

TEXT SUMMARIZATION FROM MULTIPLE DOCUMENTS

Pranav Pradeep¹, Aayush Shetty², Akshit Shetty³, Prof. Kranti Bade⁴

^{1,2,3}Student, Dept. of Computer Engineering, SIESGST, Nerul, Maharashtra, India

⁴Assistant Professor, Dept. of Computer Engineering, SIESGST, Nerul, Maharashtra, India

Abstract - In this modern age there are a lot of information available on the internet. In fact, the International Data Corporation (IDC) projects that the total amount of digital data circulating annually around the world would sprout from 4.4 zettabytes in 2013 to hit 180 zettabytes in 2025. There is an enormous amount of textual material, and it is only growing every single day. Think of the internet, consisting of web pages, news articles, status updates, blogs and so much more. The data is unstructured and the best that we can do to navigate it is to use search and skim the results. There is a great need to reduce much of this text data to shorter, focused summaries that capture the salient details, both so we can navigate it more effectively as well as check whether the larger documents contain the information that we are looking for. With such a big amount of data circulating in the digital space, there is a need to develop machine learning algorithms that can automatically shorten longer texts and deliver accurate summaries that can fluently pass the intended messages. This will help the user to navigate through and retrieve information from a large information in a short span and thus save a lot of time. Furthermore, applying text summarization accelerates the process of researching for information, and increases the amount of information that can fit in an area.

Key Words: RNN, RBM, TF-IDF, Clustering, Feature Matrix, Text Summarization.

1. INTRODUCTION

Text summarization refers to the practice of abbreviating long pieces of text. The goal is to establish a coherent and fluent overview with only the key points outlined in the text. Automatic text summarization methods are required to tackle the ever-growing amount of text data accessible through internet, both to assist you discover relevant information faster and to ingest relevant information quickly. For this sort of job, we (humans) are usually good at this, as it involves first understanding the meaning of the source text and then distilling the meaning and collecting outstanding information in the current definition.

As such, the aim of automatically generating text summaries is to make the resulting summaries as good as human-written ones. The system generated text summarization guarantees that important information is not lost and that the summarized text should not contain any redundancy depending on the algorithm used and how well it was trained to handle certain scenarios. Text summarization saves a lot of time and makes it easy for searching and analyzing.

We will be summarizing the information from multiple documents that are similar and produce summarized text. There are two main types of approaches Extraction-based summarization and Abstraction-based summarization. In our case, we have used Extractive summarization using RBM (restricted boltzmann machine) to produce proper text summary.

2. LITERATURE REVIEW

In the year 2018; Peter Liu, Mohammad Saleh and Ben Goodrich proposed a multi-document summarization of source documents [5]. They used extractive approach to roughly identify salient features and a neural abstractive model to generate the summary. They introduced a decoder-only architecture to handle long sequences which is much longer than typical encode-decoder architectures. The model can generate continuous, coherent multi-sequence paragraphs and even whole wikipedia articles. When the model is provided with reference documents, it can extract relevant information by reviewing perplexity, rouge scores and human evaluations.

In the year 2019, Yang Liu employed a pre-trained transformer model i.e. BERT to achieve good performance on multiple NLP tasks [6]. He described BERTSUM, a simple variant of BERT, to achieve extractive summarization. The system was applied on CNN/Daily mail dataset and it outperformed the previous best-performed system i.e. BERT by 1.65 on ROUGE-L. In this paper, BERT was pre-trained on a huge dataset and the powerful architecture that can learn complex features and extract them. It can further boost the performance of extractive summarization.

In 2018, Derek Miller proposed BERT model for extractive text summarization on lectures for collecting key phrases and sentences that best represent the context [7]. The system utilizes the BERT model for text embeddings and uses K-Means clustering to identify similar or closest sentences for summary selection. It can save student's valuable time by summarizing lecture content, based on their desired number of sentences. The results of the system that utilizes BERT for extractive summarization were promising, there were still some areas where the model struggled, which shows further improvement in the near future.

In August 2016, Ramesh Nallapati, Bowen Zhou, Cicero dos Santos and Bing Xiang proposed an abstractive text summarization model using Attentional Encoder-Decoder RNN [8]. They showed several novel models that address critical problems in summarization such as modeling key-words, emitting words that are rare at training time and capturing the hierarchy of sentence-to-word structure. The proposed models contribute to further improvement in performance. The system was tested on a custom made dataset consisting of multi-sentence summaries and performance benchmark is set for further research.

