NLP PROJECT

TEXT SUMMARIZER

2nd Review

By
Group - 07

Group Number 07

SAAM PRASANTH DEEVEN PEDAPALLI

(S20190010136 - saamprasanthdeevan.p19@iiits.in)

PIDAKALA ABHINANDAN BABU

(S20190010140 - abhinandanbabu.p19@iiits.in)

PREMA VIGNESH

(S20190010143 - prema.g19@iiits.in)

SRI LAKSHMI PRASANNA KONERU

(S20190010168 - srilakshmiprasanna.k19@iiits.in)

Motivation

NLP text summarizer is one of the cool applications of NLP (Natural Language Processing), which shortens the long document into a shorter one while retaining all important information from the document. Nowadays, Short news is popular everywhere because people can get the important information in very short time, hence this saves time for the people. Abstractive summarization methods aim at producing summary by interpreting the text using advanced natural language techniques in order to generate a new shorter text.

Our work plan for 1st Review

As we had mentioned earlier in the review 1, we are ahead of our work plan.

For Presentation 1 on August 31st (**COMPLETED**)

- Proposal for the project among group.
- Collecting relevant information about the proposed project.
- Presenting entire project idea as a first review. In this every one of our team is Involved.
- We should learn new topics that are relevant for this project.

Our work plan for 2nd Review

For **Presentation 2**

- We will submit a midterm report for our project with relevant documents.
- We will get a clear Information about project and number of ways to Implement this.
- We will choose the best method for summarisation and Implement it as a trail run.

Contribution

- Information relevant to the project collected by Sri Lakshmi Prasanna , Abhinandhan and Vignesh and Saam Prasanth Deeven.
- Implementation idea for Text summarisation (Spacy and heapq) by Sri Lakshmi Prasanna.
- Implementation idea for Article Summarisation by Saam Prasanth Deeven.
- Code Updates for the Text summarisation and Article Summarisation by Sri Lakshmi Prasanna, Abhinandhan babu, Vignesh and Saam Prasanth Deeven.
- Test cases done by Sri Lakshmi Prasanna, Abhinandhan babu, Vignesh and Saam Prasanth Deeven.

Information relevant to the project collected by Sri Lakshmi Prasanna , Abhinandhan and Vignesh.

- There are many ways to implement text summarization.
- Few approaches for this task are as follows.
- Text Summarization using Bert's Google model,
- Text summarization using T5 transformer model,
- Text Summarisation in nlp using nltk library in Python,
- Text Summarization in NLP using spaCy.

- Each one of us has gone through this models, among the above mentioned methods we've chosen to do our project Text Summarisation using spaCy.

Implementation idea for Text summarisation (Spacy and heapq) by Sri Lakshmi Prasanna.

- Before going into the Implementation part, Let’s have a look at spaCy !!

- We will look into spaCy and it’s salient features in the next slide

What is spaCy ?

- spaCy is designed to help you do real work to build real products, or gather real insights.
- The library respects your time, and tries to avoid wasting it.
- It's easy to install, and its API is simple and productive.
- spaCy excels at large-scale information extraction tasks.
- If your application needs to process entire web dumps, spaCy is the library you want to be using.
- Choose from a variety of plugins, integrate with your machine learning stack and build custom components and workflows.

Features of spacy

- Multi-task learning with pre-trained transformers like BERT
- Pre-trained word vectors
- State-of-the-art speed
- Production-ready training system
- Linguistically-motivated tokenization
- Components for named entity recognition, part-of-speech tagging, dependency parsing, sentence segmentation, text classification, lemmatization, morphological analysis, entity linking and more

Condt ..

- Easily extensible with custom components and attributes
- Support for custom models in PyTorch, TensorFlow and other frameworks
- Built in visualizers for syntax and NER
- Easy model packaging, deployment and workflow management
- Robust, rigorously evaluated accuracy

Text Summarisation using spaCy.

- So, what’s next ? The big question is how we Implemented it.

