Evaluation of Python Text Summarization Libraries

1Tripti Sharma, 2Anmol Ashri, 3Navin Kumar, 4Shubham Pal, 5Rajat
1Professor, 2Student, 3Student, 4Student, 5Student
Maharaja surajmal institute of technology

Abstract - **Text summarization is a process of condensing the source data into a concise version of text while preserving important information content and overall meaning. In this study we have analysed the accuracies of summaries provided by the NLTK,spaCy,Gensim,Sumy and compare those summaries with original summary provided by the dataset as reference summaries in order to calculate the BLEU score for the assessment of the summaries provided by these python libraries. Dataset used, is taken from kaggle.com, dataset contains 4515 different examples with original summaries of each example. All the results and experimentation made using Python 3 in Jupyter Notebook.**

keywords - **Text Summarization,NLTK,spaCy,Gensim,Sumy,BLEU**

1. *Introduction*

Text summarization [9] is a job of constructing a short and voluble summary storing essential data and complete meaning conveyed by the original text. It allows the readers to swiftly and easily recognized the content of the entire document without need of read the entire document. The entire purpose of text summarization is to project the essence of text by using a smaller number of words and sentences. For instance, search engines produce fragments as the sample of the documents [4]. Another example is of news websites which show headlines to facilitate browsing. It is very difficult and tedious, because when summaries produced by us(humans), we read it thoroughly then make summary of essential information. However, computers do not have human understanding and linguistic capabilities, it makes text summarization a laborious task.

Text summarization can be of two types [1], extractive summarization and abstractive summarization [5] [6]. Extractive summarization comprises of choosing essential sentences, paragraphs from original document and abridged into condense configuration. The importance of sentences is based on statistical and linguistic characteristics of sentences. Abstractive method focuses at making essential material in a unique way. In simple words, they translate and examine with the help natural language processing producing unique shorter summary which expresses the most significant information from indigenous document.

Extractive summaries [2] are produced by extracting essential text fragments from the text based on statistical inspection of single or mixed characteristics like word/phrase frequency, location or cue words to locate the sentences to be extracted. The "most treasured" content is made as the most as the "most frequent" or the "most likely placed" content. Such approach ignores any efforts on thorough text understanding.

Extractive summarization processes [3] is split into two steps: 1) Pre-processing step and 2) Processing step. Pre-processing is organized form of indigenous text. It includes: A) Sentences breaking recognition for example in language (English for example), sentence limits recognised with existence of "." (dot) at finish of sentence. B) Stop word removal-Regular words with no semiotics and which do not comprise of relevant data to the context are removed. 3) Stemming: The work of stemming is to gain the stem of all words, which focuses its semiotics. In processing step, characteristics impact the germane of sentences are calculated and then weights allocate to attributes using method (weight learning). End Score of all sentences is calculated using equation (Feature-weight). Upper rank sentences are chosen for end summary.

Consequently, in this paper we are comparing accuracies of text summaries provided by python libraries SpaCy [11], Gensim [10], NLTK [7], Sumy [8] with the original summary provided in the dataset [12]. Further this paper is divided into following sections. Section 2 discusses past works in text summarization, Section 3 discusses the detailed description of all python libraries, Section 4 will illustrate the dataset used for calculating results and scoring technique which is used for calculating accuracies of text summaries given by python libraries. Experimental outcomes are discussed in section 5. Section 6 will conclude the paper.

2. *Text Summarization Past Works*

Curiosity in automatic text summarization, arose initially in mid twentieth century. An essential paper published in 1958, proposed to use the weight the sentences from a document as a method of high redundant words [13], overlooking the very high redundant common words. Automatic text summarization system [14] in 1969, which with combination used the regular keyword method (frequency depending weights), in addition the following methods are also used for calculating the sentence weights:

1. Cue Feature: This feature or method retains the words which are to be included in end summary.

2. Title word method: Words mentioned in title and sub-heading are included in final summary as these words represents theme of indigenous text.
3. Sentence Word location method: This feature or method based on supposition that words occurring at initial position in sentence or paragraph have high significance
In 1995 the Trainable Document Summarizer [15] do the sentence extracting task, established on a number of techniques based on weighting. Following features were used and measured:
1. Sentence Length (Cut-O Feature): In this sentence are not included in the abstract which contains the words less than pre-defined number.
2. Fixed-Phrase Feature: sentences which contain certain cue words and phrases are involved.
3. Paragraph method: this method or feature similar to Sentence Word location methods in [14]
4. Thematic Word Feature: thematic words or most redundant words. Sentence scores are methods of the thematic word's redundancies.
5. Uppercase Word Feature: uppercase method in which letters such as "WHO" are regarded as prime words which tend to be included in end sentences.
A compilation of documents was used in this process which contained 188 documents with their summaries from 21 prestigious publications. Degree of resemblance between sentences found in produced by professional maestros compared with indigenous text which came out to be 80%, which is the huge majority of synopsis sentences could be called direct matches. Since then, many works have been published tackling the text summarization problems (reference [30,31] provides more information about techniques and work done in text summarization field).

3. *Python Libraries Description*

3.1 *Gensim*

Gensim is a python library dependent on NumPy [21] and SciPy libraries. Its specialities are the memory independence where there is no requirement for the whole training compilation to be on the RAM at any point in time, It provides Simplistic usage for Latent Semantic Analysis [18] and Latent Dirichlet Allocation [19], also gives the access to write simple similarity queries for documents in their semantic representation, it is based heavily on the theory of a corpus, vector, models and sparse matrices. Going by the definitions of these concepts: a corpus is an entire dataset given in a compressed or uncompressed single file path, a vector is a set of defining attributes for the corpus represented in word-by-word vector value, a model is an algorithm used to extract and find pattern in the document matrix represented by the sparse matrices. It uses TextRank [20] algorithm for text summarization.

3.1.1 *GenSim workflow*

The methods to process documents are so different and application- and language-dependent, a document is defined by the features obtained from it. It is not defined by its string form. It is using the bag of words technique. In this form, each document is viewed as one vector where each vector element presents a Q-A pair.
Several file formats are there for serializing a Vector Space to disk. Gensim implements them using its streaming corpus interface. Documents are processed from disk in a lazy fashion, one by one.It does read the whole corpus into main memory at once. It converts documents from one vector form to another. It helps in bringing out the hidden structure of the corpus. It helps in finding similarity between words to use them to define the documents in a new and better way. It also helps in making the document compact and improves efficiency.
Transformations often change in between two specific vector spaces. For training and for successive vector transformations we should use the same vector space. If the same vector space is not used then it will cause feature mismatch during transformation calls.

3.2 *NLTK*

NLTK [7] stands for Natural Language Toolkit. It was developed jointly with a computational linguistics course at the University of Pennsylvania in the year 2001 by Edward Loper and Steven Bird. It is the most popular python library used to work with human language data.
It contains a variety of text processing libraries for classification, tokenization, stemming, tagging, parsing etc. It uses TF-IDF [22] algorithm for text summarization NLTK involves follow steps:

1)Tokenize the sentence: it divides a text into a series of tokens.

2) Matrix of all words sentence by sentence is created called frequency matrix. It stores the number of times a word is appeared in each sentence.

3)Calculate Term Frequency and generate a matrix:

$$TermFrequency(p) = \frac{P}{D} \quad 1$$

Where

P is defined as how many times a particular term (term p in equation 1) appears in document/data.

D is number of entire terms in document/data.

Here, the document/data represents paragraph, the term represents word in a paragraph.

4) Table is created which contains documents per defined words.

5)Calculate IDF (Inverse Document Frequency) [22] and generate a matrix:

$$IDF(t) = \log \frac{N}{t} \quad (2)$$

Where

t is defined as how many times a particular term (term t in equation 2) appears in document/data.

N is number of entire terms in document/data.

Here, the document/data represents paragraph, the term represents word in a paragraph

6) Multiply the values from both matrix in order to estimate TF-IDF [22] and produce new matrix.

7) Score the sentences by giving weights to the paragraph using TF-IDF [22] score.

3.3 *SpaCy*

SpaCy was developed by Matthew Honnibal and Ines Montani, the founders of the software company Explosion. It is widely used for teaching and research. spaCy provides software for production usage. Some of the spaCy main significant features are: convolutional neural network[23] models for part-of-speech tagging[24], dependency parsing[25], text categorization[26], and named entity recognition [27].