In April 2017, Pierpaolo Basile, Gaetano Rossiello and Giovanni Semeraro proposed a centroid-based method for text summarization to overcome the issue of bag-of-words representation that does not allow to get hold of the semantic relationships between concepts that compare strongly related sentences with no words in common [9]. This method exploits the compositional capabilities of word embeddings. The system was evaluated on multi-document and multilingual datasets which proved that continuous vector representation was more effective compared to the bag-of-words model. This method achieves good performance when compared to more complex learning models. The method is simple, unsupervised and can be adopted in other summarization tasks.

In March 2016, Sandeep Sripada, Venu Gopal Kasturi and Gautam Kumar Parai presented three techniques for obtaining extraction-based summaries which includes a novel graph based formulation for improvement on the former methods [10]. The first approach generates the importance of a sentence using score calculator based on various semantic features and a semantic similarity score to select sentences that would represent the document. The algorithm used is

stack-decoder algorithm which is used as a template and produces summaries that are closer to optimal. The second approach generates clusters of sentences based on the above semantic similarity score and a representative is selected from each cluster that are to be included in the generated summary. The third and final approach is a novel graph problem based formulation which generates summaries based on the cliques found in the constructed graph. The graph is generated by building edges between sentences that has similar topics but are semantically not similar.

In April 2017, an approach involving neural seq-to-seq model that provide a new approach to abstractive text summarization [12]. In this paper they proposed a novel architecture that augments the standard seq-to-seq attentional model in two orthogonal ways. First, it uses a hybrid pointer generator network that can copy words from source text which will aid in reproducing accurate information while retaining novel words through generator. Second, it uses coverage to keep track if it has summarized properly which discourages repetition. The model was tested on CNN/Daily Mail dataset and outperformed the abstractive state-of-the-art by at least 2 ROUGE points.

In January 2019, Sukriti Verma and Vagisha Nidhi described a text summarization using deep learning model [13]. The method was divided into three phases: feature extraction, feature enhancement and summary generation. All these phases work together to uptake core information and generate coherent and understandable summary. The system uses Restricted Boltzmann Machine (RBM) which enhances and abstract the salient features to improve resultant summary without losing any important information. All sentences are given a score based on those enhanced features and using that an extractive summary is constructed.

3. SYSTEM ARCHITECTURE

The system uses RBM (Restricted Boltzmann Machine), a stochastic and generative neural network which is capable of learning internal representations through probability distribution over its set of inputs [4]. RBMs are non-deterministic deep learning models with only two types of nodes visible and hidden nodes. There are no output nodes so that gives them this non-deterministic feature. They learn patterns without the capability of the typical 0 or 1 type output through which patterns are learned and optimized. RBM is used

to generate the desired summary from multiple documents.

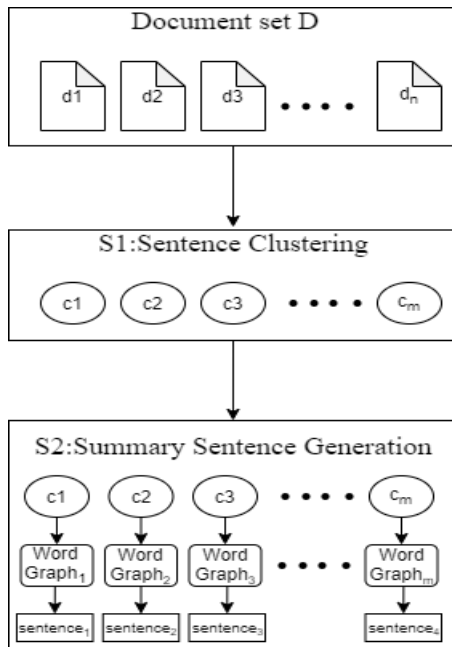


Fig 3.1: Flowchart of Project

Before that multiple documents from which the summary has to be generated are given as inputs to the system. In order to get a summarized document which should be meaningful (the context should be consistent), we have to choose the texts from the multiple input document which share the same idea/topic. For this we used k-means clustering. K-means clustering is a type of unsupervised learning, which is used when you have unlabeled data (i.e., data without defined categories or groups). After K-means clustering is done on the documents it generates a combined document which is given as input to the RBM machine to generate the summary.

