Automatic Text Summarization

A **Project report** submitted in partial fulfillment of the requirements for the degree of

Bachelor of Technology

by

Siddhi Chavan	**L-15**
Namrata Lakade	**L-32**
Rutuja Patil	**L-47**
Shweta Thakur	**L-68**

Under the guidance of

Prof. Milind Kamble

DEPARTMENT OF
ELECTRONICS & TELECOMMUNICATION ENGINEERING
VISHWAKARMA INSTITUTE OF TECHNOLOGY PUNE 2018-19

Bansilal Ramnath Agarwal Charitable Trust's **VISHWAKARMA INSTITUTE OF TECHNOLOGY, PUNE - 37** (An Autonomous Institute Affiliated to Savitribai Phule Pune University)

CERTIFICATE

This is to certify that the **Project Report** entitled **Title of Project** has been submitted in the academic year **2018-19** by

Siddhi Chavan	**L-15**
Namrata Lakade	**L-32**
Rutuja Patil	**L-47**
Shweta Thakur	**L-68**

under the supervision of **Prof. Milind Kamble** in partial fulfillment of the requirements for the degree of Bachelor of Technology in **Electronics and Telecommunication Engineering** as prescribed by Savitribai Phule Pune University.

Guide/Supervisor
Name: Prof. Milind Kamble
Signature:

Head of the Department
Name: Prof. Shripad Bhatlavande
Signature:

External Examiner
Name:
Signature:

ACKNOWLEDGEMENT

We wish to express our gratitude to Prof**. (Dr.) R. Jalnekar Director, VIT Pune** for providing the facilities of the Institute and for his encouragement during the course of this work. We also express my deep gratitude to **Prof. (Dr.) Shripad Bhatlawande**, the Head of the Department of Electronics and Telecommunication Engineering, VIT Pune for his guidance and support.

We would like to thank my **major project guide Prof. Milind Kamble** for providing us with guidance and help on every step of the way during the course and for imparting us with valuable knowledge.

We would also 1ike to gratefully acknowledge the enthusiastic supervision of our major project guide for his/ her continuous, valuable guidance, patience, constant care, and kind encouragement throughout the major project work.

VIT, Pune.

Date: Dec 2019

ABSTRACT

Propelled by the modern technological innovations, data is to the industry what oil was to the previous one. One of the major tasks is to accumulate and disseminate huge amounts of data. Textual information in the form of digital documents quickly accumulates to huge amounts of data. Most of this large volume of documents is unorganized, it is unrestricted and has not been structured into traditional databases. Processing the documents is therefore an important task, mostly due to the lack of standards. 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 a whopping 180 zettabytes in 2025. With such huge amount of data in the digital space, there is a need to develop machine learning algorithms which can effectively and accurately summarize longer texts and deliver the intended message. Summarizing is an important concept when it comes to finding and expressing vital ideas. This methodology helps us monitor own understanding, practice decision making and learn about sequencing concepts. Manual summarization of various documents is a very tedious task. Hence, automatic text summarization is the requirement of the hour. There is a necessity to reduce the data to shorter, focused summaries that capture the proper gist in order to understand it more effectively as well as check whether the large documents contain the information required. Automatic text summarization methods are used to deal with the huge chunk of text data available online to help discover relevant information and to usage of the relevant information swiftly. The main idea of summarization is to find a subset of data which contains the "information" of the entire set. Such techniques are widely used in industry today. Search Engines are also an example - summarization of documents, image collections and videos. Document summarization is to create a representative summary or abstract of the entire document, by finding the most informative sentences in order to convey the main idea, while in image summarization the system finds the most representative and important (i.e. salient) images. It also reduces the reading time, accelerates the process of researching effectively increasing the amount of information that can fit in a small space.

INDEX

Topic	Page No.

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

With the rise of internet and its unlimited sources — news, social media, office emails, etc. we now have information readily available to us. Earlier summarization was done manually but with this increased data, we need automatic summarization to cope up with the huge data. Text summarization generates brief form of huge documents thus keeping the original info intact. It can be carried out on one single document or on many similar documents. Based on its nature, summarization is classified into Abstractive and Extractive summarization.

1.1 Types of Summarization:

Extractive

The summarized text is a sub part of the original document. It works similar to a highlighter which highlights the important parts of a document. In this way of approach passages or sentences are selected from the whole content and rearranged in form of summary. Lack of cohesion is a major drawback of extractive summarization.

Abstractive

It involves generating novel sentences for the summary. The idea is to capture the whole text document in a single vector(encoding) and then use this vector to generate words of the summary one by one(decoding).

1.2 Sequence-to-Sequence modelling

Sequence-to-Sequence is a concept in machine learning which provides a general-purpose Encoder-Decoder framework which is used by TensorFlow. It is about converting sequences from one domain to sequences in another domain. A model can be trained using a single command with the help of Sequence-to-Sequence model. It maps an input of sequence to an output of sequence with a tad and an attention value. The basic idea is to use 2 RNN algorithms put together with a special token and predict the next state sequence from the previous one. The following fig 1. Shows the basic block diagram of Seq-to-Seq model:

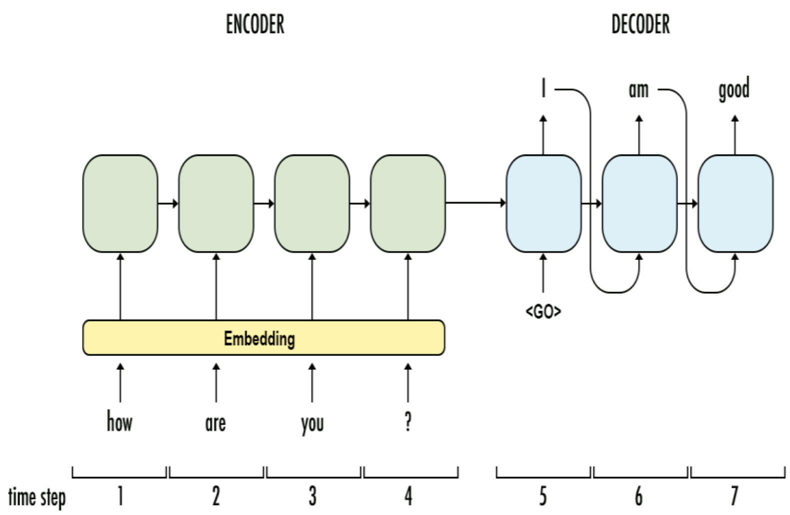


Fig. 1 Seq-to-Seq Model

1.3 LSTM

LSTM (Long Short-Term Memory) Networks are called fancy recurrent neural networks with some additional features. LSTM's have the property of selectively remembering patterns for long durations of time. LSTMs make small modifications to the given information by multiplications and additions. In LSTMs, the information flows through a mechanism known as cell states. Thus, it can selectively remember or forget things. The information at a particular cell state has three different dependencies. LSTM cell contains the following parts