- Steps taken to Implement this Text Summarization are :

1. Text cleaning
2. Sentence Tokenization
3. Word Tokenization
4. Word frequency table
5. Summarisation

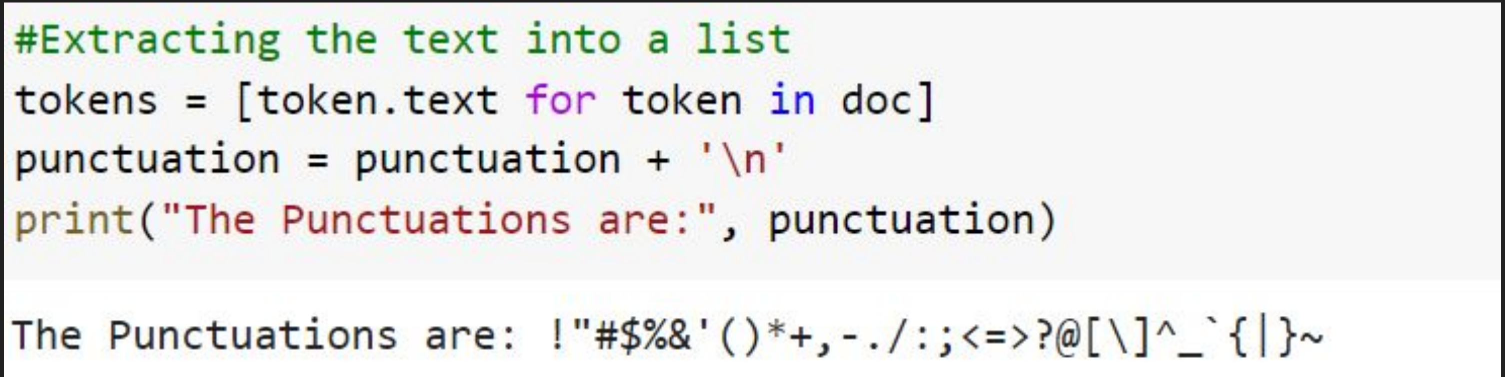
Text Cleaning

- Text cleaning is a very crucial step, without the cleaning process the dataset is often a cluster of words that the computer doesn’t understand. There are several steps to clean the data.
- Text cleaning is the process of preparing raw text for NLP (Natural Language Processing) so that machines can understand human language.
- There are some steps in Text Cleaning that are mentioned in the next coming slides.

Text Cleaning

Step 1: Punctuation

- The title text has several punctuations.
- Punctuations are often unnecessary as it doesn’t add value or meaning to the NLP model.
- The “string” library has 32 punctuations.
- The punctuations are:



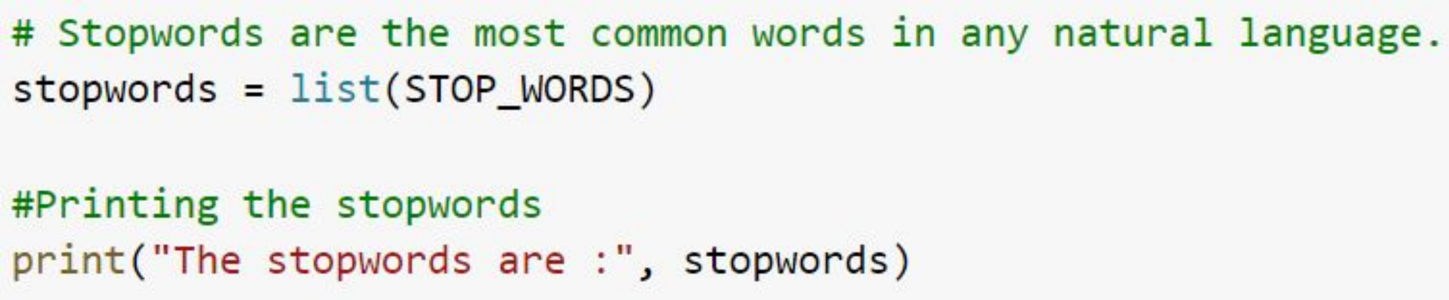
```
#Extracting the text into a list
tokens = [token.text for token in doc]
punctuation = punctuation + '\n'
print("The Punctuations are:", punctuation)
```

```
The Punctuations are: !"#$%&'()*+,-./:;<=>?@[\]^_`{|}~
```

