TEXT SUMMARIZATION USING

DEEP LEARNING

A MINI-PROJECT REPORT

*By*

Benita Rego

Nolita Rego

*Under the guidance of*

Prof. Dipali Koshti



DEPARTMENT OF COMPUTER ENGINEERING

FR. CONCEICAO RODRIGUES COLLEGE OF ENGINEERING

FR. AGNEL ASHRAM, BANDRA (W),

MUMBAI - 400 050.

UNIVERSITY OF MUMBAI

(2018 – 2019)

FR. CONCEICAO RODRIGUES COLLEGE OF ENGINEERING

FR. AGNEL ASHRAM, BANDRA (W),

MUMBAI - 400 050.



CERTIFICATE

This is to certify that the following students working on the project “Text Summarization using Deep Learning” have satisfactorily completed the requirements of the project in fulfillment of the course T.E in Computer Engineering of the University of Mumbai during academic year 2019-2020 under the guidance of “Prof. Dipali Koshti”.

Submitted By: Benita Rego (8362)

Nolita Rego (8363)

|  |  |
| --- | --- |
| Prof.  Guide | Dr. B.S.Daga  Head of the Department |

\_\_\_\_\_\_\_\_

Principal

|  |
| --- |
|  |
|  |

CERTIFICATE

This is to certify that the project synopsis entitled “Text Summarization using Deep Learning” submitted by the following students is found to be satisfactory and the report has been approved as it satisfies the academic requirements in respect of mini-project work prescribed for the course.

BENITA REGO

NOLITA REGO

|  |  |
| --- | --- |
| Internal Examiner  (Signature)  Name:  Date: | External Examiner  (Signature)  Name:  Date: |

Seal of the Institute

DECLARATION OF THE STUDENT

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources.

We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea / data / fact / source in my submission.

We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Signature of the student Date:

(Benita Rego)

(8362)

Signature of the student Date:

(Nolita Rego)

(8363)

ABSTRACT

We have built a Machine Learning with Deep Learning based project “Text Summarization” using Natural Language Processing (NLP). This project summarizes an article into sentences, so when we say "text summarization," we are talking about employing machines to perform the summarization of a document or documents using some form of heuristics or statistical methods. A summary in this case is a shortened piece of text which accurately captures and conveys the most important and relevant information contained in the document or documents we want summarized. As hinted at above, there are a number of these different tried and true text summarization techniques that are currently in use and here are a few ways of going about classifying text summarization techniques. This abstract will explore these techniques from the point of view of summarization output type. In this regard, there are 2 categories of techniques: extractive and abstractive. We use abstractive summarization method 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.

Problem Statement

To build a system that takes information from the user like an article or a document and summarizes the information provided to avoid wastage of time and for good productivity.

Table of Contents

Chapter 1: Introduction…........................................................................................................ 7

Chapter 2: Literature Review…............................................................................................... 8

2.1: Natural Language Processing………………………………….………….……….8

2.2: Text Summarization……………………………………………………………….9

Chapter 3: Proposed System…................................................................................................10

3.1: Problem statement Analysis................................................................................. 10

3.2: Design and Methodology of proposed system…................................................. 11

Chapter 4: Hardware software requirements and Implementation........................................ 16

4.1: Results

4.2: Conclusion

References............................................................................................................................. 18

1. 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? Thankfully – this technology is already here.

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. The demand for text summarization systems is spiking these days thanks to the availability of large amounts of textual data. Text summarizationrefers to the technique of shortening long pieces of text. The intention is to create a coherent and fluent summary having only the main points outlined in the document. Text summarization is a common problem in machine learning and natural language processing (NLP).

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.”

NLP can also be seen as a study of Artificial Intelligence (AI). Therefore many existing AI algorithms and methods, including neural network models, are also used for solving NLP related problems.

Text summarization can broadly be divided into two categories - Extractive Summarization and Abstractive Summarization.

1. 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.

2. Abstractive Summarization: These methods use advanced NLP techniques to generate an entirely new summary. Some parts of this summary may not even appear in the original text.

Textrank is an algorithm inspired by Google’s PageRank algorithm that helps identify key sentences from a passage. The idea behind this algorithm is that the sentence that is similar to most other sentences in the passage is probably the most important sentence in the passage. Using this idea, one can create a graph of sentences connected with all the similar sentences and run Google’s PageRank algorithm on it to find the most important sentences. These sentences would then be used to create the summary.

