Text Summarization

Abstract

Now -a-days where tremendous information is available on the internet, it is most important to provide the improved mechanism to extract the information quickly and most efficiently . It is very difficult for human beings to manually extract the summary of a large documents of text. There are plenty of text material available on the internet. So there is a problem of searching for relevant documents from the number of documents available, and absorbing relevant information from it. In order to solve the above two problems, the automatic text summarization is very much necessary. Text summarization is the process of identifying the most important meaningful information in a document or set of related documents and compressing them into a shorter version preserving its overall meanings. Text Summarization is the process of creating a summary of a certain document that contains the most important information of the original one, the purpose of it is to get summary of the main points of the document.

Abstractive Summarization of multi-documents aims to generate a concentrated version of the document while keeping the main information. Due to massive amount of data these days, the importance of summarization arisen. Text Summarization has gained more popularity in this fast-growing information age. Past few years have witnessed a rapid growth in the research of summarizing the text automatically using different approaches. Text summarization has become the necessity of many applications for example search engine, business analysis, market review. Summarization helps to gain required information in less time

Introduction

Automatic Text summarization is a subset of Natural Language Processing (NLP) and is the process of shortening the source text or set of text documents/paragraph while retaining the main information content. The main aim of the Text Summarization is to create a reduced version of the text preserving its essential information.

The main aspects which one should consider while summarization are:

- Generate short summaries.
- Less redundant information summaries.
- Preserve the important information of the source text.

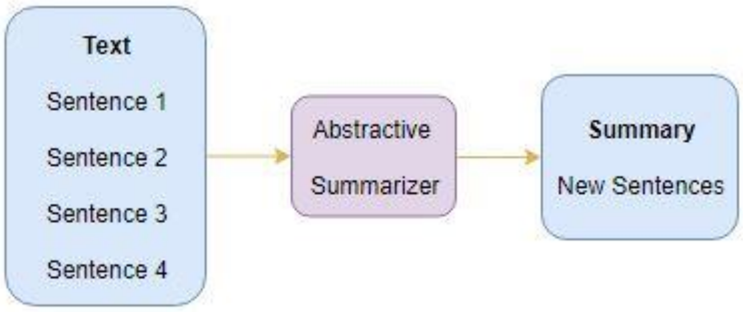
Types of summarization

1)Abstractive Summarization

2)Extractive Summarization

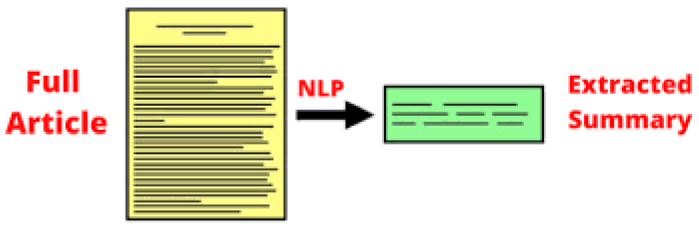
1)Abstractive Summarization:

Abstractive Text Summarization method is the process of using linguistic methods to examine and interpret the text in order to find the new concepts and expressions for generating a new shorter text that conveys the most important information from the original text. As the abstraction process uses linguistic methods and cannot be formulated logically or mathematically, this process is not as easy to implement as it requires a deep understanding of the linguistic skills and semantic understanding of the text.



2)Extractive Summarization:

Extractive Text Summarization selects important sentences, paragraphs etc. from the original text and concatenating them into shorter form without losing the meaning of the text. Majority of the research work done so far have generated summarization systems which are extractive while some work has been done in abstractive summarization as the latter one is harder to develop and one should possess an in-depth knowledge of the linguistics.



Convert Article to Summary

Automatic Text Summarization approaches involves redundancy elimination, significant sentence identification, coherent summary generation and evaluate the automatically generated summaries using evaluation metrics.

Features of extractive summarization

Most of the current automated text summarization systems use extraction method to produce a summary .Sentence extraction techniques are commonly used to produce extraction summaries. One of the methods to obtain suitable sentences is to assign some numerical measure of a sentence for the summary called sentence scoring and then select the best sentences to form document summary based on the compression rate.