Forget Gate "f" (NN with sigmoid)
Candidate layer "C`"(NN with Tanh)
Input Gate "I" (NN with sigmoid)
Output Gate "O" (NN with sigmoid)
Hidden state "H" (a vector)
Memory state "C" (a vector)

Phases

1)Training Phase: Training phase includes setting up the encoder and decoder. Then it is trained to predict the target sequence offset by one step time.

2)Inference Phase: Once the training phase of the model is completed, the model is tested using various other source sequences where the target sequence is not known. Hence, a new inference architecture is created for the purpose.

1.4 RNN

A **recurrent neural network** (**RNN**) is a class of artificial neural networks where connections between nodes form a directed graph along a temporal sequence. This allows it to exhibit temporal dynamic behavior. Unlike feedforward neural networks, RNNs can use their internal state (memory) to process sequences of inputs. This makes them applicable to tasks such as unsegmented, connected handwriting recognition or speech recognition. The term "recurrent neural network" is used indiscriminately to refer to two broad classes of networks with a similar general structure, where one is finite impulse and the other is infinite impulse. Both classes of networks exhibit temporal dynamic behavior. A finite impulse recurrent network is a directed acyclic graph that can be unrolled and replaced with a strictly feedforward neural network, while an infinite impulse recurrent network is a directed cyclic graph that cannot be unrolled. Both finite impulse and infinite impulse recurrent networks can have additional stored state, and the storage can be under direct control by the neural network. The storage can also be replaced by another network or graph, if that incorporates time delays or has feedback loops. Such controlled states are referred to as gated state or gated memory, and are part of long short-term memory networks (LSTMs) and gated recurrent units. This is also called Feedback Neural Network.In traditional neural networks, all the inputs and outputs are independent of each other, but in cases like when it is required to predict the next word of a sentence, the previous words are required and hence there is a need to remember the previous words. Thus, RNN came into existence, which solved this issue with the help of a Hidden Layer. The main and most important feature of RNN is Hidden state, which remembers some information about a sequence.

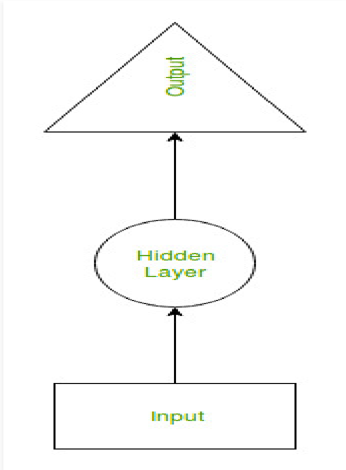


Fig. 2 RNN structure

RNN have a “memory” which remembers all information about what has been calculated. It uses the same parameters for each input as it performs the same task on all the inputs or hidden layers to produce the output. This reduces the complexity of parameters, unlike other neural networks.

Advantages of Recurrent Neural Network

1)An RNN remembers each and every information through time. It is useful in time series prediction only because of the feature to remember previous inputs as well. This is called Long Short-Term Memory.

2)Recurrent neural network is even used with convolutional layers to extend the effective pixel neighborhood.

Disadvantages of Recurrent Neural Network

1)Gradient vanishing and exploding problems.

2)Training an RNN is a very difficult task.

3)It cannot process very long sequences if using tan(h) or rel(u) as an activation function.

1.5 Spyder-Libraries

TensorFlow

Currently, the most famous deep learning library in the world is Google's TensorFlow. Google product uses machine learning in all of its products to improve the search engine, translation, image captioning or recommendations. To give a concrete example, Google users can experience a faster and more refined the search with AI. If the user types a keyword the search bar, Google provides a recommendation about what could be the next word.

Architecture

TensorFlow architecture works in three parts:

1)Preprocessing the data

2)Build the model

3)Train and estimate the model

It is called TensorFlow because it takes input as a multi-dimensional array, also known as **tensors**. You can construct a sort of **flowchart** of operations that you want to perform on that input. The input goes in at one end, and then it flows through this system of multiple operations and comes out the other end as output. This is why it is called TensorFlow because the tensor goes in it flows through a list of operations, and then it comes out the other side.

Keras

1)Keras is an Open Source Neural Network library written in Python that runs on top of Theano or TensorFlow. It is designed to be modular, fast and easy to use.
2)Keras doesn't handle low-level computation. Instead, it uses another library to do it, called the "Backend”. So Keras is high-level API wrapper for the low-level API, capable of running on top of TensorFlow, CNTK, or Theano.
3)Keras High-Level API handles the way we make models, defining layers, or set up multiple input-output models. In this level, Keras also compiles our model with loss and optimizer functions, training process with fit function. Keras doesn't handle Low-Level API such as making the computational graph, making tensors or other variables because it has been handled by the "backend" engine.

Advantages of Keras

1)Fast Deployment and Easy to understand.
2)It is an API designed with user friendly implementation as the core principle. The API is designed to be simple and consistent, and it minimises the effort programmers are required to put in to convert theory into action.
3)Keras’ modular design is another important feature. The primary idea of Keras is layers, which can be connected seamlessly.
4)Keras is extensible. If you are a researcher trying to bring in your own novel functionality, Keras can accommodate such extensions.
5)Keras is all Python, so there is no need for tricky declarative configuration files.

NLTK

The Natural Language Toolkit, or more commonly NLTK, is a suite of libraries and programs for symbolic and statistical natural language processing (NLP) for English written in the Python programming language. NLTK includes graphical demonstrations and sample data. It is accompanied by a book that explains the underlying concepts behind the language processing tasks supported by the toolkit, plus a cookbook. NLTK has been used successfully as a teaching tool, as an individual study tool, and as a platform for prototyping and building research systems. NLTK is a leading platform for building Python programs to work with human language data. 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.

