Extractive Text Summarization

REPORT BY

Subhashini Ishnarayan Gupta

Roll No: 1901002

FY MTECH COMPUTER ENGINEERING

K. J. Somaiya College of Engineering,
Vidyavihar, Mumbai- 400077.

SEMESTER 2

Introduction

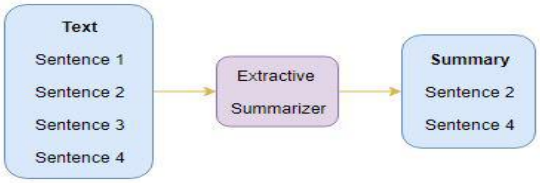
Text Summarization is one of those applications of Natural Language Processing (NLP) which is bound to have a huge impact on our lives. With growing digital media and ever growing publishing – who has the time to go through entire articles / documents / books to decide whether they are useful or not?. Text Summarization is one of the most challenging and interesting problems in the field of Natural Language Processing (NLP). It is a process of generating a concise and meaningful summary of text from multiple text resources such as books, news articles, blog posts, research papers, emails, and tweets.It reduces the time required for reading whole document and also it space problem that is needed for storing large amount of data.The mobile app **inshorts,** It's an innovative news app that converts news articles into a 60-word summary.

There are broadly two different approaches that are used for text summarization:

- Extractive Summarization
- Abstractive Summarization

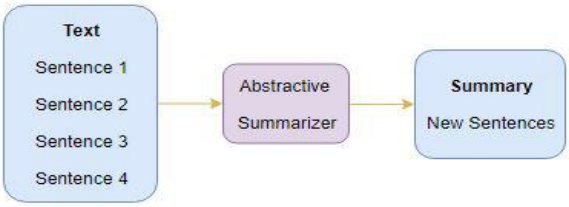
Extractive Summarization

These methods rely on extracting several parts, such as phrases and sentences, from a piece of text and stack them together to create a summary. Therefore, identifying the right sentences for summarization is of utmost importance in an extractive method.
We identify the important sentences or phrases from the original text and extract only those from the text. Those extracted sentences would be our summary.



Abstractive Summarization

This is a very interesting approach. Here, we generate new sentences from the original text. This is in contrast to the extractive approach we saw earlier where we used only the sentences that were present. The sentences generated through abstractive summarization might not be present in the original text:



Paper 1 Summary:-

Extractive Text Summarization using Neural Networks

Literature Survey

A neural network is a network or circuit of neurons, or in a modern sense, an artificial neural network, composed of artificial neurons or nodes.Thus a neural network is either a biological neural network, made up of real biological neurons, or an artificial neural network, for solving artificial intelligence (AI) problems. The connections of the biological neuron are modeled as weights. A positive weight reflects an excitatory connection, while negative values mean inhibitory connections. All inputs are modified by a weight and summed. This activity is referred to as a linear combination. Finally, an activation function controls the amplitude of the output. For example, an acceptable range of output is usually between 0 and 1, or it could be −1 and 1.These artificial networks may be used for predictive modeling, adaptive control and applications where they can be trained via a dataset. Self-learning resulting from experience can occur within networks, which can derive conclusions from a complex and seemingly unrelated set of information.

Approach used

This paper propose a fully data driven approach using neural networks which gives reliable results irrespective of the document type. This does not require predecided features for classifying the sentences. The proposed model is capable of producing summaries corresponding to documents of varying lengths. Here recursive approach is used to produce summaries of variable length documents. The model trained using DUC datasets. The proposed model is evaluated using ROUGE automatic evaluator on DUC 2002 dataset and compare the ROUGE1 and ROUGE2 (two variants of ROUGE) scores with existing models. Experimental results show that the proposed model achieves performance comparable to state-of-the-art systems without any access to linguistic information.

Methodology

The proposed model is based on a neural network which consists of one input layer, one hidden layer, and one output layer.The document is fed to the input layer, computations are carried in the hidden layer and an output is generated at the final layer.Sentences of the document were to be fed as input to the network.Since the input represented in some numerical form. for this,word2vec model was used.A language model is trained on large datasets and each of the words in the vocabulary is assigned a vector of some fixed dimension based on the context in which it appears.These vectors have some important properties (for example closely related words have similar representations) which are more representative of the language.