We have used Extractive Summarization and TextRank algorithm method for this project.

2. Literature Review

2.1 Natural Language Processing

Natural Language Processing is the technology used to aid computers to understand the human’s natural language. It’s not an easy task teaching machines to understand how we communicate.

[Leand Romaf](https://www.liveedu.tv/leandromaf/ZB3b3-how-to-retrieve-images-by-visual-similarity-using-deep-lear/), an experienced software engineer who is passionate at teaching people how artificial intelligence systems work, says that “in recent years, there have been significant breakthroughs in empowering computers to understand language just as we do.”

Natural Language Processing, usually shortened as NLP, is a branch of artificial intelligence that deals with the interaction between computers and humans using the natural language. The ultimate objective of NLP is to read, decipher, understand, and make sense of the human languages in a manner that is valuable. Most NLP techniques rely on machine learning to derive meaning from human languages.

The following is a list of some of the most commonly researched tasks in natural language processing. Some of these tasks have direct real-world applications, while others more commonly serve as subtasks that are used to aid in solving larger tasks.

Though natural language processing tasks are closely intertwined, they are frequently subdivided into categories for convenience. A coarse division is given below.

a) G[rammar induction](https://en.wikipedia.org/wiki/Grammar_induction)

b) [Lemmatization](https://en.wikipedia.org/wiki/Lemmatisation)

c) [Morphological segmentation](https://en.wikipedia.org/wiki/Morphology_(linguistics))

d) [Part-of-speech tagging](https://en.wikipedia.org/wiki/Part-of-speech_tagging)

e) [Parsing](https://en.wikipedia.org/wiki/Parsing)

f) [Stemming](https://en.wikipedia.org/wiki/Stemming)

g) [Word segmentation](https://en.wikipedia.org/wiki/Word_segmentation)

h) [Terminology extraction](https://en.wikipedia.org/wiki/Terminology_extraction)

I) [Lexical semantics](https://en.wikipedia.org/wiki/Lexical_semantics)

j) [Distributional semantics](https://en.wikipedia.org/wiki/Distributional_semantics)

k) [Machine translation](https://en.wikipedia.org/wiki/Machine_translation)

l) [Named entity recognition](https://en.wikipedia.org/wiki/Named_entity_recognition) (NER)

m) [Natural language generation](https://en.wikipedia.org/wiki/Natural_language_generation)

n) [Natural language understanding](https://en.wikipedia.org/wiki/Natural_language_understanding)

o) [Question answering](https://en.wikipedia.org/wiki/Question_answering)

p) [Recognizing Textual entailment](https://en.wikipedia.org/wiki/Textual_entailment)

q) [Relationship extraction](https://en.wikipedia.org/wiki/Relationship_extraction)

r) [Word sense disambiguation](https://en.wikipedia.org/wiki/Word_sense_disambiguation)

s) [Automatic summarization](https://en.wikipedia.org/wiki/Automatic_summarization)

t) [Coreference resolution](https://en.wikipedia.org/wiki/Coreference)

u) [Discourse analysis](https://en.wikipedia.org/wiki/Discourse_analysis)

v) [Speech recognition](https://en.wikipedia.org/wiki/Speech_recognition)

w) [Speech segmentation](https://en.wikipedia.org/wiki/Speech_segmentation)

x) [Text-to-speech](https://en.wikipedia.org/wiki/Text-to-speech)

2.2 Text Summarization

Text summarization is a subdomain of Natural Language Processing (NLP) that deals with extracting summaries from huge chunks of texts. There are two main types of techniques used for text summarization: NLP-based techniques and deep learning-based techniques. In this article, we will see a simple NLP-based technique for text summarization.Text summarization is the problem of creating a short, accurate, and fluent summary of a longer text document. Automatic text summarization methods are greatly needed to address the ever-growing amount of text data available online to both better help discover relevant information and to consume relevant information faster.In this post, you will discover the problem of text summarization in [natural language processing](https://machinelearningmastery.com/natural-language-processing/). Text summarization is the process of distilling the most important information from a source to produce an abdriged version for a particular user and task.Text summarization is important, especially given the wealth of text available on the internet. You might encounter examples of text summarization you may encounter every single day. The application and promise of deep learning methods for automatic text summarization. There is an enormous amount of textual material, and it is only growing every single day. Think of the internet, comprised of web pages, news articles, status updates, blogs and so much more. The data is unstructured and the best that we can do to navigate it is to use search and skim the results. There is a great need to reduce much of this text data to shorter, focused summaries that capture the salient details, both so we can navigate it more effectively as well as check whether the larger documents contain the information that we are looking for.