4. DETAILED WORKING

In this project, we use K-means clustering and Restricted Boltzmann machine(RBM) technique to summarize multiple documents. Through k-means we can group/cluster the documents based on similar topics correctly. The cluster thus obtained is then summarized through the extractive method of summarization using Restricted Boltzmann machine technique.

4.1 Clustering of Multiple Documents using K-means

When Multiple Documents have to be summarized, then the input to the system are in the form of multiple documents and output is a single summarized document. In order for the Summarized Document obtained to be meaningful(the context should be consistent), we have to choose the texts from the multiple input document which share the same idea/topic. For this we used k-means clustering. K-means clustering is a type of unsupervised learning, which is used when you have data without defined categories or groups. This algorithm finds groups in the data, with the number of groups represented by the variable K(ie.no. of clusters). The algorithm works iteratively to assign each data point to one of K groups based on the features that are provided. Data points are clustered based on feature similarity.

The k-means process starts with vectorizing the sentence i.e converting sentences into numeric representations. Here we have used paragraphs instead of sentence because a paragraph is a series of related sentences developing a central idea and also it will be easier to obtain the document which shares the same topic. For vectorizing we used Scikit-learn's Count Vectorizer which just counts the no. of occurrence of the individual words in the documents. It also enables the pre-processing of text data prior to generating the vector representation. Then the clustering is done using the Scikit-learn's k-means clustering which basically plots all of the numbers on a graph and grabs the ones that group together and this is done by calculating cosine distance between each document as a measure of similarity. In the below example, there are 12 paragraphs obtained from various documents which are grouped into 5 clusters(0-4 are the cluster names).For example, the 1st sentence belongs to cluster 3 , 2nd sentence belongs to cluster 4 and so on. Then the cluster which contains the maximum no. of paragraph is selected and all those paragraphs are merged into one document. Thus a single document is obtained with texts that share similar idea.

4.2 Extractive Summarization

Our approach of Extractive summarization is: RBM (Restricted Boltzmann Machine), a stochastic and generative neural network which is capable of learning internal representations through probability

distribution over its set of inputs [4]. They are a two-layered artificial neural network, the first layer is called the visible layer or the input layer and other one is called the hidden layer.

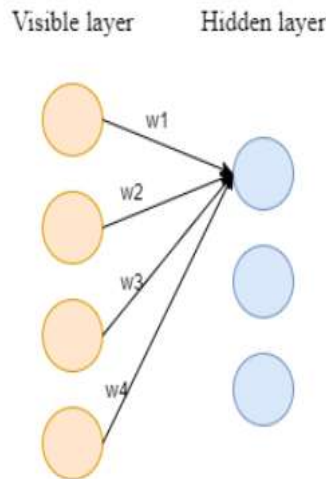


Fig 4.1: RBM Layers

Each layer contains a neuron-like unit called nodes, and the calculations take place at nodes. The nodes are connected to each other across layers, but no two nodes of the same layer are linked. The inputs are fed through the visible layer and then passed on to the nodes of the hidden layer with the respective weights of each node of the visible layer. Then the inputs are multiplied with the respective weights and added to a so-called bias. The result of these operation are fed into an activation function, which produces the nodes output.

After this, the Restricted Boltzmann machine learns to reconstruct data by themselves in an unsupervised manner. This is done by reversing the above process i.e. the hidden layer becomes the input layer with activations as the new input. Then these activations are again multiplied with previous weights that are associated with the visible layer nodes and these products are added to the visible layer bias at each visible node. Thus obtained results are called the reconstructions which are then compared to the original input. Since the weights are randomly initialized, therefore the reconstruction and the original input largely differ from each other. The obtained values are fed to the system in an iterative manner until the difference is minimized. The process of reconstruction is, in a sense, learning which group of feature(text) tends to occur repeatedly for a given set of input. Thus choosing only those texts for the final

output. The input is in the form of Single Text Document, which then goes under pre-processing.