Chapter 2

Literature Survey

1. A Review Paper on Text Summarization(Deepali K. Gaikwad, 2016)**:** In This paper discusses various approaches using the abstractive and extractive methods. The main features are term frequency, location, cue method, title word, sentence length, similarity, proper nouns, proximity are described.

2. Text Summarization: An Overview: (Babar, Oct 2013) This paper talks about the two groups of summarizations i.e. “indicative” and “informative”. It also describes the intrinsic and the extrinsic measures of summary evaluation.

3. A Survey on Extractive Text Summarization- (N. Moratanch, S. Chitrakala, Dept. of CSE, Anna University, CEG, 2017)**:** The paper in the 1st section depicts the features for extractive text summarization e.g. Word level features and sentence level features

4. The Automatic Creation of Literature Abstracts (H. P. Luhn,, April 1958)**:** The method proposed selects those among all the sentences of an article that are the most representative of pertinent information. This technique is auto-abstract which is based on various properties of writing which are analyzed by various styles of literature

5. Text Summarization with Pretrained Encoders: (Yang Liu, 22 Aug 2019) The paper describes a model extending the idea of word embeddings by learning contextual representations by using language modelling object (BERT). This paper successfully proves that BERT can be used in a Novel document level encoder (Text summarization).

6. How to Write Summaries with Patterns? Learning towards Abstractive Summarization through Prototype Editing (ShenGao, 19 sept 2019) **:** The paper aims to present data in a particular pattern for which it has a prototype, like in courts there are only certain ways in which statements are written. This can be achieved by calculating cross dependency between prototype document - summary pair to obtain summary pattern and prototype fact.

7. WikiHow: A Large-Scale Text Summarization: (Mahnaz Koupaee, Oct 18, 2018) This paper talks about a large-scale dataset (WikiHow) and employing different techniques for different datasets. It introduces level of abstractedness and compression ratio metrics to show how abstractive the new dataset is.

8. New Methods in Automatic Extraction: (H. P. Edmundson, April 1969) There are four ways in which this paper proposes to extract text. Method 1 is the cue method in which the relevance of a sentence is affected by the presence of pragmatic word like "hardly”. Method 2 is key and suggests high frequency words are positively relevant. Method 3 talks about the positive relevance a title provides in the final document as an author tries to circumscribe the text in the title. Method 4 shows that the location of a word or the place of the occurrence of the word provides positive weight.

9. Abstractive Text Summarization using Sequence-to-Sequence RNNs and Beyond: (Ramesh Nallapati, 2016) The model consists of an encoder and decoder which evaluates using full length Rouge F1 matric that we employed for the Gigaword corpus. It is found that the good summary outputs were more prevalent in the system than the bad ones. This method does perform well but has a length constrain.

10. An Extensive Survey on Deep Learning Applications : (Usha Devi N, Feb 2017) The performance of machine learning methods are heavily dependent on the choice of data therefore the designing of pre-processing pipelines and data transformation takes maximum effort but it's inability to extract and organize the discriminative information from the data is a concern . To overcome this, feature engineering is used and it is highly desirable to make learning algorithms to expand the scope and ease of applicability of machine learning.

11. Machine Learning Approach for Automatic Text Summarization Using Neural Network: (Meetkumar Patel, Jan 2018)The paper, describes about the ML approach using ANN to generate summaries of arbitrary length articles. The architecture consists of two neural networks working in parallel simultaneously- an encoder that takes the input sequence and produces a vector output and the decoder that takes the previous vector output as its input and generates the final output sequence.

12. Neural Approaches Towards Text Summarization Extractive: (Singh, July 2018) The encoder decoder architecture modeled using RNN is central approach in the system. The Deep-classifier proposed by Nallapati uses GRU-RNN to sequentially accept or reject each sentence in the document for being present in the summary.

13. Review of Text Summarization using Gated Neural Networks, Networks: (Touseef Iqbal, Apr 2018) The essential thought behind GRU is to make utilization of a few gates for controlling the stream of data from past strides to the present advances. By applying the gates, mapping starting with one point then onto the next can be learned by any repetitive unit.

14. A Deep Learning Approach to Understanding Cloud Service Level Agreements: (Srishty Saha, May 24, 2017) This paper proposes a framework to extract semantically similar terms and entities across cloud service documents using word embeddings and neural networks. It is intended to aid cloud service consumers across a variety of fields by providing the ability to understand the services and requirements offered by large-scale commercial cloud service.

15) Abstractive and Extractive Text Summarization using Document Context Vector and Recurrent Neural Networks: (Chandra Khatri (Amazon lab), Gyanit Singh (Ebay Inc.), Nish Parikh (Google), July 2018) The paper proposes a novel Document-Context based Seq2Seq models using RNNs for abstractive and extractive summarizations. The algorithm is trained on human-extracted-golden-summaries. It overcomes the problem of generative models.

16. Long Short-Term Memory Neural: (Sepp Hochreiter, 1997) Learning to store information over extended time intervals via recurrent back propagation takes a very long time, mostly due to insufficient decaying error back. Truncating the gradient where this does not do harm, LSTM can learn to bridge minimal time lags in excess of 1000 discrete time steps by enforcing constant error through\constant error carrousels within special units.

17. Deep Learning: Methods and Applications: (Li Deng, Dong Yu, 2013) Deep learning that is discussed in the paper is about learning with deep architectures for signal and information processing. It is not about deep understanding of the signal or information, although in many cases they may be related. two key aspects: (1) models consisting of multiple layers or stages of nonlinear information processing; and (2) methods for supervised or unsupervised learning of feature representation at successively higher, more abstract layers

18. Improving Abstraction in Text Summarization: (Wojciech Krysci ´ nski ´, 2018) Abstractive text summarization aims to shorten long text documents into a human readable form that contains the most important facts from the original document. The paper proposes two techniques to improve the level of abstraction of generated summaries. Including mechanisms to promote paraphrase generation in the summary generator could be an interesting direction.

19. An efficient single document Arabic text summarization using a combination of statistical and semantic features: (Aziz Qaroush, 26 March 2019) In this paper, it is proposed that an automatic, generic, and extractive Arabic single document summarizing method aiming at producing a sufficiently informative summary. This method evaluates each sentence based on a combination of statistical and semantic features in which a novel formulation is used taking into account sentence importance, coverage and diversity.