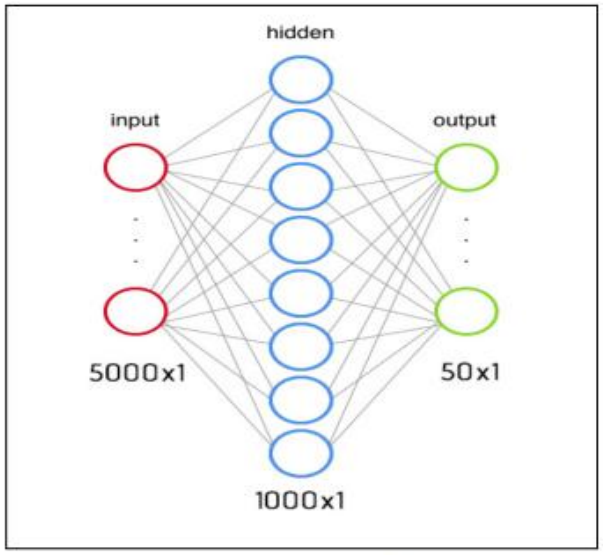


Fig. 1. Proposed Neural Network with 'page_len' = 50.

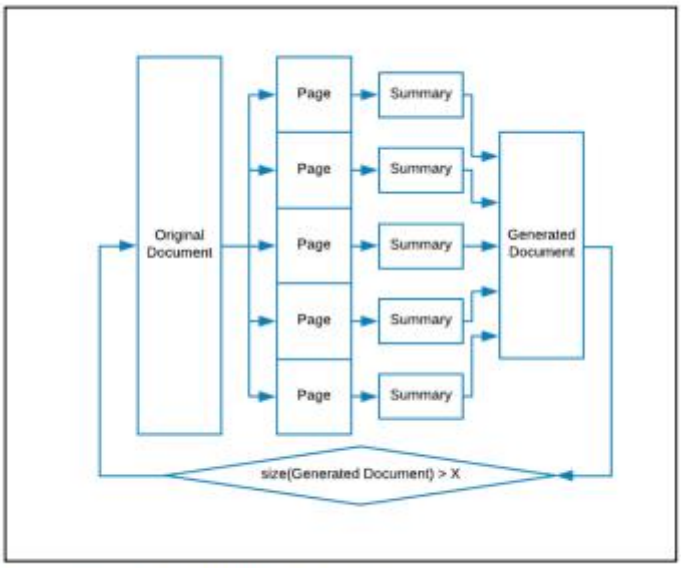


Fig. 2. Flow diagram of the proposed model.

After obtaining the word vectors, vector representation for sentences had to be created. In this model,Fasttext library is used which is provided by Facebook to convert sentences to vectors. The model takes input as sentences of the English language, vector representation of words and converts the sentences to fixed dimension vectors. Every document has different length in terms of the number of sentences and a summarizer should work well for all sizes. Because of this, Recurrent Neural Networks and End to End learning have been used.

Let the number of sentences in the document be 'doc_len'. Now document is divided into segments, each having a fixed number of sentences. Each such segment is called a 'page' and let this fixed number be 'page_len'. In this way we obtain 'num_pg' pages, where 'num_pg' equals to ceil(doc_len/page_len). Thus, for each run of the network sentences of a page are converted into their corresponding vectors (each having 100 entries). All such vectors are concatenated in order to form a page_len*100-dimension vector which is fed to the input layer of the network. For pages with the number of sentences less than 'page_len', the input vector is padded with zeros. Note that 'page_len' is fixed for the model.

A softmax activation function is applied to the output at the last layer.Error/loss from the correct prediction is calculated using cross-entropy between the predicted output and the correct hot vector. This error is then fed back into the network for training. Thus the weights and bias matrices are adjusted in each iteration by back-propagating the error.

For the generation of the summary of a given document, the entire text is broken into pages. The summary length in terms of the number of sentences is fixed and known before summary generation.Using the final output vector, corresponding sentences are picked up from the document and concatenated in order to produce the final summary.

The best way to evaluate this model would be to use ROUGE. It stands for Recall-Oriented Understudy for Evaluation. To evaluate the neural network, ROUGE compares the summaries generated by the network to human-generated summaries. This is the reason why it's used extensively for evaluating automatic summaries and sometimes also for machine translations.

Paper 2 Summary:-

Extractive Text Summarization Using Sentence Ranking

Literature Survey

Extractive text summarization is divided in two phases: 1) Pre-processing 2) Processing.The NLTK module is a massive tool kit, aimed at helping us with the entire Natural Language Processing (NLP) methodology. NLTK will aid us with everything from splitting sentences from paragraphs, splitting up words, recognizing the part of speech of those words, highlighting the main subjects, and then even with helping your machine to understand what the text is all about.NLTK includes graphical demonstrations and sample data.