In the extraction method, compression rate is an important factor used to define the ratio between the length of the summary and the source text. As the compression rate increases, the summary will be larger, and more insignificant content is contained. While the compression rate decreases the summary to be short, more information is lost. In fact, when the compression rate is 5-30%, the quality of summary is acceptable.

Existing system

1)Term Frequency-Inverse Document Frequency (TF-IDF) method: It is a numerical statistic which reflects how important a word is in a given document. The TF-IDF value increases proportionally to the number of times a word appears in the document. This method mainly works in the weighted term-frequency and inverse sentence frequency paradigm .where sentence-frequency is the number of sentences in the document that contain that term. These sentence vectors are then scored by similarity to the query and the highest scoring sentences are picked to be part of the summary. Summarization is query-specific . The hypothesis assumed by this approach is that if there are ‘‘more specific words’’ in a given sentence, then the sentence is relatively more important. The target words are usually nouns .This method performs a comparison between the term frequency (tf) in a document -in this case each sentence is treated as a document and the document frequency (df), which means the number of times that the word occurs along all documents.

2)Cluster based method: In this method, the semantic nature of a given document is captured and expressed in natural language by a set of triplets (subjects, verbs, objects related to each sentence).Cluster these triplets using similar information. The triplets statements are considered as the basic unit in the process of summarization. More similar the triplets are, the more the information is useless repeated; thus, a summary may be constructed using a sequence of sentences related the computed clusters.

3)Machine Learning approach : In this method, the training dataset is used for reference and the summarization process is modelled as a classification problem: sentences are classified as summary sentences and non-summary sentences based on the features that they possess. The classification probabilities are learnt statistically from the training data, using Bayes’ rule: where, s is a sentence from the document collection, F1, F2...FN are features used in classification. S is the summary to be generated, and P (s∈< S | F1, F2, ..., FN) is the probability that sentence s will be chosen to form the summary given that it possesses features F1,F2...FN

4)**Text summarization with neural networks:** In this method, each document is converted into a list of sentences. Each sentence is represented as a vector [f1,f2,...,f7], composed of 7 features. Seven Features of a Document 1) f1 Paragraph follows title 2) f2 Paragraph location in document 3) f3 Sentence location in paragraph 4) f4 First sentence in paragraph 5) f5 Sentence length 6) f6 Number of thematic words in the sentence 7) f7 Number of title words in the sentence The first phase of the process involves training the neural networks to learn the types of sentences that should be included in the summary. Once the network has learned the features that must exist in summary sentences, we need to discover the trends and relationships among the features that are inherent in the majority of sentences. This is accomplished by the feature fusion phase, which consists of two steps: 1) eliminating uncommon features; and 2) collapsing the effects of common features.

Proposed system

Main steps for text summarization:

There are three main steps for summarizing documents.

These are topic identification, interpretation and summary generation.

Topic Identification : The most prominent information in the text is identified .

There are different techniques for topic identification are used which are Position, Cue Phrases, word frequency. Methods which are based on the position of phrases are the most useful methods for topic identification.

Interpretation : Abstract summaries need to go through interpretation step.

In This step, different subjects are fused in order to form a general content.

Summary Generation : In this step, the system uses text generation method.

NLTK is a leading platform for building Python programs to work with human language data.

NLP is a field of computer science , artificial intelligence and linguistic processing

It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active discussion forum.

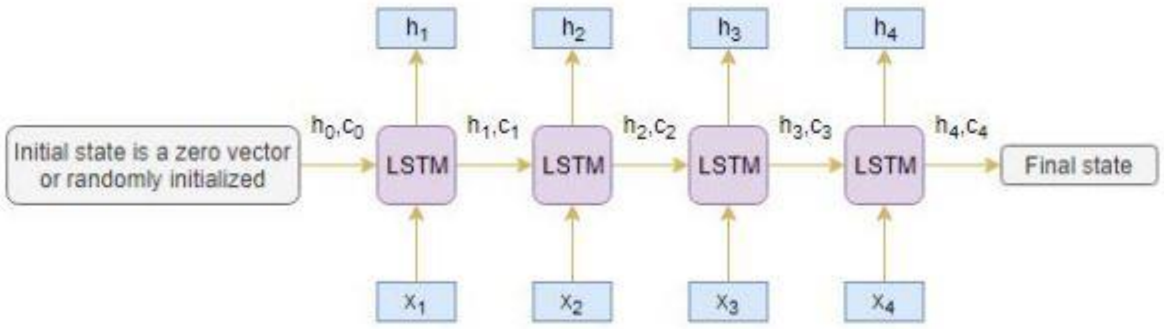
NLTK has been called “a wonderful tool for teaching, and working in, computational linguistics using Python,” and “an amazing library to play with natural language.”

