A Hybrid Solution To Abstractive Multi-Document Summarization Using Supervised and Unsupervised Learning

Gaurav Bhagchandani
Information Technology
Sardar Patel Inst. of Technology
Mumbai, India
gpunjabi28@gmail.com

Deep Bodra
Information Technology
Sardar Patel Inst. of Technology
Mumbai, India
deepbodra96@gmail.com

Abhishek Gangan
Information Technology
Sardar Patel Inst. of Technology
Mumbai, India
gangan.abhishek@gmail.com

Nikahat Mulla
Information Technology
Sardar Patel Inst. of Technology
Mumbai, India
nikahat_kazi@spit.ac.in

***Abstract*—In this work, we aim to develop an abstractive summarization system in the multi-document setup. The main challenge in this kind of a system is the identification of redundant information. Our approach hybridizes three components, viz. Clustering, Word Graphs, Neural Networks. In clustering, all the information from multiple documents is divided amongst clusters based on context and importance analysis, such that each cluster possesses sentences of a similar context - Redundancy Identification. Further, Shortest Path Detection in Word Graphs reduces the text. Along with that, we use a sequence to sequence sentence compression and perform paraphrasing using Supervised Recurrent Neural Network to generate an almost completely abstractive summary. The dataset DUC 2004 that was used indicates that the proposed system outperforms other systems in terms of metrics like ROUGE[1] and BLEU[2].**

Keywords—Abstractive summarization; multi-document; clustering; sentence compression; paraphrasing; neural networks; natural language processing; long short-term memory cell (LSTM); redundancy detection; ROUGE; BLEU

I. INTRODUCTION

The internet has become a hub of information and every single day, the articles on the internet are rising at a rapid pace, and so does the associated redundancy. It is necessary to read all these articles to get a complete insight into the topic, which is a time-consuming process.

It is essential to find a solution to reduce the information into meaningful and manageable summaries. These summaries will be expected to focus on the salient details, while reducing the redundancy.

We propose to use neural networks and Natural Language Processing techniques to develop an efficient and accurate tool to summarize multiple documents pertaining to one topic, into a single concise piece of abstractive summary.

Text Summarization is a way of converting a given large block of information into a smaller version preserving its content, overall semantics, meaning and purpose. Manual summarization of articles is a very difficult task.

The two main ways of automatic summarization are: extraction and abstraction. Extractive methods of summarization try to identify the important parts of the text and only extract the whole sentences from the original documents. On the other hand, abstractive summarization methods create summaries in a more humanlike way. They use advanced natural language techniques and interpret the text, to generate a summary that uses a new set of sentences altogether.

In this paper, we have covered abstractive summarization using an unsupervised graph-based approach to convert an entire document into a word-graph and then traverse it to figure out valuable patterns and form sentences using the most important parts of the document. Word graphs are generated for documents that are clustered based on their contexts and importance analysis. The findings and inferences achieved from our work in extractive summarization are briefly discussed and further used effectively in the actual innovation in the form of an abstractive multi-document summarizer. An actual abstractive summarizer would have a natural understanding of how a language works and can generate sentences like humans do.

Finally, we evaluate our results using ROUGE[1] metric which is Recall-Oriented Understudy for Gisting Evaluation and BLEU[2] metric which is short for Bilingual Evaluation Understudy. These metrics quantitatively describe how accurately a system generated summary reflects the main contents of all documents.

II. RELATED WORK

Profound work has already been done in this field of text summarization and during our research, we thoroughly studied a wide range of sources. In the following section, we briefly discuss the inferences and findings from few of the sources that we find most important as far as our proposed solution is considered.

A survey[3] [by Shah et. al., 2016] briefly talks about various different techniques to do multi-document summarization. It proposes an idea of giving equal importance to all the documents in the input suite and then after summarizing each document separately, system can form clusters of all the summaries. Though this technique performs really well as far as the information reduction is concerned, it may encourage redundant ideas in the final cluster.

Another study[4] [Cai et. al., 2013] includes a suggestion of using document bi-type graphs, while ranking within and across document clusters. Then while looking at all the information as a whole, rather than separate pieces, a summary is generated.

The methodology[5] [suggested by Nenkova et. al., 2005] is mainly based on the probability distribution of occurring for different words in a corpus. Sentences are ranked based upon the weights assigned to the individual words present in that particular sentence. It believes in the idea that the importance of a sentence is directly proportional to the ranking of those sentences based on the frequency of occurrences, where sentences containing high probability words are considered more important. It successfully creates a summary and achieves content reduction, while maintaining information inclusion, but it does not identify and understand the context of sentences present in the articles.