20. Automatic Text Summarization Using Reinforcement Learning with Embedding Features: (Gyoung Ho Lee)They propose a novel deep learning network for estimating Q-values used in Reinforcement learning. They evaluate their model by using ROUGE scores with DUC 2001, 2002, Wikipedia, ACL-ARC data. Evaluation results show that the model is competitive with the previous models.

Literature Survey Summary:

The survey suggests various ways of compressing a rather large amount of data or text and all confirm the better effectiveness of abstractive and deep learning method to be superior among the rest. In a paper it has been suggested that the significant factor that is the frequency or occurrence of a word determines higher probability to be selected into the summarized text and fails to understands different forms of word. This can be highly unreliable and therefore the Lstm and Seq2Seq algorithms can be employed as it provides a better understanding of occurrence of words just like that of humans. Since a summary is highly subjective to human’s extractive method is found highly redundant and coherent. The mechanism of encoding and decoding through Lstm is attractive due to its ability to present the comprehensiveness of the data and build a systematic outcome. Hence, to avoid grammatical inconsistency and poor representation of data the course of route for summarization chosen was deep learning and RNN for better prediction efficiency and performance. Further it was discovered that the encoder-decoder structure has a very large time complexity and wasn’t realizable. Hence, Extractive method of summarization which comprised of the Term frequency method was implemented.

CHAPTER 3

Methodology

3.1 Sequence-to-Sequence Modelling

A sequence to sequence model maps fixed length input to fixed length output where the output length can be varied i.e. small length output can be obtained to form a summary, with a tag and attention value. The lengths of output and input vector may vary. The model runs based on two recurrent neural networks working in parallel. The two RNNs will co-ordinate with an exchange of a special token and then try to predict the next state sequence from the previous sequence. The model consists of an encoder, intermediate encoder vector and a decoder.

3.2 Encoder-Decoder Architecture

Encoder

The encoder is a pile of several recurrent units where every unit accepts a single element from the input sequence, gathers information and then sends it forward. Each word is denoted by y_t where I is the order of that word. The hidden states h_i is calculated by the formula in eqn 1:

$$h_t = f(W^{(hh)}h_{t-1} + W^{(hx)}x_t)$$

……………………………………………….......**eqn 1**

Now by assigning proper weights to the previous hidden states and the input vector x, prediction can be carried forward. The encoder vector tries to wrap up the information of all input elements so that the decoder can work with precision. This works the first hidden state of the decoder.

Decoder

This pile of recurrent units predicts an output y_t at a given time step t. These recurrent unit intakes the hidden states from the previous segments and generates output as well as its own hidden state. Hidden state is calculated by the formula

$$h_t = f(W^{(hh)}h_{t-1})$$

…………………………………………........................**eqn 2**

Next according to the respective weights, the output is calculated being considerate of the hidden states and the present time step. The advantage of this system is that it can map sequences which are not equal in length with each other. Fig 3. shows an encoder decoder architecture:

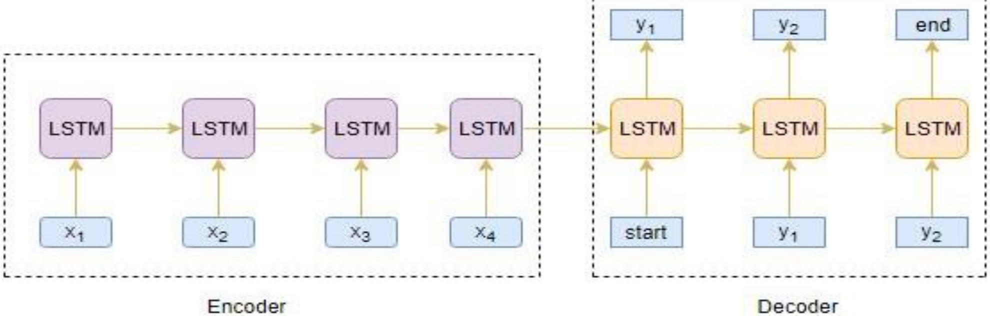


Fig. 3 Encoder-Decoder architecture

3.3 Attention Mechanism

Attention is one component of a network's architecture. The basic idea: each time the model predicts an output word, it only uses parts of an input where the most relevant information is concentrated instead of an entire sentence, i.e. it only pays attention to some input words. It is in charge of managing and quantifying the interdependence:

1)Between the input and output elements (General Attention)

2)Within the input elements (Self-Attention)

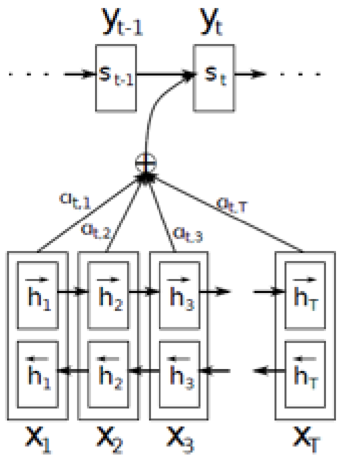


Fig.4 Attention Mechanism

Fig. 4 shows basic attention mechanism. The Bidirectional LSTM used in attention mechanism generates a sequence of annotat*ions (h1, h2, ..., hTx)* for each input sentence. All the vectors h1, h2.., etc., used in their work are basically the concatenation of forward and backward hidden states in the encoder.

$$h_j = \left[\overrightarrow{h}_j^\top ; \overleftarrow{h}_j^\top\right]^\top$$ ………………………………………..........................**eqn 3**

All the vectors h1, h2, h3…., hTx are representations of Tx number of words in the input sentence. In the simple encoder and decoder model, only the last state of the encoder LSTM was used (hTx in this case) as the context vector. The context vector ci for the output word yi is generated using the weighted sum of the annotations. eij is the output score of a feedforward neural network described by the function a that attempts to capture the alignment between input at j and output at i. Basically, if the encoder produces Tx number of "annotations" (the hidden state vectors) each having dimension d, then the input dimension of the feedforward network is (Tx , 2d) (assuming the previous state of the decoder also has d dimensions and these two vectors are concatenated). This input is multiplied with a matrix Wa of (2d, 1) dimensions (of course followed by addition of the bias term) to get scores eij (having a dimension (Tx , 1)). On the top of these eij scores, a tan hyperbolic function is applied followed by a SoftMax to get the normalized alignment scores for output j:

E = I [Tx*2d] * Wa [2d * 1] + B[Tx*1] ……………………………………………......**eqn 4**

α = SoftMax(tanh(E)) ...……………………………………….................................**eqn 5**

C= IT * α………………………………………...**eqn 6**

So, ***α*** is a (Tx, 1) dimensional vector and its elements are the weights corresponding to each word in the input sentence.