Advantages are:

1. Summaries reduce reading time.

2. When researching documents, summaries make the selection process easier.

3. Automatic summarization improves the effectiveness of indexing.

4. Automatic summarization algorithms are less biased than human summarizers.

5. Personalized summaries are useful in question-answering systems as they provide personalized information.

6. Using automatic or semi-automatic summarization systems enables commercial abstract services to increase the number of texts they are able to process.

Recently deep learning methods have shown promising results for text summarization.

Approaches have been proposed inspired by the application of deep learning methods for automatic machine translation, specifically by framing the problem of text summarization as a sequence-to-sequence learning problem.

Abstractive text summarization is the task of generating a headline or a short summary consisting of a few sentences that captures the salient ideas of an article or a passage. This task can also be naturally cast as mapping an input sequence of words in a source document to a target sequence of words called summary. [Abstractive Text Summarization Using Sequence-to-Sequence RNNs and Beyond](https://arxiv.org/abs/1602.06023), 2016. These deep learning approaches to automatic text summarization may be considered abstractive methods and generate a wholly new description by learning a language generation model specific to the source documents. [Get To The Point: Summarization with Pointer-Generator Networks](https://arxiv.org/abs/1704.04368), 2017. The results of deep learning methods are not yet state-of-the-art compared to extractive methods, yet impressive results have been achieved on constrained problems such as generating headlines for news articles that rival or out-perform other abstractive methods.

The promise of the approach is that the models can be trained end-to-end without specialized data preparation or submodels and that the models are entirely data-driven, without the preparation of specialized vocabulary or expertly pre-processed source documents.

3. Report on the proposed system

This Deep Learning project uses Natural Language Processing (NLP) to summarize an article. Our system focuses on abstractive text summarization methods which employs more powerful natural language processing techniques to interpret text and generate new summary text, as opposed to selecting the most representative existing excerpts to perform the summarization.

While both are valid approaches to text summarization, it should not be difficult to convince you that abstractive techniques are far more difficult to implement. In fact, the majority of summarization processes today are extraction-based. This doesn't mean that abstractive methods should be discounted or ignored; on the contrary, research into their implementation — and true semantic understanding of human language in general — is a worthy pursuit, and much work is needed before we can confidently say that we have gained a true foothold in this endeavor.

The method first converts the paragraph into sentences then we remove all special characters from the sentences by text preprocessing. The sentence is then tokenized to get all the words that exist in a sentence. Next we need to find the weighted frequency of occurrences of all the words. The next step is to plug the weighted frequency in place of the corresponding words in original sentences and finding their sum. The final step is to sort the sentences in inverse order of their sum. The sentences give a pretty good summarization of what was said in the paragraph.

3.1. Problem Statement Analysis

Since I read news daily in the morning, I always try to keep myself updated with what’s trending and happening. However, this has proven to be a rather difficult job. Tt takes me so long to finish reading each and every news article. There are way too many resources and time is a constraint. Therefore, I decided to design a system that could prepare a bullet-point summary for me by scanning through multiple articles. This is essentially a single-domain-multiple-documents summarization task, i.e., we will take multiple articles as input and generate a single bullet-point summary. Multi-domain text summarization is not covered in this article, but feel free to try that out at your end.