Pre-processing:

- Stemming-It is a process of reducing a word to its stem/base/root word. For example-processing is converted to process, likes to like etc.
- Part of speech Tagging-POS tagging is the process of marking up a word in a corpus to a corresponding part of a speech tag, based on its context and definition.
- Stop word filtering-the words which are useless or do not have any meaning of their own. Words such as a,an,the etc are filtered out here.
- Punctuation marks removal-any punctuations (“,”,”.”;”etc) are also removed.

Feature matrix generation:

After the Preprocessing is done, the document is then structured into a matrix. A Sentence matrix is generated of order $m \times n$, where m is the no. of sentence and n is the no. of features of each each sentence. The various feature of a sentence are-

- Title similarity-a sentence in the document is said to be important for the summary if it is similar to title of the document [11]. The similarity is calculated as the ratio of no. of words which are common in both the sentence and as well as in the title to the total no. of words in the title
- Sentence position- a sentence in the document is also scored based on its position in the document i.e if its appears at the starting 20 percent or at the last 20 percent part of the document it is scored 1 else 0; because the sentence that appear at starting or at the end are considered to be important.
- Term weight-it means the occurrence and occurrence of a word in the whole document. This is calculated through Tf-IDF(Term frequency inverse document frequency).
- Sentence length-sentences which are too short or too large are discarded as short sentences may not give any useful information and large sentences may contain too much unnecessary information.
- Proper noun score-it is count of the total proper nouns present in the document as proper nouns give the information about what or whom the author is referring to.
- Number of thematic words-The 10 most frequently occurring words of the text are found. These are thematic words [13]. For each sentence, the ratio of no. of thematic words to total words is calculated[13].

- Sentence position relative to the paragraph-as per the observation the starting and the ending sentence of a paragraph usually represents the beginning of a new discussion and the conclusion respectively. Therefore is considered to be important. If it is a starting or an ending sentence in the paragraph it is assigned as 1 else 0.
- No. of Numerals-Since the numerals in a document represents facts, it is crucial to have sentences which have certain figures. It is calculated by dividing the the number of thematic words by the total no. of words in the sentence.

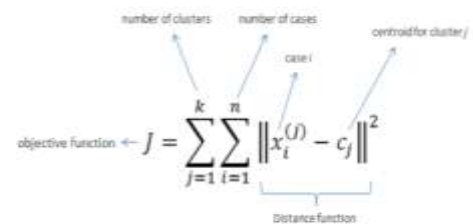
The above features are calculated for each sentence and are stored in the form of a matrix called the Sentence matrix with each sentence having the feature vector value. This Sentence matrix is given as the input to the visible layer of the Restricted Boltzmann machine. Here we have one visible and one hidden layer. We use Persistent Contrastive Divergence method to sample during the learning process [2]. We have trained the RBM for 5 epochs with a batch size of 4 and 4 parallel Gibbs Chains, used for sampling using Persistent CD method [11]. Each sentence feature vector is passed through the hidden layer in which feature vector values for each sentence are multiplied by learned weights and a bias value is added to all the feature vector values which is also learned by the RBM [13]. After this an enhanced matrix is obtained which are summed together to obtain the score which are then arranged in descending order of value such that the 1st sentence is the most relevant/important sentence and is added to the final document which will be the summary. The following sentences for the summary are selected which have the highest Jaccard similarity with respect to the 1st sentence and are added to the final summary where they are rearranged in the order that they had appeared in the original document.

5. ALGORITHM

5.1 K-means

K-Means Clustering is an unsupervised machine learning algorithm. In contrast to traditional supervised machine learning algorithms, K-Means attempts to classify data without having first been trained with labeled data [3]. Here k is the no. of cluster centers called as centroids i.e it is the no. of clusters to be formed.

Algorithm first works by selecting k random points in the pool of data points. Then these data points are assigned to the closest cluster based on the distance from each centroid. After this a new centroid is determined by calculating the average of the data points of each cluster. Then the data points are again rearranged based on their closeness to the centroids. This process is repetitive until no further rearrangements are possible.



5.2 Restricted Boltzmann Machine

A Restricted Boltzmann machine is an algorithm useful for dimensionality reduction, classification, regression, collaborative filtering, feature learning and topic modeling [1]. They are a two layered neural network namely the visible layer and the hidden layer. The sentence-feature matrix is given as the input to the visible layer.