Steps using deep learning

1)Import all the necessary libraries.
2)Implement attention layer.
3)Read the dataset.
4)Drop duplicate values.
5)Preprocessing of the data.
6)Text Cleaning.
7)Summary Cleaning.
8)Prepare tokenizer (text and summary).
9)Model Building
   9.1) Inference
   9.2) Metrics
   9.3) Callback
   9.4) Model fitting
10)Diagnostic plot
11)Reverse Dictionary
12)Inference
    12.1) Inference Setup
    12.2) Inference Process
13)Integer to text
14)Predictions

3.4 TERM FREQUENCY

Term frequency is used to connect the information retrievals and shows the frequency of a term (number of times term occurs). It indicates the significance of that term in the entire document. This value is always written in context of inverse document frequency. For the computation of the keyword density the term frequency value has to consulted. This ensures that frequent words are not too heavily weighted and the common or rare words are not weighted too lower.

Steps:

1)Load the corpus.
2)Tokenize the paragraphs/sentences.
3)Find weighted frequency of occurrence.
4)Replace words by weighted frequencies in original corpus.
5)Sort sentences in descending order of sum.
6)Calculating sentence scores.
7)Generate the summary.

CHAPTER 4

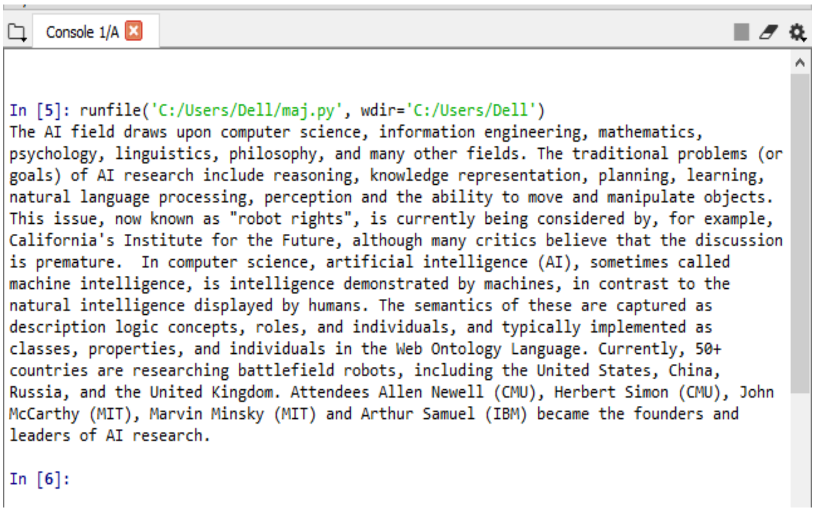
Experiment and Results

Dataset:

1) 15 web pages were used for the experimental purpose.
2) The pages used were from scientific domain.
3) The pages comprised of approximately 7626.87 words on average.
4) The dataset used for the abstractive summarization part comprised of a total 568455 reviews. They were read using a .csv file. They were reviews about general consumer products like dog food, ice-cream, etc. The dataset consisted of 8 parameters i.e product id, user id, profile name, helpfulness numerator, helpfulness denominator, score, time, actual summary, text.

Results

Input: https://en.wikipedia.org/wiki/Artificial_intelligence
Output:



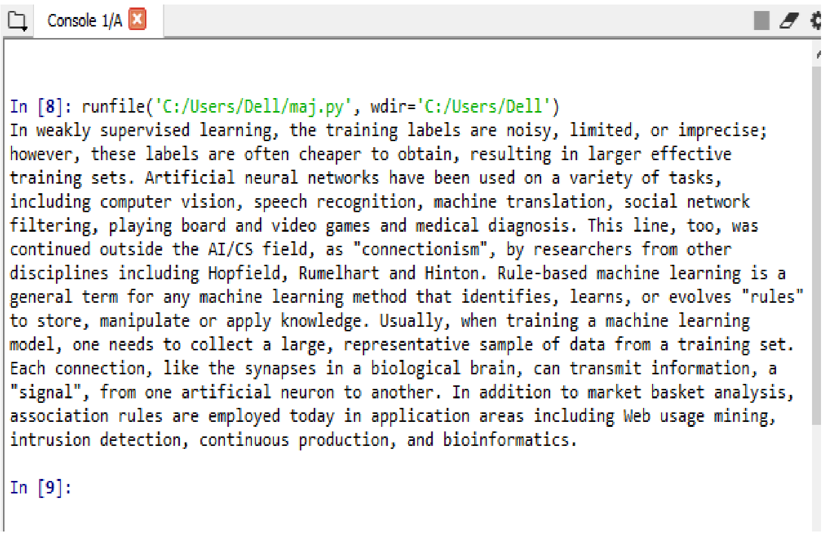
```
Console 1/A

In [5]: runfile('C:/Users/Dell/maj.py', wdir='C:/Users/Dell')
The AI field draws upon computer science, information engineering, mathematics,
psychology, linguistics, philosophy, and many other fields. The traditional problems (or
goals) of AI research include reasoning, knowledge representation, planning, learning,
natural language processing, perception and the ability to move and manipulate objects.
This issue, now known as "robot rights", is currently being considered by, for example,
California's Institute for the Future, although many critics believe that the discussion
is premature.  In computer science, artificial intelligence (AI), sometimes called
machine intelligence, is intelligence demonstrated by machines, in contrast to the
natural intelligence displayed by humans. The semantics of these are captured as
description logic concepts, roles, and individuals, and typically implemented as
classes, properties, and individuals in the Web Ontology Language. Currently, 50+
countries are researching battlefield robots, including the United States, China,
Russia, and the United Kingdom. Attendees Allen Newell (CMU), Herbert Simon (CMU), John
McCarthy (MIT), Marvin Minsky (MIT) and Arthur Samuel (IBM) became the founders and
leaders of AI research.

In [6]:
```