3.2. Design and Methodology of proposed system

Text summarization is implemented by TextRank Algorithm. The algorithm works as follows:

\* **We import the libraries**

import numpy as np

import pandas as pd

import nltk

nltk.download('punkt')

import re

**\* Read the dataset**

df = pd.read\_csv("tennis\_articles\_v4.csv")

For quick glance of data i.e. inspection of data we can type df.head()

**\* We split the text in the articles into sentences**

sentences = []

for s in df['headline\_text']:

sentences.append(sent\_tokenize(s))

sentences = [y for x in sentences for y in x]

**\* Text cleaning by text preprocessing**

clean\_sentences = pd.Series(sentences).str.replace("[^a-zA-Z]", " ")

clean\_sentences = [s.lower() for s in clean\_sentences]

We get rid of stopwords present in the sentence by executing the following code:

nltk.download('stopwords')

from nltk.corpus import stopwords

stop\_words = stopwords.words('english')

Function to remove stopwords:

def remove\_stopwords(sen):

sen\_new = " ".join([i for i in sen if i not in stop\_words])

return sen\_new

clean\_sentences = [remove\_stopwords(r.split()) for r in clean\_sentences]

**\* To download pretrained GloVe word embeddings:**

! wget http://nlp.stanford.edu/data/glove.6B.zip

**\* Vector representation of a sentences**

Extracting word vectors:

word\_embeddings = {}

f = open('glove.6B.100d.txt', encoding='utf-8')

for line in f:

values = line.split()

word = values[0]

coefs = np.asarray(values[1:], dtype='float32')

word\_embeddings[word] = coefs

f.close()

sentence\_vectors = []

for i in clean\_sentences:

if len(i) != 0:

v = sum([word\_embeddings.get(w, np.zeros((100,))) for w in i.split()])/(len(i.split())+0.001)

else:

v = np.zeros((100,))

sentence\_vectors.append(v)

**\* For similarity matrix representation:**

We create an empty similarity matrix for this task and populate it with cosine similarities of the sentences.

sim\_mat = np.zeros([len(sentences), len(sentences)])

from sklearn.metrics.pairwise import cosine\_similarity

for i in range(len(sentences)):

for j in range(len(sentences)):

if i != j:

sim\_mat[i][j] = cosine\_similarity(sentence\_vectors[i].reshape(1,100), sentence\_vectors[j].reshape(1,100))[0,0]

**\* Applying PageRank Algorithm**

We will apply the PageRank algorithm to arrive at the sentence rankings.

import networkx as nx

nx\_graph = nx.from\_numpy\_array(sim\_mat)

scores = nx.pagerank(nx\_graph)

**\* Summary Extraction**

Finally, we extract the top 10 sentences as summary.

ranked\_sentences = sorted(((scores[i],s) for i,s in enumerate(sentences)), reverse=True)

for i in range(10):

print(ranked\_sentences[i][1])

4. Hardware software requirements and Implementation

This project has used Google Colab, a Jupyter notebook environment that requires no setup to use and runs entirely in the cloud.

The dataset is taken from Kaggle.com, which is a News Headlines csv file.

We have used the following packages:

import numpy as np

import pandas as pd

import nltk

from nltk.tokenize import sent\_tokenize

import io

Implementation:

a) Import Required Libraries

b) Read the Data

c) Inspect the Data

d) Split Text into Sentences

e) Download GloVe Word Embeddings ([GloVe](https://nlp.stanford.edu/projects/glove/) word embeddings are vector representation of words. These word embeddings will be used to create vectors for our sentences. We could have also used the Bag-of-Words or TF-IDF approaches to create features for our sentences, but these methods ignore the order of the words (and the number of features is usually pretty large)).

f) Text Preprocessing

g) Vector Representation of Sentences

h) Similarity Matrix Preparation

I) Applying PageRank Algorithm

j) Summary Extraction

Result:

We have successfully built a system that could prepare a bullet-point summary for me by scanning through multiple articles. This is essentially a single-domain-multiple-documents summarization task, i.e., we will take multiple articles as input and generate a single bullet-point summary. Multi-domain text summarization is not covered in this article, but feel free to try that out at your end.

Conclusion:

Thus, text summariztions helps reduce time contraint, works instantly and in any language, does not miss and important facts and improves productivity.

6. References:

<https://www.analyticsvidhya.com/blog/2018/11/introduction-text-summarization-textrank-python/>

<https://becominghuman.ai/text-summarization-in-5-steps-using-nltk-65b21e352b65>

<https://stackabuse.com/text-summarization-with-nltk-in-python/>

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

https://youtu.be/V6YxeKC1ZJk

https://youtu.be/ogrJaOIuBx4