Another approach[6] [proposed in the study by Christensen et. al., 2014] focuses on creating levels and hierarchy of cluster, wherein the different main ideas from all the documents are clustered separately, and then these clusters are divided into children clusters according to the sub-ideas present under the main idea. Finally, the summarization is based on giving importance to different levels in the generated hierarchy. This approach is slightly difficult to achieve, but it provides better insights about the topic and the objective function associated holds up the inter-cluster and intra-cluster coherence.

An extension to one of the above methods[7] [study by Vanderwende et. al.] proposes a SumBasic algorithm where various shortened forms of a sentence are used. This algorithm recognizes redundancy and analyzes the articles to understand which sentences will be the best for the summary. Summary quality is improved by duplication removal and reranking. This approach not only ensures that more data will be present in the same summary length, but it also handles more context-sensitive summaries, owing to its feature called 'wordWeight'.

The proposed solution[8] [in a survey by Thanh Le et. al., 2013] first creates a word graph and then uses two stages, namely Sentence Reduction and Sentence Combination. Though it creates an abstractive summary, this approach is bounded by a few constraints. It requires an extractive summary as an input and the final output summary does not score well on ROUGE metric evaluation.

The abstractive approaches of summarization[9] [given by Banerjee et. al., 2016] focus on creating new sentences from existing ones, where the output summary is more human-like summaries. The method mentioned in this paper first selects, from the set of documents, the document that has the highest importance, and after performing sentence clustering, it produces a word graph, based on which an abstractive summary is generated.

Phrase Selection and Merging approach[10] [suggested by Bing et. al., 2015] firstly decomposes sentences into its component noun phrases and verb phrases. Specific costs are associated with these phrases, and along with that, certain constraints such as Noun Phrase validity, Verb Phrase legality, Short Sentence Avoidance, Sentence Number are enforced. Finally, with all the costs and associated constraints, the summarization is formulated as a optimization problem. Post-Processing is done on the final result. The results are very impressive as far as the content reduction is considered, but the grammar quality of the summary may just get compromised.

III. METHODOLOGY

The main challenge in summarization is identification of redundant information and getting rid of it. Many of the existing approaches (as discussed in previous section) of summarization however, take the highest ranked sentences without redundancy detection. Further, reduction of textual data is usually done using sentence compression. These techniques end up using words from the existing vocabulary of input documents, and hence do not produce true abstractive summaries. We address this problem using a hybridized solution of traditional algorithms along with clustering and neural network. Fig. 1. shows stepwise approach that we propose.

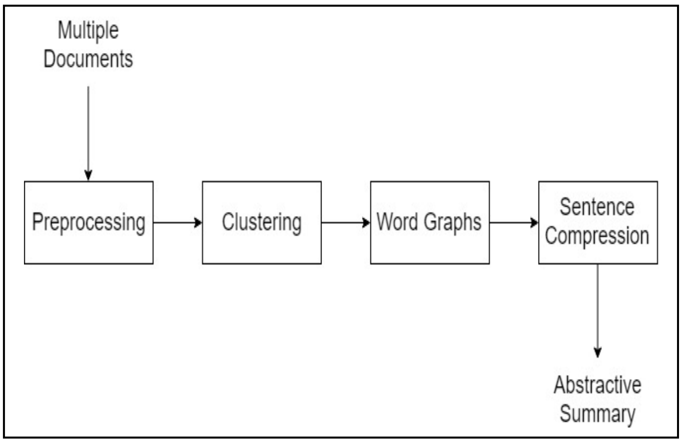

Fig. 1. Block diagram of proposed solution

A. Pre-Processing

The standard pre-processing is conducted on the input files, which includes normalization of passages by converting to lowercase and removal of stop words. Stop words are the words that are used in a language so frequently that they carry least information and hence become least important. This is followed by tokenization of the files into sentences and then into words. All words are lemmatized, rather than stemmed. In Stemming, we reduce words to their root forms, so that words with similar origins are considered the same. However, the actual stem might not be a real word, which causes complications in Word2Vec[11] conversion. To remedy this, we use Lemmatization instead, which reduces words to lemmas, which are actual words and are therefore easier to use with Word2Vec. The final output of the pre-processing module is a single list of strings, containing all the pre-processed sentences of the input documents.

B. *Summarization*

The input to the system consists of multiple documents pertaining to a specific topic. Firstly, we treat all the information as a single document and then divide it into new divisions using clustering.