3.3.1 *Library Architecture*

The spaCy has two data structures named Doc and Vocab. The Doc object has the sequence of tokens and their annotations. The Vocab has a HashMap to query the common information quickly in documents. It centralized string, word vectors & lexical attributes to avoid redundancy. This saves memory and reduces inconsistency. The tokenizer is used to make the Doc object, then it is modified in place by the pipeline's component. Language Object has these components and also orchestrates the training and serialization.

3.3.2 *The pre-processing pipeline*

In the spaCy pre-processing, the text is first tokenized to produce a Doc object. Doc is then processed in several different steps which is called a pipeline.

3.3.2.1 *Tokenizing text*

In spaCy tokenization is the work of dividing a text into befitting components, called tokens. These small components could be words, punctuations, numbers, or some other special characters that may form a sentence. Input is generally a Unicode text and the output is a Doc object.

3.3.2.2 *Tagging*

In this part, the doc is tensorized by encoding the internal representation of the doc into an array of floats. This is required as spaCy is a neural model and only speaks tensors. In this, each token of a sentence is marked with an appropriate part of speech such as nouns, verbs.

3.3.2.3 *Named entity recognition*

This is the last part of the pipeline. In this, the spaCy perform named entity recognition in which an entity (a real-world object) is assigned a name. For example, PERSON for people, ORG for agencies,companies etc, LANGUAGE used for any language.

3.3.2.4 *Ruled based matching*

By default, spaCy's pipeline also perform rule-based matching. This annotates tokens with additional information and is considered important during pre-processing. The custom rules can be added to this part of the pipeline.

3.4 *Sumy*

Sumy is a python package for automatic summarization of text documents and HTML pages.It uses multiple algorithms and methods for summarization.

1) USING LEX-RANK [29]
It is an unsupervised approach to summarise text focused on graph-based centrality scoring of sentences. It uses the approach of finding similar sentences which will likely be of great importance. To install use pip install lexrank.
For ex:- # Using LexRank

```
summarizer = LexRankSummarizer()
#Summarize the document with 2 sentences
summary = summarizer(parser.document, 2)
```

2) USING LUHN [28]
Luhn is a heuristic method for summarising text. It is based on the frequency of most important words.
For ex:- from sumy.summarizers.luhn import LuhnSummarizer
summarizer_luhn = LuhnSummarizer()

```
summary_1 =summarizer_luhn(parser.document,2)
```

3) USING LSA[18[
LSA stands for Latent Semantic Analysis. It is uses term frequency techniques with a singular value break up to summarize text.
For ex:- from sumy.summarizers.lsa import LsaSummarizer
summarizer_lsa = LsaSummarizer()

```
summary_2 =summarizer_lsa(parser.document,2)
```

4. *Dataset and Scoring technique*

4.1 *Dataset*

In this paper the dataset [12] used is taken from kaggle.com. The dataset consists of 4515 examples which consist of Authors name, Headlines, URL of article, Summary and full article. Summary contained in these examples are gathered from Inshorts and only scraped articles from prestigious and authentic news agencies such as Hindu, Indian Times and Guardian. This dataset contains articles ranges from February to august 2017. Accuracies of summary given by python libraries are calculated by comparing with original summary provided by the dataset.

4.2 *Scoring Technique*

Bilingual Evaluation Understudy [16] (BLEU) is an algorithm used for estimating the quality of text which is translated from one language(natural) to another using machine language. Quality is measure of communication between a machine's result and that of a human: "the closer a machine translation is to a professional human translation, the better it is" –BLEU is based on this statement. This technique within different metrics claims one of greater association with qualities [17] judged by human evaluations. It remains as most economical and favoured mechanized metrics.
To calculated the overall quality of translations, scores are calculated for discreet transcribed segments by comparing with a good quality of examples (reference translations). After this step, the scores are averaged with whole compilation. BLEU score always ranges between 0 and 1. Higher the value, higher will be similarity of the candidate text to reference text.

5. *Experimental Results*

We have taken six different text from dataset [12] with their original summary as shown in table 1 which provided by the dataset. These six different texts are given as inputs to each python library Sumy,NLTK,Gensim,SpaCy. Each library summarizes the text individually then summary provided by each library compared to the original summary using BLEU scoring technique. BLEU score is calculated as shown in table 1. Average of these scores is calculated as shown in table 2. The library which has highest BLEU score will provide better summary.

Table 1. Representation of BLEU score of each library with original summary and headlines.