Text Cleaning

Step 2 - Stop words

- Stop words are the irrelevant words that won’t help in identifying a text as a real or fake.
- Now, we have a list of words without any Punctuations. Let’s get a head and remove the stop words.



```
# Stopwords are the most common words in any natural language.
stopwords = list(STOP_WORDS)

#Printing the stopwords
print("The stopwords are :", stopwords)
```

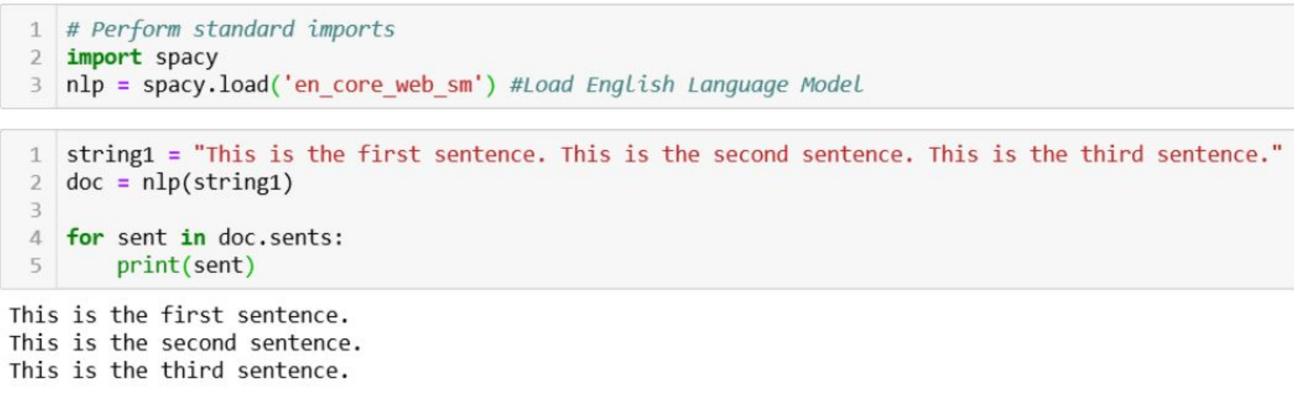
Text Cleaning

Step 3 - Tokenization

- Tokenizing is the process of splitting strings into a list of words.
- Tokenization is important because the meaning of the text could easily be interpreted by analyzing the words present in the text.
- we can utilize the awesomeness of spaCy to perform tokenization.
- We will use spacy.lang.en which supports the English language.

Sentence Tokenization

- Sentence Tokenization is the process of identifying different sentences among group of words.
- Spacy library designed for Natural Language Processing, perform the sentence segmentation with much higher accuracy.
- So with this basic idea, we would say that we can split the string based on dot and get the different sentences. However it keeps reading until it reaches the dot which actually ends the sentence.



```python
# Perform standard imports
import spacy
nlp = spacy.load('en_core_web_sm') #Load English Language Model
```

```python
string1 = "This is the first sentence. This is the second sentence. This is the third sentence."
doc = nlp(string1)

for sent in doc.sents:
    print(sent)
```

```
This is the first sentence.
This is the second sentence.
This is the third sentence.
```

Word Tokenization

- In Spacy, the process of tokenizing a text into segments of words and punctuation is done in various steps.
- It processes the text from left to right.
- First, the tokenizer split the text on whitespace similar to the split() function.
- Then the tokenizer checks whether the substring matches the tokenizer exception rules.

Word Frequency table

- One of the key steps in NLP is the ability to **count** the frequency of the terms used in a text document or table.
- To achieve this we must tokenize the words so that they represent individual objects that can be counted.

Implementation idea for Article Summarisation by Saam Prasanth Deeven.

- For this Article Summarisation, we have Implemented with the help of spaCy.
- In the earlier slides it is clearly mentioned about spaCy and about it Salient features.
- There are many ways to Implement this Article Summarization.
- Some of the techniques are Basic data manipulation and visualization, Rouge scoring, Tokenization, Word embeddings and Neural network architectures like convolutional neural networks and recurrent neural networks

Thank you