Clustering[4][12] is a technique that uses unsupervised machine learning to combine a set of objects into groups such that the objects that are in the same group will have similar characteristics. On the other hand, objects from different groups will have highly dissimilar characteristics. Clustering on textual data requires the text to be converted into an equivalent mathematical form, which accurately represents the text. To do this, a machine learning model called Word2Vec is used to generate Word Embedding which are mathematical representations of each word of the vocabulary. Word2Vec converts a word to an n-dimensional vector of real numbers. We have used 2 methods of clustering, viz. K-Means and Hierarchical Clustering.

All these clusters make the input for Word Graph Compression[13][14]. Each cluster can be reduced to a single sentence which contains most of the information. This decreases ideological redundancy, because sentences with similar ideas are present in every individual cluster. The effectiveness of this module is heavily based on the type of clustering done. In the case of K-Means Clustering, it is dependent on the number of clusters created, whereas in Hierarchical clustering, it depends on the similarity value selected. Once all clusters are condensed into sentences, TextRank Algorithm[15] is used to rank the sentences.

TextRank algorithm is similar to PageRank algorithm which is used to rank web pages for search engines. TextRank is an unsupervised and extractive text summarization technique. First, the text is separated into individual sentences, and word embeddings or vector representations are calculated for each of the sentences. A similarity matrix is created that stores the similarities between these sentence vectors, calculated using any of the standard similarity measures. Finally, to calculate the ranking of sentences, this similarity matrix is converted into a graph that has sentences as its nodes and calculated similarity values on its edges. Based on the compression required, the top N sentences are chosen and passed onto the next module to be compressed using neural networks. Neural Network model, as shown in the Fig. 2., performs compression and paraphrasing to generate the final abstractive summary.

Sentence Compression is used to generate a short grammatically correct sentence from a given input sentence such that it contains sufficient important information. Sentence compression is done for each selected sentence using a Sequence to Sequence Encoder-Decoder model. Every word in the input sentence is represented as a vector of fixed dimension using the embedding models and is fed into the input layer of the network. The input layer is connected to an encoder which contains Bi-directional Long Short-Term Memory[16] cells that helps the encoder to understand the context better. As a result, the input is processed forwards as well as backwards. Finally, both the outputs are concatenated and the resultant is sent to the next layer. Context vector that we get as an output from the encoder network, represents the entire input which the decoder network processes in order to compress the sentence. Next word of the sentence is predicted using a Beam Search Decoder, having 'w' as its beam width and takes into account - 'w' probable words.

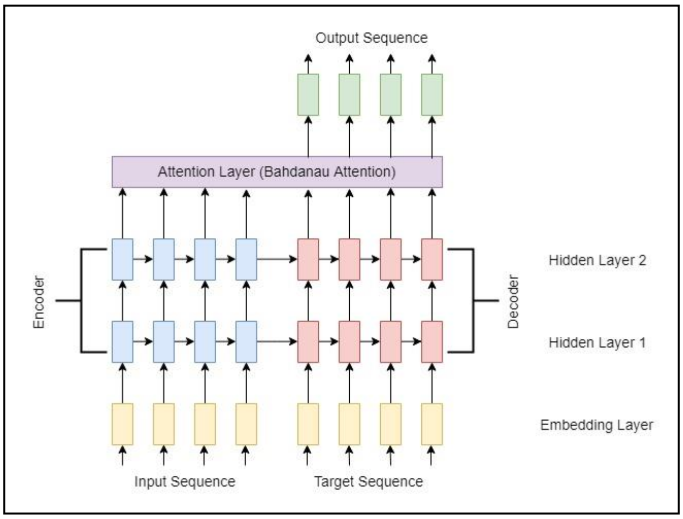

Fig. 2. Architecture of neural network

We use an attention mechanism to enable the neural network to focus on a subset of input features while generating the output sequence. To prevent the network from overfitting, dropout layers are also added.

We use GloVe (Global Vectors) embeddings to vectorize the input words into a fixed dimension of 1×300. The encoder and decoder networks are 2 layers deep each, with each layer containing 150 Long Short-Term Memory cells. The beam width of the Beam Search Decoder was set to 10. Bahdanau attention[17] mechanism was used, in which 20% of the cells were dropped out in the dropout layer. The learning rate, batch size and number of epochs were assigned the values, 0.001, 32 and 25, respectively. The dataset used for training is the sentence compression dataset by google[18] containing 200,000 sentences out of which 100,000 were used.

IV. RESULTS

The following sections discuss the experimental results we obtained at each stage in our model.