Headlines	Summary	SUMY	NLTK	GENSIM	SPACY
Malaika slams user who trolled her for 'divorcing rich man'	Malaika Arora slammed an Instagram user who trolled her for "divorcing a rich man" and "having fun with the alimony". "Her life now is all about wearing short clothes, going to gym or salon, enjoying vacation[s]," the user commented. Malaika responded, "You certainly got to get your damn facts right before spewing sh*t on me...when you know nothing about me."	0.2487939831 6482115	0.080332653 48702884	0.04023759 7032697	0.0842030 769026514 8
'Virgin' now corrected to 'Unmarrie	The Indira Gandhi Institute of Medical Sciences (IGIMS) in Patna on Thursday made corrections in its Marital Declaration Form by changing 'Virgin' option to 'Unmarried'. Earlier, Bihar Health Minister	0.0416316906 35454625	0.089332603 7825818	0.08933260 37825818	0.0450412 007458820 54

d' in IGIMS' form	defined virgin as being an unmarried woman and did not consider the term objectionable. The institute, however, faced strong backlash for asking new recruits to declare their virginity in the form.				
Aaj aapne pakad liya: LeT man Dujana before being killed	Lashkar-e-Taiba's Kashmir commander Abu Dujana, who was killed by security forces, said "Kabhi hum aage, kabhi aap, aaj aapne pakad liya, mubarak ho aapko (Today you caught me. Congratulations)" after being caught. He added that he won't surrender, and whatever is in his fate will happen to him. "Hum nikley they shaheed hone (had left home for martyrdom)," he added.	0.0880084235 4501276	0.2006637291 9054257	0.0459180384195934	0.1606963979536032 4
Hotel staff to get training to spot signs of sex trafficking	Hotels in Maharashtra will train their staff to spot signs of sex trafficking, including frequent requests for bed linen changes and 'Do not disturb' signs left on room doors for days. A mobile phone app called Rescue Me, which will allow staff to alert police of suspicious behaviour, will be developed. The initiative has been backed by the Maharashtra government.	0.2212376831 8690065	0.5063446673040602	0.20140704210713187	0.2342465967624591 8
Man found dead at Delhi police station, kin allege foul play	A 32-year-old man on Wednesday was found hanging inside the washroom of a Delhi police station after he was called for interrogation. His family alleged that he could have been emotionally and physically tortured. Police said the man was named as a suspect in the kidnapping case of a married woman with whom he had been in a relationship earlier.	0.1226265909 024768	0.2919899731923766	0.23847803864963912	0.2045736134437221 6
Delhi HC reduces aid for 'negligent' accident victim by 45%	The Delhi High Court reduced the compensation awarded to a motor accident victim by 45% after it found negligence on the part of both parties. A compensation of ?10 lakh was earlier awarded to the victim. The court observed, "It's possible despite the vehicle being driven in permissible limit, an accident can occur when a jaywalker suddenly appears on road."	0.1294560146 941643	0.2398253876 0449895	0.17886801218313006	0.1161356425493082 8

Table 2. Average BLEU score of each Library.

	SUMY	NLTK	GENSIM	SPACY
Average	0.141959	0.234748	0.158653	0.114536

From Table 2 we can analyse that NLTK has the highest average BLEU score and SpaCy has lowest BLEU score. Hence for the dataset taken in this paper, NLTK provides better text summary then rest of the library used in this paper.

6 *Conclusion*

In this paper, we have studied and focused on presenting the analysis of summaries given by python text summarization libraries-NLTK,spaCy,Sumy,Gensim. Comparing those summaries with original summaries provided by the dataset, further calculating the BLEU score for each summary provided by the libraries individually and then averaging those values, we found that NLTK gives highest average BLEU score while spaCy gives least BLEU score. Hence, for the examples provided by the dataset, NLTK gives better summary then other python text summarization libraries.

References

[1] Richa Sharma, Prachi Sharma, "A Survey of Extractive Text Summarization", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 6, 2016.

[2] Farshad Kyoomarsi, Hamid Khosravi, Esfandiar Eslami and Pooya Khosravyan Dehkordy, "Optimizing Text Summarization Based on Fuzzy Logic", In proceedings of Seventh IEEE/ACIS International Conference on Computer and Information Science, IEEE, University of Shahid Bahonar Kerman, UK, 347-352, 2008.