Ranking in information retrieval is the process of getting documents most relevant to a user query where the documents are ranked according to their degree of relevance, importance, etc. Ranking in terms of information retrieval is an important concept in computer science and is used in many different applications such as search engine queries and recommender systems. A majority of search engines like Google use ranking algorithms to provide users with accurate and relevant results.The main idea behind a ranking algorithm is that given a query containing some keywords, find an optimal order of results (web pages / documents) in a way that the top results meet the user's information need.

Approach

In this proposed approach, the text extractive method is used to get summary of given input.Here .txt file is used as a input.

1) Firstly, the input file is tokenized in order to get tokens of the terms.
2) After tokenisation, the stop words are removed from the file. The words which are remained are considered as a keyword.
3) The keywords are taken as an input for that a part of tag is attached to each keyword.
4) After completing this pre-processing step we are calculating frequency of each keyword like how frequently that key word has occurred from this maximum frequency of the keyword is taken.
5) Now weighted frequency of the word is calculated by dividing frequency of the keywords by maximum frequency of the key words.
6) In this step the sum of weighted frequencies is calculated. Finally, summarizer will extract the high weighted frequency sentences and the extracted sentences are converted into document form.

Methodology

In the extractive summarization, the summarizer takes input as text file and tokenization of an input text is done in-order to find the terms of the text. Then stop words are removed in order to filter the text. And finally, part-of-speech tag is added to each token.

Step 1: After adding the parts-of-speech tag to tokens or terms each individual weight are assigned to the tokens. The term weight is calculated as follows:
Wt= frequency of term/tokenization of terms in document

Step 2: Now maximum weight of the token is considered after finding maximum weight. The weighted frequency of the document is calculated as follows:
Wtf=frequency of a term/ maximum frequency of the term

Step 3: In this step, the frequencies are connecting in place of corresponding words in sentence and sum of it is found. The ranks are found based on the weighted frequency. The sentences are sorted based on their Weighted frequency ranks like highest rank to lowest. The sentences are arranged in descending order.

Step 4: Finally, summarizer will extract sentences which rank is highest form the document and the sentences which are extracted are converted into appropriate summery form.

Fig. 1. Input text

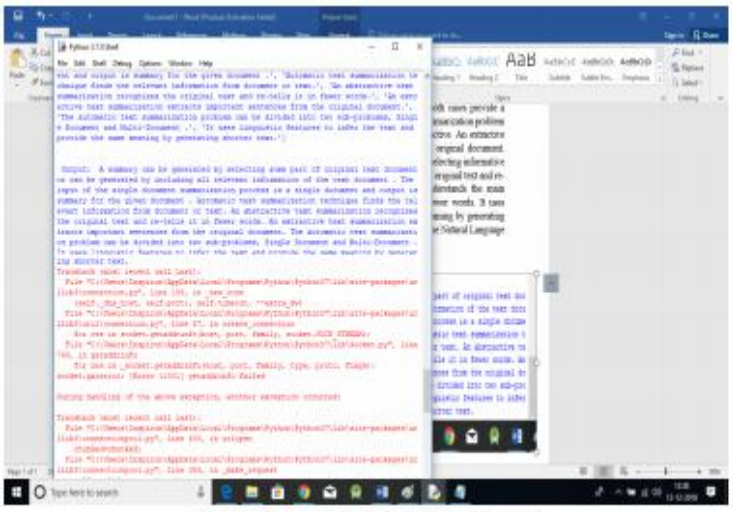


Fig. 2. Output generated by system

Paper 3 Summary:-

Extractive Summarization using Deep Learning

Literature Survey

Most early work on text summarization was focused on technical documents and early studies on summarization aimed at summarizing from pre-given documents without any other requirements, which is usually known as generic summarization.A restricted Boltzmann machine (RBM) is a generative stochastic artificial neural network that can learn a probability distribution over its set of inputs.RBMs are a variant of Boltzmann machines, with the restriction that their neurons must form a bipartite graph: a pair of nodes from each of the two groups of units may have a symmetric connection between them; and there are no connections between nodes within a group."unrestricted" Boltzmann machines may have connections between hidden units. This restriction allows for more efficient training algorithms than are available for the general class of Boltzmann machines, in particular the gradient-based contrastive divergence algorithm. Restricted Boltzmann machines can also be used in deep learning networks. In particular, deep belief networks can be formed by "stacking" RBMs and optionally fine-tuning the resulting deep network with gradient descent and backpropagation.