It also includes graphical demonstrations and sample data sets as well as accompanied by a cook book and a book which explains the principles behind the underlying language processing tasks that NLTK supports.

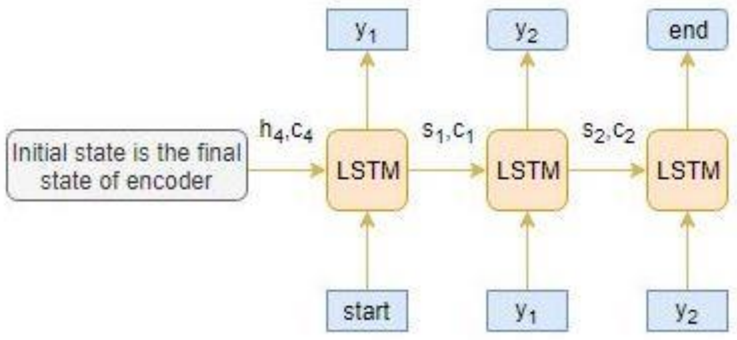
<u>Seq2Seq model:</u>

Seq2Seq model is a model that takes a stream of sentences as an input and outputs another stream of sentences. This can be seen in Neural Machine Translation where input sentences is one language and output sentences are translated versions of that language. Encoder and Decoder are the two main techniques used in seq2seq modeling. Let's see about them.

Encoder Model: Encoder Model is used to encode or transform the input sentences and generate feedback after every step. This feedback can be an internal state i.e hidden state or cell state if we are using the LSTM layer. Encoder models capture the vital information from the input sentences while maintaining the context throughout.
In Neural Machine translation, our input language will be passed into the encoder model where it will capture the contextual information without modifying the meaning of the input sequence. Outputs from the encoder model are then passed into the decoder model to get the output sequences.



Decoder Model: The decoder model is used to decode or predict the target sentences word by word. Decoder input data takes the input of target sentences and predicts the next word which is then fed into the next layer for the prediction. '<start>' (start of target sentence) and '<end>' (end of target sentence) are the two words that help the model to know what will be the initial variable to predict the next word and the ending variable to know the ending of the sentence. While training the model, we first provide the word '<start>', the model then predicts the next word that is the decoder target data. This word is then fed as input data for the next timestep to get the next word prediction.



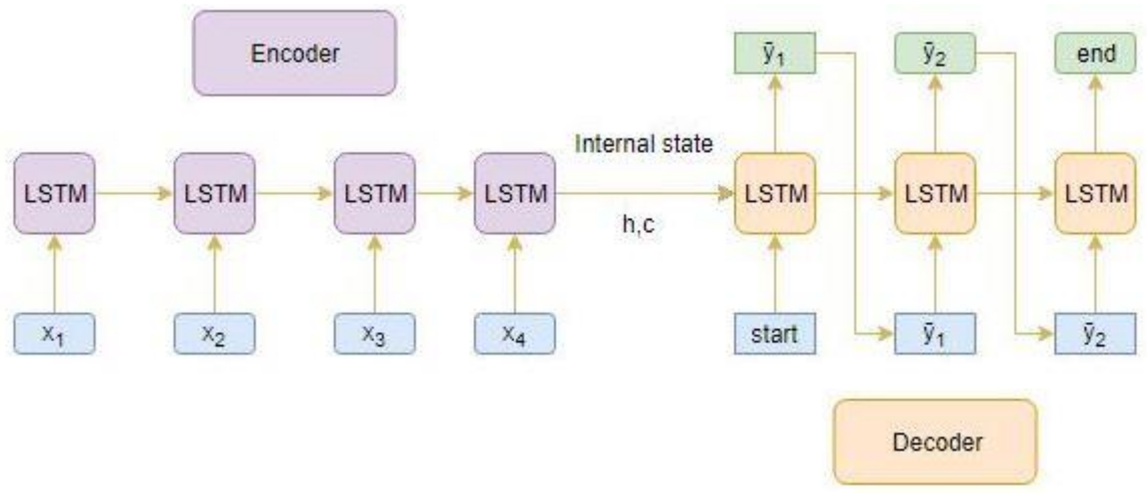
We can set up the Encoder-Decoder in 2 phases:

- Training phase
- Inference phase

In the training phase, we will first set up the encoder and decoder. We will then train the model to predict the target sequence offset by one timestep.

Inference Phase

After training, the model is tested on new source sequences for which the target sequence is unknown. So, we need to set up the inference architecture to decode a test sequence:



Working of inference process Work

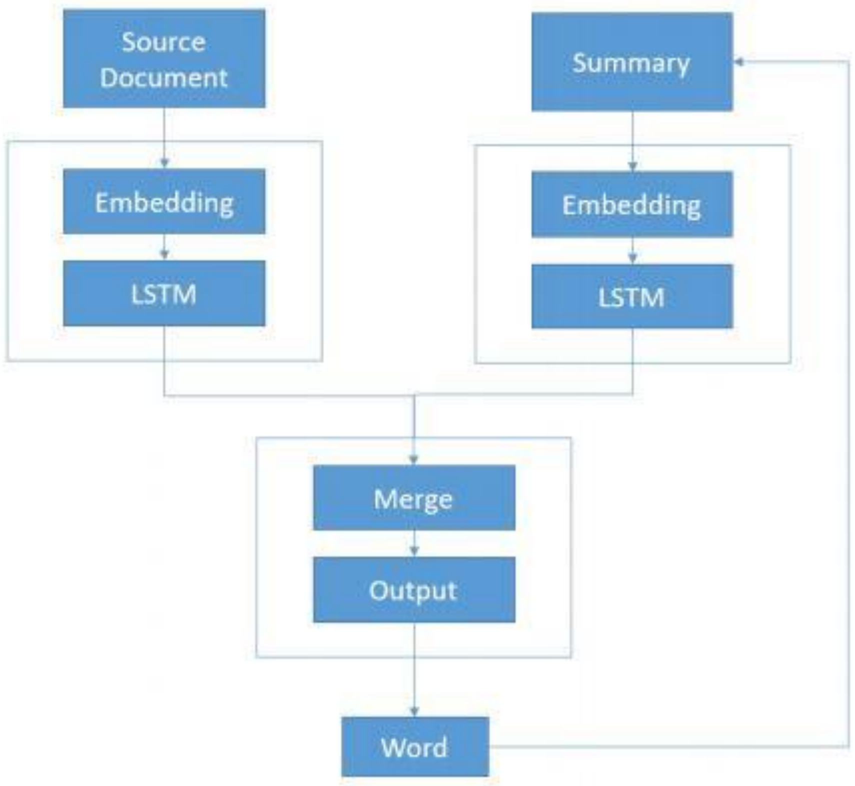
1. Encode the entire input sequence and initialize the decoder with internal states of the encoder
2. Pass <**start>** token as an input to the decoder
3. Run the decoder for one timestep with the internal states
4. The output will be the probability for the next word. The word with the maximum probability will be selected
5. Pass the sampled word as an input to the decoder in the next timestep and update the internal states with the current time step
6. Repeat steps 3 – 5 until we generate <**end>** token or hit the maximum length of the target sequence

.