Fig 5 Output 1

Input: https://en.wikipedia.org/wiki/Machine_learning
Output:



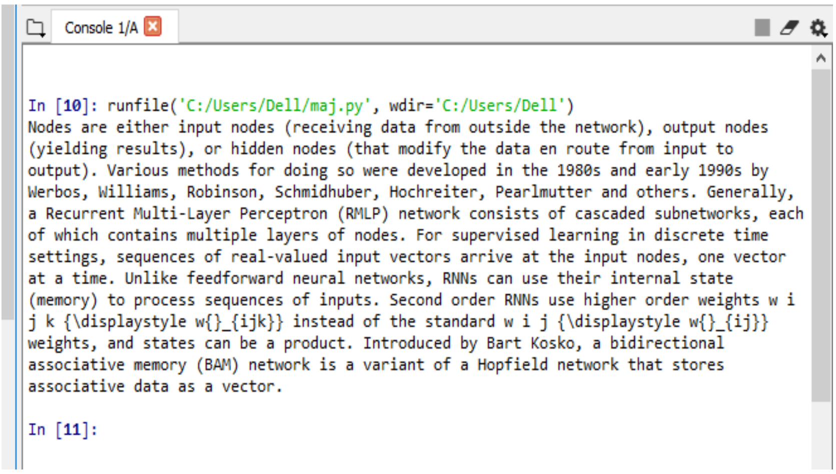
```
Console 1/A

In [8]: runfile('C:/Users/Dell/maj.py', wdir='C:/Users/Dell')
In weakly supervised learning, the training labels are noisy, limited, or imprecise;
however, these labels are often cheaper to obtain, resulting in larger effective
training sets. Artificial neural networks have been used on a variety of tasks,
including computer vision, speech recognition, machine translation, social network
filtering, playing board and video games and medical diagnosis. This line, too, was
continued outside the AI/CS field, as "connectionism", by researchers from other
disciplines including Hopfield, Rumelhart and Hinton. Rule-based machine learning is a
general term for any machine learning method that identifies, learns, or evolves "rules"
to store, manipulate or apply knowledge. Usually, when training a machine learning
model, one needs to collect a large, representative sample of data from a training set.
Each connection, like the synapses in a biological brain, can transmit information, a
"signal", from one artificial neuron to another. In addition to market basket analysis,
association rules are employed today in application areas including Web usage mining,
intrusion detection, continuous production, and bioinformatics.

In [9]:
```

Fig 6 Output 2

Input: https://en.wikipedia.org/wiki/Recurrent_neural_network
Output:

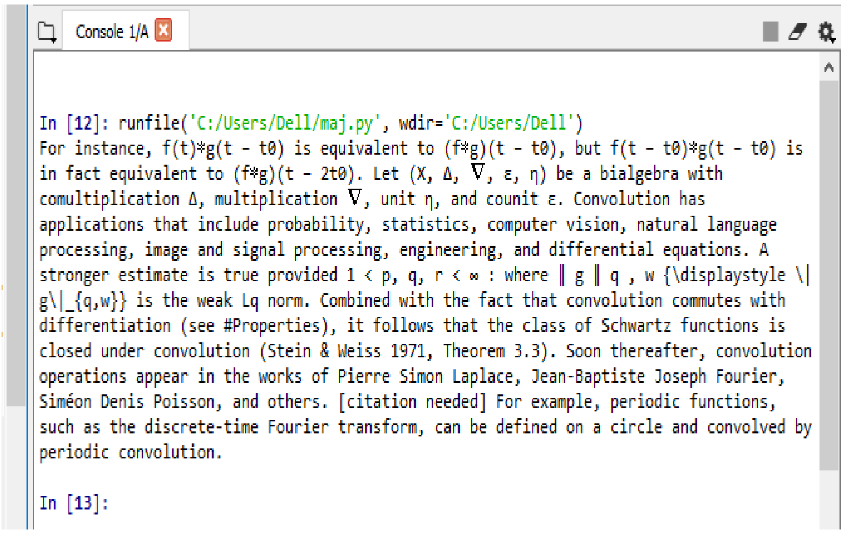


```
In [10]: runfile('C:/Users/Dell/maj.py', wdir='C:/Users/Dell')
Nodes are either input nodes (receiving data from outside the network), output nodes
(yielding results), or hidden nodes (that modify the data en route from input to
output). Various methods for doing so were developed in the 1980s and early 1990s by
Werbos, Williams, Robinson, Schmidhuber, Hochreiter, Pearlmutter and others. Generally,
a Recurrent Multi-Layer Perceptron (RMLP) network consists of cascaded subnetworks, each
of which contains multiple layers of nodes. For supervised learning in discrete time
settings, sequences of real-valued input vectors arrive at the input nodes, one vector
at a time. Unlike feedforward neural networks, RNNs can use their internal state
(memory) to process sequences of inputs. Second order RNNs use higher order weights w i
j k {\displaystyle w{}_{ijk}} instead of the standard w i j {\displaystyle w{}_{ij}}
weights, and states can be a product. Introduced by Bart Kosko, a bidirectional
associative memory (BAM) network is a variant of a Hopfield network that stores
associative data as a vector.

In [11]:
```

Fig 7 Output 3

Input: https://en.wikipedia.org/wiki/Convolution
Output:

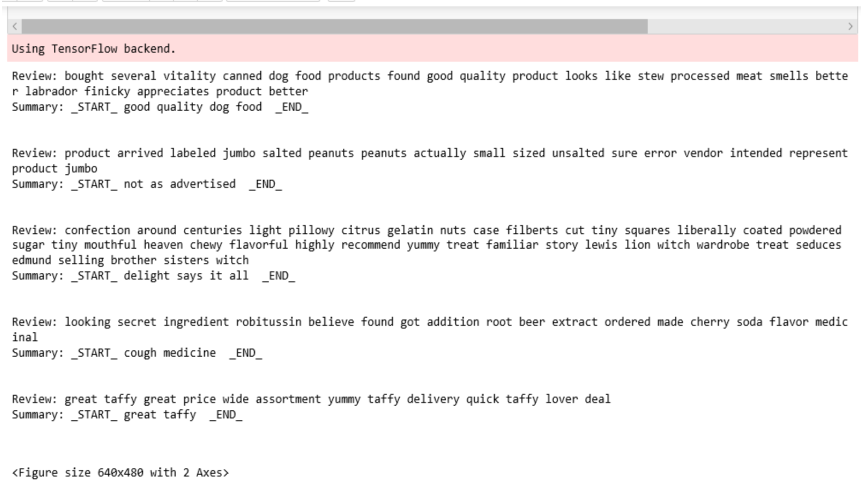


```
In [12]: runfile('C:/Users/Dell/maj.py', wdir='C:/Users/Dell')
For instance, f(t)*g(t - t0) is equivalent to (f*g)(t - t0), but f(t - t0)*g(t - t0) is
in fact equivalent to (f*g)(t - 2t0). Let (X, Δ, ∇, ε, η) be a bialgebra with
comultiplication Δ, multiplication ∇, unit η, and counit ε. Convolution has
applications that include probability, statistics, computer vision, natural language
processing, image and signal processing, engineering, and differential equations. A
stronger estimate is true provided 1 < p, q, r < ∞ : where ‖ g ‖ q , w {\displaystyle \|
g\|_{q,w}} is the weak Lq norm. Combined with the fact that convolution commutes with
differentiation (see #Properties), it follows that the class of Schwartz functions is
closed under convolution (Stein & Weiss 1971, Theorem 3.3). Soon thereafter, convolution
operations appear in the works of Pierre Simon Laplace, Jean-Baptiste Joseph Fourier,
Siméon Denis Poisson, and others. [citation needed] For example, periodic functions,
such as the discrete-time Fourier transform, can be defined on a circle and convolved by
periodic convolution.

In [13]:
```

Fig 8 Output 4

Text summarization using LSTM



```
Using TensorFlow backend.

Review: bought several vitality canned dog food products found good quality product looks like stew processed meat smells bette
r labrador finicky appreciates product better
Summary: _START_ good quality dog food  _END_


Review: product arrived labeled jumbo salted peanuts peanuts actually small sized unsalted sure error vendor intended represent
product jumbo
Summary: _START_ not as advertised  _END_


Review: confection around centuries light pillowy citrus gelatin nuts case filberts cut tiny squares liberally coated powdered
sugar tiny mouthful heaven chewy flavorful highly recommend yummy treat familiar story lewis lion witch wardrobe treat seduces
edmund selling brother sisters witch
Summary: _START_ delight says it all  _END_


Review: looking secret ingredient robitussin believe found got addition root beer extract ordered made cherry soda flavor medic
inal
Summary: _START_ cough medicine  _END_


Review: great taffy great price wide assortment yummy taffy delivery quick taffy lover deal
Summary: _START_ great taffy  _END_


<Figure size 640x480 with 2 Axes>
```

Fig 9 Output LSTM

CHAPTER 5

Result Analysis

The corpus taken were web pages. Hence, the corpus size was undefined. Large documents were taken into consideration. After applying the extractive term frequency text summarizer over each corpus, the amount of information was almost reduced to great extents. Approximately, a summary of 115 to 180 words was generated. It covered almost all the vital information points through the full corpus. A maximum reduction of 99.81 and a minimum of 90.49 is obtained.

Input Dataset	Total word count	Summary word count	% reduction
Artificial Intelligence	29982	158	99.47
Machine Learning	9599	166	98.27
RNN	7183	160	97.77
Convolution	7252	141	99.81
TensorFlow	2056	147	92.85
Keras	758	127	98.32
NLTK	595	128	97.85
NLP	4297	159	96.30
Deep Learning	12095	176	98.54
CNN	11576	179	98.45
ANN	10273	135	98.69
Support-Vector Machine	7949	119	98.50
Random Forests	5619	138	97.54
Supervised ML	3486	143	95.90
Unsupervised ML	1638	160	90.49

Table no.1

CHAPTER 6

Conclusion and Future Scope

The abstractive method for summarization is more efficient way of computing but due to its expensiveness of time and memory complexity it serves to be difficult to be tackled for small applications. Hence, through the extractive method a corpus (web page) is condensed into a summary. For further framework in the domain, the concept can be used for topic-specific purposes. In addition to this, efforts to reduce the time and memory complexity of the encoder-decoder can be employed. Hence, efforts to build the abstractive model can be employed.

Applications and Uses

These are some use cases where automatic summarization can be used across the enterprise:

1. Media monitoring

The problem of information overload and "content shock" has been widely discussed. Automatic summarization presents an opportunity to condense the continuous torrent of information into smaller pieces of information.

2. Newsletters

Many weekly newsletters take the form of an introduction followed by a curated selection of relevant articles. Summarization would allow organizations to further enrich newsletters with a stream of summaries (versus a list of links), which can be a particularly convenient format in mobile.

3. Internal document workflow

Large companies are constantly producing internal knowledge, which frequently gets stored and under-used in databases as unstructured data. These companies should embrace tools that let them re-use already existing knowledge. Summarization can enable analysts to quickly understand everything the company has already done in a given subject, and quickly assemble reports that incorporate different points of view.

4. Financial research

Investment banking firms spend large amounts of money acquiring information to drive their decision-making, including automated stock trading. When you are a financial analyst looking at market reports and news every day, you will inevitably hit a wall and won't be able to read everything. Summarization systems tailored to financial documents like earning reports and financial news can help analysts quickly derive market signals from content.

5. Legal contract analysis

Related to point 4 (internal document workflow), more specific summarization systems could be developed to analyze legal documents. In this case, a summarizer might add value by condensing a contract to the riskier clauses, or help you compare agreements.

6. Social media marketing

Companies producing long-form content, like whitepapers, e-books and blogs, might be able to leverage summarization to break down this content and make it sharable on social media sites like Twitter or Facebook. This would allow companies to further re-use existing content.

7. Question answering and bots

Personal assistants are taking over the workplace and the smart home. However, most assistants are fairly limited to very specific tasks. Large-scale summarization could become a powerful question answering technique. By collecting the most relevant documents for a particular question, a summarizer could assemble a cohesive answer in the form of a multi-document summary.