Let S be the set of sentences.

$$S = s_1, s_2, s_3, \dots, s_n$$

Where $s_i = f_1, f_2, f_3, \dots, f_m, i_j = n$

Where n is the no. of sentences and m is the no. of features. These inputs are multiplied with respective weights of the nodes of visible node and are passed to the hidden node. Here we have used one hidden layer and for it a set of bias value is selected which are randomly generated.

$$H_0 = h_1, h_2, h_3, \dots, h_n$$

After the cycle(forward and the reconstruction phase) a new refined matrix is generated which is calculated by:

$$\sum_1^n s_i + h_i$$

6. RESULT AND OUTPUT

6.1 Implementation

In our Project we have made a website where the user can select the Text documents which they want to summarize. After selecting the documents, these documents are stored in the input Database so that the user can see the uploaded files.

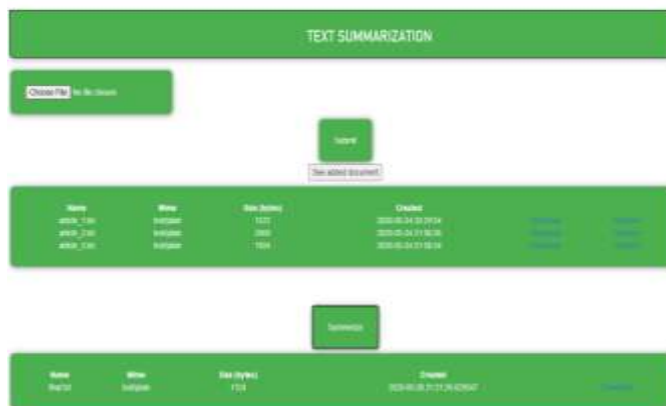


Fig 6.1: User Interface of the System

These documents are extracted from the database and are then divided into paragraphs and these paragraphs are stored in an excel sheet.



Fig 6.2: Excel Sheet to store the Paragraphs

These paragraphs then go under the k-means clustering algorithm, so that the most similar paragraphs are obtained. Here the sentences are vectorized i.e representing them in numeric values and these values are subjected to the clustering algorithm. This process will give the no. of paragraphs in each cluster, which is then arranged in an descending order. The paragraphs present in the first cluster in the order are selected and are append together to form a single document, thus obtaining the texts from different documents which share the same idea. The new text document is then summarized using the Restricted Boltzmann machine algorithm. The text document then goes under pre-processing to remove all the unnecessary information from the machine. Then 5 different feature values are

calculated for all the sentences and these are stored in a matrix called the Sentence-feature matrix of order $m \times n$, where m is the no. of sentences and n is the no. of features. This matrix is then given to the Restricted Boltzmann machine as input which refines the matrix in a repetitive order to give the top most important/relevant sentences for the final summary. The final summary is then upload to the output Database and from here it is displayed in the website where the user can download the summarized file.

6.2 Testing And Results

We had selected 4 text Documents of the same topic since they would have more common content in them. These documents are appended together to form a single document where are then divided paragraph wise. In the clustering process we had assigned no. of clusters as five. The highest no. of paragraphs in a cluster was 6 out of the total of 12 paragraphs. These were appended together to form a new document which is then subjected to summarization process. After the preprocessing the feature matrix was calculated for each sentence.

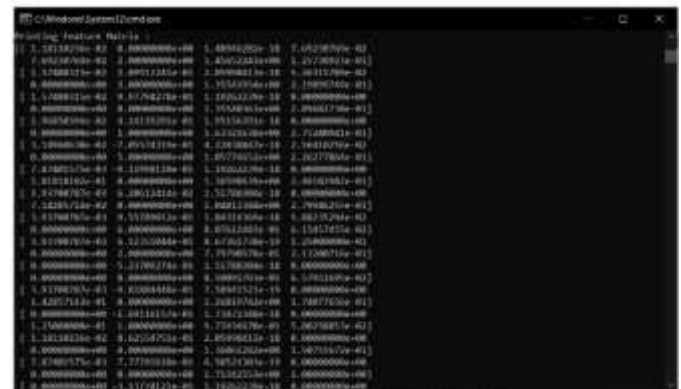


Fig 6.3: Feature Matrix

The RBM that we are using has 9 perceptrons in each layer with a learning rate of 0.1 [4]. Each sentence feature vector is passed through the hidden layer in which feature vector values for each sentence are multiplied by learned weights and a bias value is added to all the feature vector values which is also learned by the RBM [4]. After this a refined and enhanced matrix is obtained.