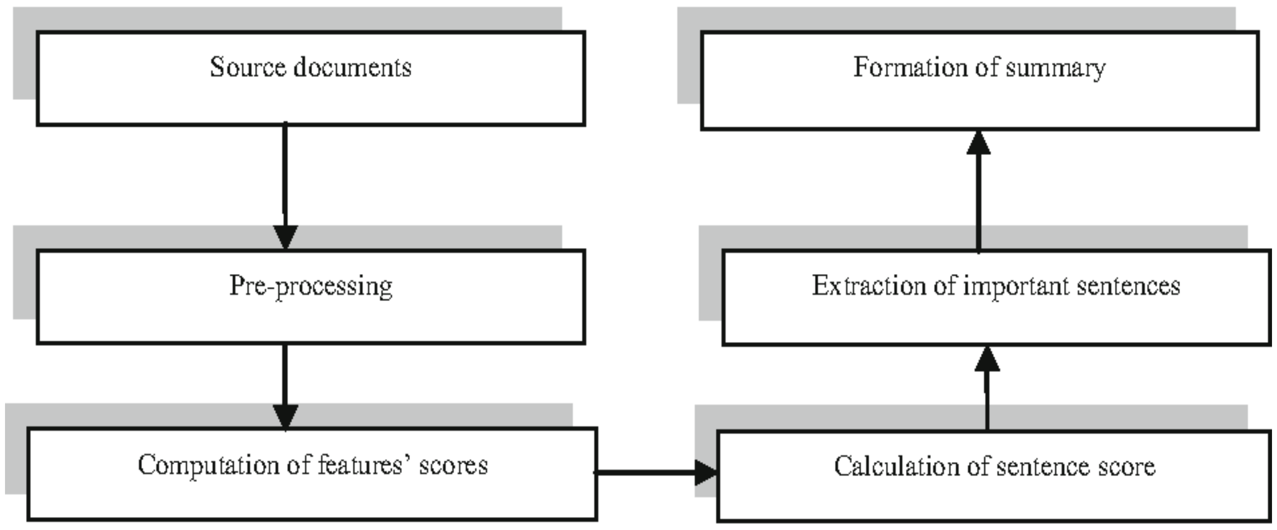
Literature Survey

The author introduced Recall Oriented Understudy for Gisting Evaluation ROUGE. That is an automatic evaluation package for text summarization. The paper also introduced four different measures of ROUGE: - ROUGE-N, ROUGE-L, ROUGE-W and ROUGE-S. It measures the quality of summary by comparing the generated summary with other ideal summaries that are created by humans. These methods are efficient for automatic evaluation of single document summary as well as multi-document summaries. The author has analyzed and compared the performance of three different algorithms. Firstly, the different text summarization techniques explained. Extraction based techniques are used to extract important keywords to be included in the summary. For comparison three comparison three keyword extraction algorithms namely Text Rank, Lex Rank, Latent Semantic Analysis (LSA) were used. Three algorithms are explained and implemented in python language. The ROUGE 1 is used to evaluate the effectiveness of the extracted keywords. The results of the algorithms compared with the handwritten summaries and evaluate the performance. In the end, the Text Rank Algorithm gives a better result than other two algorithms. The author has proposed a system to generate the abstractive summary from the extractive summary using WordNet ontology. The multiple documents had been used like text, pdf, word files etc. The author has discussed various text summarization techniques then author discussed step by step the multiple document text summarization approaches. The experiment result is compared with the existing online extractive tools as well as with human generated summaries and shows the proposed system gives good results. At last the author proposed for the future that the Literature Review on Automatic Text Summarization 9781 summarization accuracy can be increased by comparing this abstractive system with some other abstractive system. In this paper the author uses the extractive text summarization. The author gives the Wikipedia Articles as input to the system and identifies text scoring.

