177
D391	0.41050	0.41134	0.41092
Average	0.34505	0.54474	0.40635

Table 9: ROUGE-2 Extractive (DUC-2007)

Doc-Id	Recall	Precision	F-Score
D0701A	0.15177	0.03173	0.05249
D0702A	0.17049	0.06013	0.08891
D0703A	0.18436	0.08291	0.11438
D0705A	0.18883	0.0732	0.10550
D0706B	0.20433	0.07222	0.10672
D0707B	0.07480	0.06013	0.06667
D0708B	0.18966	0.09510	0.12668
D0709B	0.36747	0.06887	0.11601
D0710C	0.13208	0.10124	0.11462
D0711C	0.25001	0.11017	0.15294
D0712C	0.18845	0.10420	0.13420
D0713C	0.12605	0.05488	0.07647
D0714D	0.13470	0.07521	0.09652
D0715D	0.23429	0.10904	0.14882
D0716D	0.18343	0.07561	0.10708
Average	0.18538	0.07831	0.10720

Table 6: ROUGE-1 Human written (DUC-2007)

Doc-Id	Recall	Precision	F-Score
D0701A	0.73705	0.05485	0.10210
D0702A	0.67704	0.08788	0.15557
D0703A	0.44531	0.28571	0.34809
D0705A	0.71311	0.17920	0.28642
D0706B	0.61508	0.13158	0.21678
D0707B	0.48606	0.38486	0.42958
D0708B	0.49407	0.35920	0.41597
D0709B	0.76384	0.07788	0.14135
D0710C	0.60079	0.31340	0.41192
D0711C	0.76378	0.13691	0.2322
D0712C	0.63878	0.28188	0.39115
D0713C	0.62451	0.19245	0.29423
D0714D	0.67969	0.11083	0.19058
D0715D	0.58120	0.18061	0.27558
D0716D	0.61089	0.19123	0.29128
Average	0.62874	0.19789	0.27885

Table 8: ROUGE-2 Human written (DUC-2005)

Doc-Id	Recall	Precision	F-Score
D301	0.11554	0.05650	0.07589
D307	0.12379	0.06471	0.08499
D311	0.13556	0.17274	0.15191
D313	0.22111	0.10552	0.14286
D321	0.17939	0.06729	0.09786
D324	0.07609	0.25529	0.11724
D331	0.30040	0.07011	0.11369
D332	0.16444	0.03862	0.06255
D343	0.15736	0.09580	0.11909
D345	0.07858	0.12686	0.09705
D346	0.09778	0.05198	0.06788
D347	0.13407	0.02856	0.04709
D350	0.11668	0.06156	0.08059
D354	0.07639	0.03218	0.04528
D391	0.12335	0.06924	0.08869
Average	0.14003	0.08646	0.09284

Table 10: ROUGE-2 Human written (DUC-2007)

Doc-Id	Recall	Precision	F-Score
D0701A	0.28151	0.01987	0.03712
D0702A	0.16667	0.02072	0.03685
D0703A	0.17501	0.10553	0.13166
D0705A	0.22944	0.05464	0.08826
D0706B	0.16309	0.03229	0.05390
D0707B	0.12971	0.09810	0.11171
D0708B	0.14815	0.10375	0.12203
D0709B	0.22780	0.02221	0.04047
D0710C	0.18930	0.09504	0.12655
D0711C	0.24473	0.04096	0.07018
D0712C	0.22984	0.09580	0.13523
D0713C	0.21992	0.06463	0.09991
D0714D	0.22449	0.03505	0.06064
D0715D	0.21364	0.06250	0.09671
D0716D	0.23770	0.07073	0.10902
Average	0.20539	0.06145	0.08801