Fig 6.4: Training Phase

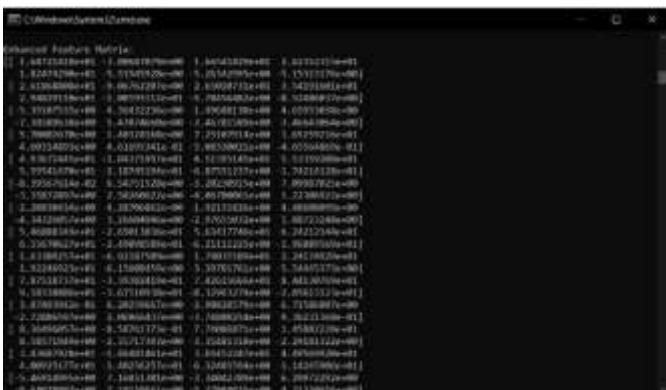


Fig 6.5: Enhanced Feature Matrix

After the summarization process the final document thus obtained was a single-paragraph which had all the Main essence of the documents which were fed as the document. On an average the process took 1 min and 42 secs to complete.

7. CONCLUSION

With the immense increase in the number of information on the internet, there is a need for a system to reduce the human time for retrieving the useful data from them. Automatic Summarization is the solution for the information overload. Summaries helps in analysing the texts i.e extracting the most useful information or for obtaining the general idea of the whole document thus saving the human effort of going through the whole document. Our system is capable of extracting the most relevant information from a no. of documents in a very less time. More no. of Documents will help obtaining better information from them. In the future, with the machine learning algorithms the System could be improved by making the generated summaries more human-like i.e generating their own sentences instead of just extracting the sentences from the document.

8. ACKNOWLEDGEMENT

We would like to express our sincere gratitude to Prof. Kranti Bade for all the valuable guidance and encouragement in carrying out this project. We would like to thank her for constantly giving us feedback and assisting us in completing this project.

REFERENCES

- [1] A beginner's guide to restricted boltzmann machines (rbms). <https://pathmind.com/wiki/restricted-boltzmann-machine>.
- [2] Deep learning tutorials. <http://deeplearning.net/tutorial/>.
- [3] K-means clustering python example. <https://towardsdatascience.com/machine-learning-algorithms-part-9-k-means-example-in-python-f2ad05ed5203>.
- [4] Restricted boltzmann machines — simplified. <https://towardsdatascience.com/restricted-boltzmann-machines-simplified-eab1e5878976>.
- [5] Peter J. Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. "Generating wikipedia by summarizing long sequences", 2018.
- [6] Yang Liu. "Fine-tune bert for extractive summarization", 2019.
- [7] Derek Miller. "Leveraging bert for extractive text summarization on lectures". arXiv preprint arXiv:1906.04165, 2019.
- [8] Ramesh Nallapati, Bowen Zhou, Cicero Nogueira dos santos, Caglar Gulcehre, and Bing Xiang. "Abstractive text summarization using sequence-to-sequence rnns and beyond", 2016.
- [9] Gaetano Rossiello, Pierpaolo Basile, and Giovanni Semeraro. "Centroidbased text summarization through compositionality of word embeddings". In Proceedings of the MultiLing 2017 Workshop on Summarization and Summary Evaluation Across Source Types and Genres, Valencia, Spain, April 2017.

[10] Gautam Kumar Parai Sandeep Venu Gopal Kasturi. "Multi-document extraction based summarization". <https://nlp.stanford.edu/courses/cs224n/2010/reports/ssandeeep-venuk-gkparai.pdf>, 2016.

[11] Gautam Kumar Parai Sandeep Venu Gopal Kasturi. "Text summarization using restricted boltzmann machine: Unsupervised deep learning approach". <http://ijsart.com/Content/PDFDocuments/IJSARTV4I623858.pdf>, 2018.

[12] Abigail See, Peter J. Liu, and Christopher D. Manning. "Get to the point: Summarization with pointer-generator networks", 2017.

[13] Sukriti Verma and Vagisha Nidhi. "Extractive summarization using deep learning", 2017.