Design



Recursive Text Summarization Model A


Source documents
Pre-processing
Computation of features' scores
Calculation of sentence score
Extraction of important sentences
Formation of summary

Model(b)

Requirements Specifications

Requirement identification is the first step of any project until the necessary requirements are identified the task cannot be begin

System Requirements

It can be divided into two software & Hardware

Hardware Requirements are

RAM(4GB)

ROM(128GB)

I3/I5

processor

Desktop

Software Requirements are

operating system: Windows or MAC

python 3.8 software or Jupiter notebook, with necessary libraries installed in it

Methodology and Implementation

Data set

1. We take data from internet which consists of News articles for making extractive based summarization
2. We can also use novels or story books which consists of maximum pages as a dataset for making extractive based summarization
3. We can also use information of particular topic from Wikipedia as a dataset for extractive based summarization

For LSTM
We downloaded dataset from Kaggle this dataset consists of reviews of fine foods from Amazon. The total datasets include more than 400 rows

Steps involved to perform text summarization

1) Data collection from Wikipedia using web scraping(using Urllib library)

2) Parsing the URL content of the data(using BeautifulSoup library)

3) Data clean-up like removing special characters, numeric values, stop words and punctuations.

4) Tokenization — Creation of tokens (Word tokens and Sentence tokens)

5) Calculate the word frequency for each word.

6) Calculate the weighted frequency for each sentence.

7) Creation of summary choosing 30% of top weighted sentences.

FOR LSTM THESE STEPS ARE INCLUDED

Import the Libraries :

Firstly we will create a file called ‘text_summarizer.py’ and import all the libraries which have been shared in the prerequisites section.

Preprocessing :

Real-world texts are incomplete and they cannot be sent directly to the model that will cause certain errors. So, we clean all our texts and convert them into a presentable form for prediction tasks. So, firstly we will initialize all the variables and methods.

Stemming: Stemming is the process of reducing words into their root words.
For example, if the text contains word like “chocollate” which might be misspelled for “chocolate”. If we don’t stem our words then the model will treat them as two different words. Stemmer will stem or reduce that error word to its root word i.e. “chocol”. As a result, “chocol” is the root word for both “chocolate” and “chocollate”

Splitting the records :

Split the dataset records into training and testing sets. We will be splitting in the 80:20 ratio where 80% record will be for training sets and 20% for testing sets.

Text Vectorization :

We will convert our word into integer sequence using vectorization technique.After converting to integer sequence we will also make all the input and target texts to the same length for our model. So we will take the length of input sentences which has the highest frequency and store it in the ‘max_in_length’ variable, and repeat the same for target data also. Now we will pad arrays of 0’s to the texts if it is less than the assigned maximum input length.

Build the model :

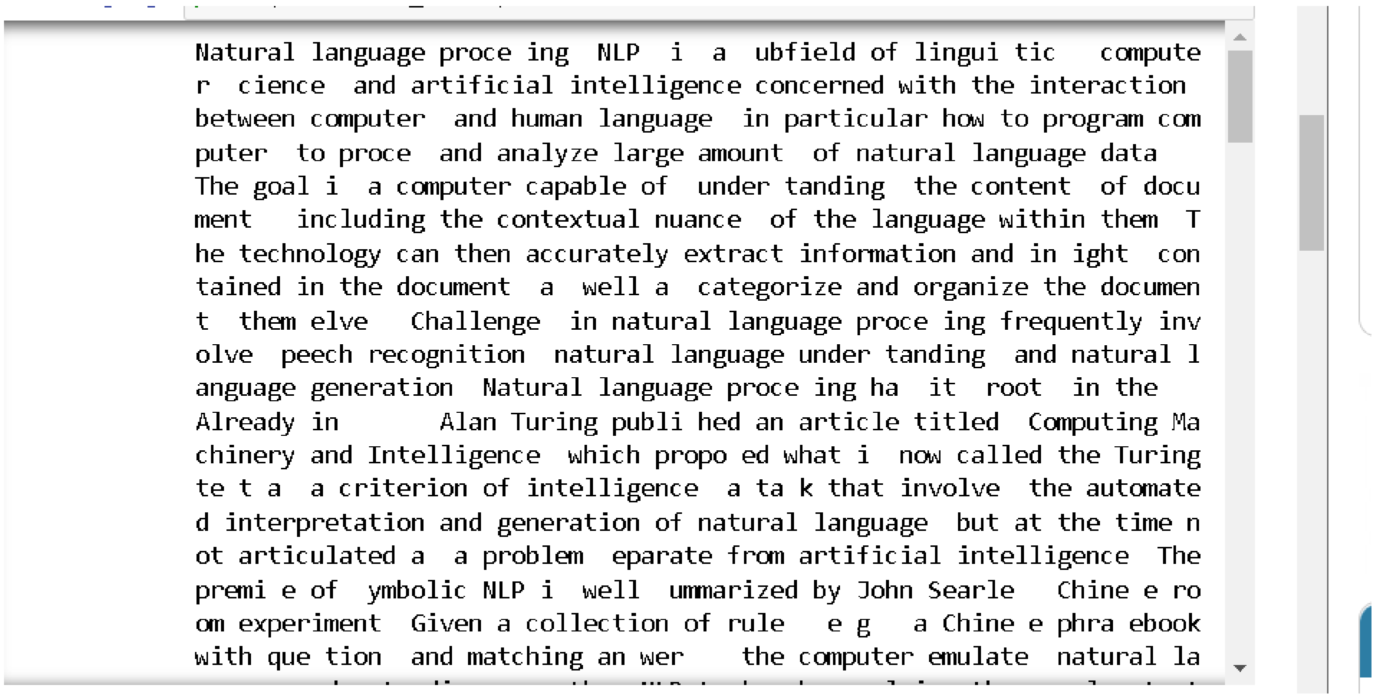
We are using Stacked LSTM containing 3 layers of LSTM stacked on top of each other. This will make our prediction much better. As per your requirement, you can have more also. Let’s understand our encoder model and decoder model.