[3] Vishal Gupta, G.Sl Lehal, "A Survey of Text Mining Techniques and Applications", Journal of Emerging Technologies in Web Intelligence, VOL. 1, NO. 1, 60-76, AUGUST 2009.
[4] Andrew Turpin, Yohannes Tsegay, David Hawking, and Hugh E Williams. 2007. Fast generation of result snippets in web search.In Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 127–134.
[5] G Erkan and Dragomir R. Radev, "LexRank: Graph-based Centrality as Salience in Text Summarization", Journal of Artificial Intelligence Research, Re-search, Vol. 22, pp. 457-479 2004.
[6] Udo Hahn and Martin Romacker, "The SYNDIKATE text Knowledge base generator", Proceedings of the first International conference on Human language technology research, Association for Computational Linguistics , ACM, Morristown, NJ, USA , 2001.
[7] Edward Loper and Steven Bird. NLTK: The Natural Language Toolkit.
[8] Chintan Shah and Dr. Anjali Jivani. Multi-document summarization: study on existing techniques.
[9] N. Moratanch and S. Chitrakala, "A survey on abstractive text summarization," in Circuit, Power and Computing Technologies (ICCPCT), 2016 International Conference on. IEEE, 2016, pp. 1-7.
[10] Islam. Akef and Juan S. Munoz Arango. "Mallet vs GenSim: Topic modeling for 20 news groups report".
[11] Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing.
[12] https://www.kaggle.com/sunnysai12345/news-summary
[13] H. P. Luhn, "The Automatic Creation of Literature Abstracts", Presented at IRE National Convention, New York, 159-165, 1958.
[14] H. P. Edmundson.," New methods in automatic extracting", Journal of the ACM, 16(2):264-285, April 1969.
[15] J. Kupiec, J. Pedersen, and F. Chen, "A trainable document summarizer", In Proceedings of the 18th ACMSIGIR Conference, pages 68-73, 1995.
[16] Papineni, K.; Roukos, S.; Ward, T.; Zhu, W. J. (2002). "BLEU: a method for automatic evaluation of machine translation ". ACL-2002: 40th Annual meeting of the Association for Computational Linguistics. pp. 311–318.
[17] Callison-Burch, C., Osborne, M. and Koehn, P. (2006) "Re-evaluating the Role of BLEU in Machine Translation Research" in 11th Conference of the European Chapter of the Association for Computational Linguistics: EACL 2006 pp. 249–256.
[18] Peter Foltz, "Latent Semantic Analysis for Text-Based Research", Pearson Inc.
[19] David M. Blei, Andrew Y. Ng, Michael I. Jordan, "Latent Dirichlet Allocation", Journal of Machine Learning Research 3.
[20] Tanwi, Satanik Ghosh, Viplav Kumar, Yashika S Jain, Mr. Avinash." Automatic Text Summarization using Text Rank".
[21] Stéfan van der Walt, S. Chris Colbert, Gael Varoquaux, "The NumPy array: a structure for efficient numerical computation".
[22] Rahim Khan, Yurong Qian, Sajid Naeem, "Extractive based Text Summarization Using K-Means and TF-IDF".
[23] Saad Albawi; Tareq Abed Mohammed; Saad Al-Zawi ,"Understanding of a convolutional neural network".
[24] Yuan Tian; David Lo; "A comparative study on the effectiveness of part-of-speech tagging techniques on bug reports".
[25] Joakim Nivre," Dependency Parsing".
[26] M. IKONOMAKIS, S. KOTSIANTIS, V. TAMPAKAS," Text Classification Using Machine Learning Techniques.
[27] Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. A Survey on Deep Learning for Named Entity Recognition".
[28] Khalid Waleed Hussein , Dr. Nor Fazlida Mohd. Sani, Professor Dr. Ramlan Mahmod, Dr. Mohd. Taufik Abdullah," Enhance Luhn Algorithm for Validation of Credit Cards Numbers.
[29] Gunes Erkan, Dragomir R. Radev. LexRank: Graph-based Lexical Centrality As Salience in Text Summarization".
[30] Günes Erkan and Dragomir R Radev. 2004. LexRank: Graph-based lexical centrality as salience in text summarization. J. Artif. Intell. Res.(JAIR) 22, 1 (2004), 457–479.
[31] Rasim Alguliev, Ramiz Aliguliyev. Effective summarization method of text documents.