8. Video scripting

Video is becoming one of the most important marketing mediums. Besides video-focused platforms like YouTube or Vimeo, people are now sharing videos on professional networks like LinkedIn. Depending on the type of video, more or less scripting might be required. Summarization can get to be an ally when looking to produce a script that incorporates research from many sources.

9. Medical cases

With the growth of tele-health, there is a growing need to better manage medical cases, which are now fully digital. As telemedicine networks promise a more accessible and open healthcare system, technology has to make the process scalable. Summarization can be a crucial component in the tele-health supply chain when it comes to analyzing medical cases and routing these to the appropriate health professional.

10. Books and literature

Google has reportedly worked on projects that attempt to understand novels. Summarization can help consumers quickly understand what a book is about as part of their buying process.

11. Email overload

Companies like Slack were born to keep us away from constant emailing. Summarization could surface the most important content within email and let us skim emails faster.

12. Science and R&D

Academic papers typically include a human-made abstract that acts as a summary. However, when you are tasked with monitoring trends and innovation in a given sector, it can become overwhelming to read every abstract. Systems that can group papers and further compress abstracts can become useful for this task.

13. Programming languages

There have been multiple attempts to build AI technology that could write code and build websites by itself. It is a possibility that custom “code summarizers” will emerge to help developers get the big picture out of a new project.

References

[1] A Review Paper on Text Summarization- Deepali K. Gaikwad, C. Namrata Mahender., Dr. B. A. M. U., Maharashtra, India, International Journal of Advanced Research in Computer and Communication Engineering, March 2016 (Deepali K. Gaikwad, 2016)

[2] Text Summarization: An Overview- Mr. S. A. Babar, MTech-CSE, RIT, Oct 2013 (Babar, Oct 2013)

[3] A Survey on Extractive Text Summarization- N.Moratanch, S.Chitrakala, Dept. of CSE, Anna University, CEG, IEEE International Conference on Computer, Communication, and Signal Processing, 2017 (N. Moratanch, S. Chitrakala, Dept. of CSE, Anna University, CEG, 2017)

[4] The Automatic Creation of Literature Abstracts – H. P. Luhn, IBM Journal of Research and Development, April 1958 (H. P. Luhn,, April 1958)

[5] Text Summarization with Pretrained Encoders- Yang Liu, Mirella Lapata Institute for Language, Cognition and Computation School of Informatics, University of Edinburgh, 22 Aug 2019 (Yang Liu, 22 Aug 2019)

[6] How to Write Summaries with Patterns? Learning towards Abstractive Summarization through Prototype Editing ShenGao, Piji Li, Zhangming Chan, Dongyan Zhao, Rui Yan Wang xuan Institute of Computer Technology, Peking University, China, 19 sept 2019 (ShenGao, 19 sept 2019)

[7] WikiHow: A Large-Scale Text Summarization Dataset Mahnaz Koupaee University of California, William Yang Wang University of California, Santa Barbara, Oct 18, 2018 (Mahnaz Koupaee, Oct 18, 2018)

[8] New Methods in Automatic Extraction- H. P. Edmundson, University of Maryland, Journal of the Association for Computing Machinery, April 1969 (H. P. Edmundson, April 1969)

[9] Abstractive Text Summarization using Sequence-to-Sequence RNNs and Beyond- Ramesh Nallapati, Bowen Zhou, Cicerodos Santos, IBM Watson, 2016 (Ramesh Nallapati, 2016)

[10] An Extensive Survey on Deep Learning Applications - SSSN Usha Devi N, Dr R. Mohan, Dr P. Kiran Sree, Department of Computer Science & Engineering, NITTrichy, India, Feb 2017 (Usha Devi N, Feb 2017)

[11] A Deep Learning Approach to Understanding Cloud Service Level Agreements- Srishty Saha, Karuna P. Joshi, Aditi Gupta, Renee Frank, Computer Science and Electrical Engineering University of Maryland, Baltimore County, May 24, 2017 (Srishty Saha, May 24, 2017)

[12] Abstractive and Extractive Text Summarization using Document Context Vector and Recurrent Neural Networks- Chandra Khatri (Amazon lab), Gyanit Singh (Ebay Inc.), Nish Parikh (Google), Arvix, India, July 2018 (Chandra Khatri (Amazon lab), Gyanit Singh (Ebay Inc.), Nish Parikh (Google), July 2018)

[13] Machine Learning Approach for Automatic Text Summarization Using Neural Networks- Meetkumar Patel, Adwaita Chokshi, Satyadev Vyas, Khushbu Maurya Department of Computer Engineering, GTU, Ahmedabad, India, Jan 2018 (Meetkumar Patel, Jan 2018)

[14] A Review of Text Summarization using Gated Neural Networks- Touseef Iqbal, Abhishek Singh Sambyal, Devanand, Computer Science & IT, Central University of Jammu, India, Apr 2018 (Touseef Iqbal, Apr 2018)

[15] Neural Approaches Towards Text Summarization- Abhishek Kumar Singh, Computer Science and Engineering, IIT Hyderabad, India, July 2018 (Singh, July 2018)

[16] Long Short-Term Memory Neural Computation 9(8), 1997, Sepp Hochreiter, Jurgen Schmidhuber, IDSIA, Germany (Sepp Hochreiter, 1997)

[17] Deep Learning: Methods and Applications, Li Deng, Dong Yu, Microsoft Research One Microsoft Way Redmond, WA 98052, USA, 2013 (Li Deng, Dong Yu, 2013)

[18] Improving Abstraction in Text Summarization, Wojciech Krysci ´ nski ´, KTH Royal Institute of Technology, Romain Paulus, Caiming Xiong, Richard Socher, Salesforce Research, 2018 (Wojciech Krysci ´ nski ´, 2018)

[19] An efficient single document Arabic text summarization using a combination of statistical and semantic features, Aziz Qaroush, Ibrahim Abu, Farha Wasel, Ghanem Mahdi Washaha, Eman Maalia Department of Electrical and Computer Engineering, Birzeit, University, Palestine, 26 March 2019. (Aziz Qaroush, 26 March 2019)

[20] Automatic Text Summarization Using Reinforcement Learning with Embedding Features, Gyoung Ho Lee, Kong Joo Lee, Information Retrieval & Knowledge Engineering Lab, Chungnam National University, Daejeon, Korea. (Gyoung Ho Lee)