LSTM: Now we will create 3 stacked LSTM layers where the first LSTM layer will have input of encoder and like that create a continuous sequence of LSTM layers.
The LSTM layer will capture all the contextual information present in the input sequence. We will return hidden state output and also states i.e. hidden state and cell state after execution of every LSTM layer.

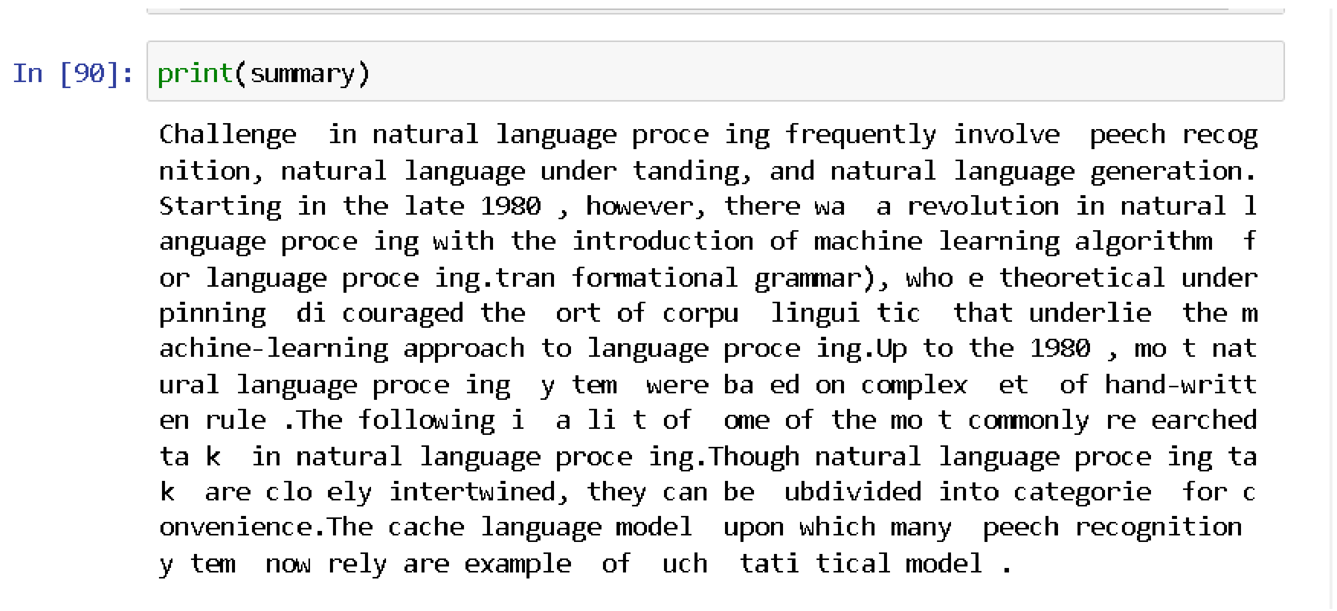
Train the model :

Finally, we will initialize our Model class with input and output data from the encoder and decoder layers. We can plot the model layers and also get the summary of our model.