A. *Sentence Compression*

Sentence compression is the core component in our system and results obtained from this module decide the performance of our whole approach.

The model was trained on Google Colab for a total of 25 epochs, with each epoch taking around 4 hours.

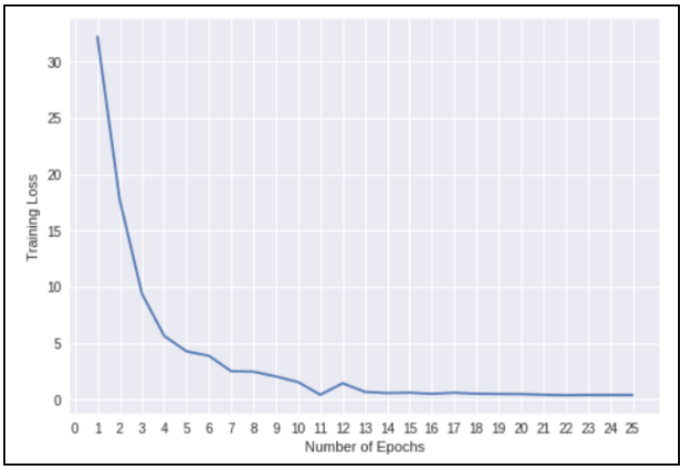

Fig. 3. Variation in Training Loss over the 25 Epochs of training

As seen in the Fig. 3, the training loss decreases rapidly in the earlier stages of training but keeps fluctuating between 0 to 2 in the later stages.

Table I shows some examples of output sentences generated by the module with input sentences taken from Google's dataset for sentence compression[15].

TABLE I. RESULTS OF SENTENCE COMPRESSION

Input Sentence	Compressed Sentence
Jamie McMurray, in his first race back with Ganassi Racing, won the Daytona 500	McMurray wins Daytona 500
The head of the nation's largest public employee union plans to step down next year, setting up a heated contest to guide a political powerhouse that has been one of the biggest spenders in Democratic campaigns	The head of our largest public employee union plans to step down next year
President-elect Barack Obama plans to meet with vanquished Republican rival John McCain on Monday in Chicago, his transition office announced Friday	Obama to meet with McCain on Monday
China Power Investment Corporation, one of the country's five power giants, will be investing 2.8bn yuan in a wind power plant, the largest of its kind in the northeastern Liaoning Province	China Power investment Corporation to invest 2 # 8bn yuan power plant
Junior police officers in Nyanza have been ordered to surrender all small arms assigned to them	Junior police officers ordered to surrender all arms

The output sentences are compressed, meaningful, grammatically correct, and carry correct context.

B. ROUGE and BLEU Scores

The results of the hybrid implementations clearly suggest that there are improvements in BLEU and ROUGE scores in case of combined hybrid systems, rather than the individual modules. Especially, the notable improvement in the traditional TextRank algorithm when preceded by one or more clustering techniques, was a significant finding of our research. BLEU scores are calculated with 60% weight for 1-gram words and 40% weight for 2-gram words.

Using the parameters as, (1) Number of clusters for k-mean clustering (k) = 20, (2) Expected number of sentences in the final summary = 14, and (3) Similarity measure = 8 which is used for flattening of the dendrograms in case of hierarchical clustering, the metric scores for various hybrid systems we developed and tested are shown in table II.

TABLE II. ROUGE AND BLEU SCORES FOR VARIOUS TYPES OF HYBRID SOLUTIONS

TYPE	BLEU	ROUGE		
		ROUGE - 1	*ROUGE - 2*	*ROUGE - L*
TextRank	0.1908	0.2633	0.0629	0.1718
Hierarchical + Word-Graph + Neural Network	0.1578	0.2097	0.0226	0.1598
Hierarchical+ Word-Graph	0.2943	0.2715	0.0508	0.2301
K-Means + Word-Graph + Neural Network	0.1612	0.2312	0.0215	0.1727
K-Means + Word-Graph	0.3383	0.2894	0.0544	0.2518

The BLEU, ROUGE-1 and ROUGE-2 scores have been found highest when Hierarchical Clustering and Word-Graph are used in combination whereas the ROUGE-L score is highest when K-Means Clustering is used instead, but the difference is negligible. When the output of the Word-Graph is passed to the neural network, there is a significant decrease in the scores mainly because of the limitations associated with the LSTM network (discussed in the next section). Another reason is that the ROUGE and BLEU metrics are sensitive to n-grams. But most of the sentences are coherent and grammatically correct.