Approach

Text Summarization can be done for one document, known as single-document summarization,, or for multiple documents,known as multi-document summarization. Here for extractive Text summarization of single document, the deep learning approach is used.This approach broken down into three phases: feature extraction, feature enhancement, and summary generation based on values of those features. Since it can be very difficult to construct high-level,abstract features from raw data, so deep learning is used in the second phase i.e feature enhancement.These extracted features depend highly on how factual the given document is.

Methodology

1.Preprocessing

In this phase we do,
1. Document Segmentation: The text is divided into paragraphs.
2. Paragraph Segmentation: The paragraphs are further divided into sentences.
3. Word Normalization: Each sentence is broken down into words and the words are normalized. Normalization involves lemmatization and results in all words being in one common verb form, crudely stemmed down to their roots with all ambiguities removed. For this purpose, we use Porters algorithm.
4. Stop Word Filtering: Each token is analyzed to remove high frequency stop words.
5. PoS Tagging: Remaining tokens are Part-of-Speech tagged into verb, noun, adjective etc. using the PoS Tagging module supplied by NLTK .

2.Feature Extraction:-

1.Number of thematic words:- The 10 most frequently occurring words are thematic words.
For each sentence, the ratio of no. of thematic words to total words is calculated.
Sentence_Thematic = No. of thematic words/Total words.

2.Sentence position:-This feature is calculated as follows.

$$Sentence\_Position = \begin{cases} 1, \text{if its the first or last sentence of the text} \\ cos((SenPos - min)((1/max) - min)), \text{otherwise} \end{cases}$$

where, SenPos = position of sentence in the text min = th x N , max = th x 2 x N
N is total number of sentences in document this threshold calculated as 0.2 x N By this, we get a high feature value towards the beginning and ending of the document, and a progressively decremented value towards the middle.
3. **Sentence length:-**This feature is used to exclude sentences that are too short as those sentences will not be able to convey much information.

$$Sentence\_Length = \begin{cases} 0, \text{if number of words is less than 3} \\ No.\ of\ words\ in\ the\ sentence, \text{otherwise} \end{cases}$$

4. Sentence position relative to paragraph:-

$$Position\_In\_Para = \begin{cases} 1, \text{if it is the first or last sentence of a paragraph} \\ 0, \text{otherwise} \end{cases}$$

5. **Number of proper nouns:-** Here, we count the total number of words that have been PoS tagged as proper nouns for each sentence.
6. **Number of numerals:-** This feature gives importance to sentences having certain figures. For each sentence we calculate the ratio of numerals to total number of words in the sentence.

$$Sentence\_Numerals = \frac{No.\ of\ numerals}{Total\ words}$$

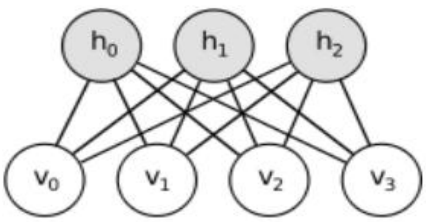
7. **Number of named entities:-** Sentences having references to named entities like a company, a group of people etc.
8. Term Frequency-Inverse Sentence Frequency (TF-ISF): Frequency of each word in a particular sentence is multiplied by the total number of occurrences of that word in all the other sentences.

$$TF - ISF = \frac{log(\sum_{all\ words} TF * ISF)}{Total\ words}$$

9. **Sentence to Centroid similarity:-**Sentence having the highest TF-ISF score is considered as the centroid sentence. Then, we calculate cosine similarity of each sentence with that centroid sentence.
Sentence_Similarity = cosine_sim(sentence,centroid).
After calculating all this 9 feature,we have a sentence-feature matrix.

3. Feature Enhancement:-

Once we get the sentence-feature matrix,we recalculate this matrix using **Restricted Boltzmann Machine** which has one visible layer and one hidden layer.



4. Summary Generation:-

The enhanced feature vector values are used to generate a score against each sentence. The sentences are then sorted according to decreasing score value.The most relevant sentence is the first sentence in this sorted list and is chosen as part of the subset of sentences which will form the summary. This process is

recursively and incrementally repeated to select more sentences until a user specified summary limit is reached.The sentences are then re-arranged in the order of appearance in the original text.