Test results


```
Natural language proce ing  NLP  i  a  ubfield of lingui tic   compute
r  cience  and artificial intelligence concerned with the interaction
between computer  and human language  in particular how to program com
puter  to proce  and analyze large amount  of natural language data
The goal i  a computer capable of  under tanding  the content  of docu
ment   including the contextual nuance  of the language within them  T
he technology can then accurately extract information and in ight  con
tained in the document  a  well a  categorize and organize the documen
t  them elve  Challenge  in natural language proce ing frequently inv
olve  peech recognition  natural language under tanding  and natural l
anguage generation  Natural language proce ing ha  it  root  in the
Already in        Alan Turing publi hed an article titled  Computing Ma
chinery and Intelligence  which propo ed what i  now called the Turing
te t a  a criterion of intelligence  a ta k that involve  the automate
d interpretation and generation of natural language  but at the time n
ot articulated a  a problem  eparate from artificial intelligence  The
premi e of  ymbolic NLP i  well  ummarized by John Searle   Chine e ro
om experiment  Given a collection of rule   e g   a Chine e phra ebook
with que tion  and matching an wer    the computer emulate  natural la
```

output


```
In [90]: print(summary)

Challenge  in natural language proce ing frequently involve  peech recog
nition, natural language under tanding, and natural language generation.
Starting in the late 1980 , however, there wa  a revolution in natural l
anguage proce ing with the introduction of machine learning algorithm  f
or language proce ing.tran formational grammar), who e theoretical under
pinning  di couraged the  ort of corpu  lingui tic  that underlie  the m
achine-learning approach to language proce ing.Up to the 1980 , mo t nat
ural language proce ing  y tem  were ba ed on complex  et  of hand-writt
en rule .The following i  a li t of  ome of the mo t commonly re earched
ta k  in natural language proce ing.Though natural language proce ing ta
k  are clo ely intertwined, they can be  ubdivided into categorie  for c
onvenience.The cache language model  upon which many  peech recognition
 y tem  now rely are example  of  uch  tati tical model .
```

Conclusion

The purpose of extractive document summarization is to automatically select a number of indicative sentences, passages, or paragraphs from the original document . Text summarization approaches based on Neural Network, Graph Theoretic, Fuzzy and Cluster have, to an extent, succeeded in making an effective summary of a document. Automatic text summarization is an old challenge but the current research direction diverts towards emerging trends in biomedicine, product review, education domains, emails and blogs. This is due to the fact that there is information overload in these areas, especially on the World Wide Web. Automated summarization is an important area in NLP (Natural Language Processing) research. It consists of automatically creating a summary of one or more texts.

The purpose of extractive document summarization is to automatically select a number of indicative sentences, passages, or paragraphs from the original document Both extractive and abstractive methods have been researched. Most summarization techniques are based on extractive methods. Abstractive method is similar to summaries made by humans. Abstractive summarization as of now requires heavy machinery for language generation and is difficult to replicate into the domain specific areas.

Future scope

This work focuses on single document summarization. It can be extended to multi document and multilingual summarization

For e-book stores we can implement this concept so that book buyer can read summary and then purchase.

Bibliography

https://machinelearningmastery.com/gentle-introduction-text-summarization/

https://www.machinelearningplus.com/nlp/text-summarization-approaches-nlp-example/

https://towardsdatascience.com/nlp-preprocessing-with-nltk-3c04ee00edc0

https://medium.com/analytics-vidhya/simple-text-summarization-using-nltk-eedc36ebaaf8

https://www.sciencedirect.com/science/article/pii/S1319157820303712

https://www.researchgate.net/publication/257947528_Text_SummarizationAn_Overview