We were able to identify the proposed approach to be faster than any other multi-document abstractive summarization technique, with a reasonable degree of accuracy, precision and data reduction.

Change in the parameters like, number of clusters (k), number of sentences in the final summary, and similarity measure, may bring about deviations in the scores. Since these metrics were specifically made only for extractive summarization, the scores we end up with for the fully hybridized abstractive systems may or may not convey the actual capabilities of our proposal. According to the subjective analysis and human testing though, the proposed system's performance is remarkable.

V. LIMITATIONS AND FUTURE WORK

There are few limitations associated with LSTM neural networks. They tend to replace some words in the input sentence with a different word. For example, words 'California' and 'Florida' are often interchanged as they are present very close to each other in the GloVe Space i.e. their word vectors are too close to each other, and as a result, the decoder is unable to distinguish them. Their word vectors are close as they are two states of the same country.

Another problem may be observed with numerical values, where a numerical value in an article, for example, number of deaths, percentage tax values, stock values, etc. may be replaced by another value. Reason for this forced and unwanted replacement is same as mentioned above in the example of 'Florida' and 'California'.

An effort can be made in future to get rid of the limitations foreseen. The Bidirectional LSTM network can be replaced by Pointer Generated Networks[19]. A specialized training model can be developed to make the system understand the proper nouns, names of people, words specifically belonging to the news articles, peculiar numeric values, quotes, etc. This model would have an ability to understand all these components through the context they come in and the model would be able to assign them with unusually large weights. As a result, there won't be any changes made to these set of words.

REFERENCES

[1] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In Text Summarization Branches Out: Proceedings of the ACL-04 Workshop, pages 74–81.

[2] Papineni, K., S. Roukos, T. Ward, and W.-J. Zhu. 2001. BLEU: A method for automatic evaluation of machine translation. IBM Research Report RC22176 (W0109-022).

[3] Chintan Shah, Anjali G. Jivani, "Literature Study on Multi-document Text Summarization Techniques", 2016.

[4] Xiaoyan Cai, Wenjie Li, "Ranking Through Clustering: An Integrated Approach to Multi-Document Summarization", IEEE Transactions On Audio, Speech, And Language Processing, 2013.

[5] Ani Nenkova, Lucy Vanderwende , "SumBasic: The Impact of Frequency on Summarization", 2005.

[6] Janara Christensen, Stephen Soderland, Gagan Bansal, Mausam, "Hierarchical Summarization: Scaling Up Multi-Document Summarization", Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, 2014.

[7] Lucy Vanderwende, Ani Nenkova, Hisami Suzuki, Chris Brockett, "Beyond SumBasic: Task-Focused Summarization with Sentence Simplification and Lexical Expansion".

[8] Huong Thanh Le, Tien Manh Le, "An approach to abstractive text summarization", 2013 International Conference on Soft Computing and Pattern Recognition (SoCPaR).

[9] Siddhartha Banerjee, Kazunari Sugiyama, "Multi-Document Abstractive Summarization Using ILP based Multi-Sentence Compression", IEEE Transactions On Computation & Languages, 2016.

[10] Lidong Bing, Piji Li, Yi Liao, Wai Lam, Weiwei Guo, Rebecca J. Passonneau, "Abstractive Multi-Document Summarization via Phrase Selection and Merging, 2015", Computation and Language & Artificial Intelligence.

[11] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed Representations of Words and Phrases and their Compositionality. Accepted to NIPS 2013.

[12] Janara Christensen, Stephen Soderland, Gagan Bansal, Mausam, "Hierarchical Summarization: Scaling Up Multi-Document Summarization", Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, 2014.

[13] Katja Filippova, Multi-Sentence Compression: Finding Shortest Paths in Word Graphs, Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010).

[14] Florian Boudin and Emmanuel Morin, Keyphrase Extraction for N-best Reranking in Multi-Sentence Compression, Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2013), 2013.

[15] Rada Mihalcea and Paul Tarau, "TextRank: Bringing Order into Texts", Conference on Empirical Methods in Natural Language Processing, 2004.

[16] Alex Graves, Abdel-rahman Mohamed, Geoffrey Hinton, "Speech Recognition With Deep Recurrent Neural Networks", Cornell University, 2013.

[17] Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate", Natural Language Processing Conference, ICLR, 2015.

[18] Large corpus of uncompressed and compressed sentences from news articles, https://github.com/google-research-datasets/sentence-compression.

[19] Abigail See, Peter J. Liu, and Christopher D. Manning. Get to the point: Summarization with pointer-generator networks. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1073–1083, July 2017.