Comparison of approaches / methods:-

Extractive Summarization using Deep Learning	Extractive Text Summarization Using Sentence Ranking	Extractive Text Summarization using Neural Networks
Deep learning technique is used in feature enhancement.	Sentence Ranking is used	Neural Network technique is used.
Deep learning is part of a broader family of machine learning methods based on artificial neural networks with representation learning. Learning can be supervised, semi-supervised or unsupervised.	Sentences are ranked by assigning weights and they are ranked based on their weights. Highly ranked sentences are extracted from the input document so it extracts important sentences which directs to a high-quality summary of the input document	Neural networks are a set of algorithms, modeled loosely after the human brain, that are designed to recognize patterns.
Restricted Boltzmann Machine model is used for summarization.	Novel statistical method method is used to get summary.	Recursive approach is used to produce summaries of variable length documents.
Here 3 steps are carried out pre-processing, feature extraction(deep learning is used in this step), and feature enhancement to create a final summary.	Here, firstly form input file stop is removed than for the remaining keyword frequency is calculated. The weighted frequency of the word is calculated by dividing the frequency of the keywords by maximum frequency of the keywords. Finally, the highest weighted frequency sentence is considered as summery.	Here first, the document is divided into an equal number of pages, and then summarization is done for each page, and the final summary is produced based on summarized pages.
The RBM that are using has 9 perceptrons in each layer with a learning rate of 0.1.	The sentences whose rank is greater than 8 are generated as an output by the summarizer.	For extractive text summarization, a recursive technique is used, so it will take more time to produce a final summary.
The RBM will have to be trained for each new document that has to be summarized.Since each document is unique in the features extracted,the RBM will have to be freshly trained for each new document.	The summarizer model doesn't uses any training model.	The proposed model is trained with two datasets. First is DUC 2002 datasets consist of XML pages that need to be pre-processed. The preprocessing involved converting the dataset into text documents.
Here proposed approach gives less precision value than neural network	Proposed approach doesn't uses any precision value.	This proposed approach gives better precision results when compared with the deep learning approaches.

Identification of research gap

Extractive Text Summarization using Neural Networks

The problem with this technique is that it uses the same sentences as in the original document, and sometimes this may lead to a summary containing less important information because a part of a sentence may contain useful information, but the rest of it may be useless. Another issue is the use of pronouns in the original document. When a certain sentence in the document containing a pronoun is included in the summary, it may be ambiguous to what the pronoun is referring to.Proposed network achieves maximum performance (considering both ROUGE-1 and ROUGE-2 results) when 'page_len' value is set to 40. So when the page_len of the given document is less than 40, the model may not give an accurate result, and it will affect the final summary result. As the Neural network uses a data-driven recursive approach, so when document page_len is more, than the recursive approach takes more iteration and time to process the input.

Extractive Text Summarization Using Sentence Ranking

When a document contains fewer sentences and if we use this document as an input to the summarizer. The summarizer model will not process properly, because here the sentence rank is already set to 8.

Extractive Summarization using Deep Learning

The Restricted Boltzmann Machine model may take some more time in processing and error in encoding values that may affect the final value. RBM model is difficult to train well since the common algorithm used, requires sampling from a Monte Carlo Markov Chain, and as such requires a bit of care to get things just right. Term Frequency-Inverse Sentence Frequency is based on bag-of-words model, therefore it does not capture the position in the text, semantics, co-occurrences in different documents, etc. For this reason, Term Frequency-Inverse Sentence Frequency is only useful as a lexical level feature.

Proposed solution

Extractive Text Summarization using Neural Networks

There is use of pronouns in the original document. When a certain sentence in the document containing a pronoun is included in the summary, it may be ambiguous to what the pronoun is referring to. This can be improved by using k-means clustering algorithm.The K-means algorithm identifies k number of centroids, and then allocates every data point to the nearest cluster, while keeping the centroids as small as possible. k-means algorithm reduces the computation intensity of the neural network, by reducing the input set of samples to be learned.

Extractive Text Summarization Using Sentence Ranking

Sentence Ranking is time taken process,so to improve sentence ranking we could use the indexing after calculating the frequency of each keywords. Indexing scheme are found to perform better than that with the conventional indexing schemes.

Extractive Summarization using Deep Learning

Once we get sentence-feature matrix from feature extraction where we had used the deep learning technique.we can use Restricted Boltzmann Machine and Fuzzy logic both on sentence-feature matrix for adjusting the hyperparameters of the RBM to minimize processing and error in encoded values.By using Restricted Boltzmann Machine and Fuzzy logic both the nature of the summary can be improved.