International Journal of Future Generation Communication and Networking
Vol. 13, No. 3, (2020), pp. 1645–1652

Text Summarization Using Spacy Algorithm

Vijay. M.N[1], Vignesh. R.R[2], Sivaprasath. V[3], Tamilalakan. S[4], Indhu. R[5*]
Computer Science and Engineering,
KPR Institute of Engineering and Technology, Coimbatore, India
**Corresponding author*

Abstract
Creating quick summaries of Documents is obtaining Selective information from an authentic text document. The extracted information is obtained as a summarized report and consulted as a concise reduction to the user. It is unconditionally sharp-witted for us to value and to describe the content of the text. The extractive summarization advances focus on choosing how paragraphs, essential sentences, etc., creates the progressive documents in exact form and presents a summary that only contains parts of the original document. The experience of epitome resides in having identifying and presenting the key entities in the document. It aims at creating an extractive trimming of multiple documents and enables us to find the relevance of the contents in those documents. This is enabled beside a user interface to bit a query on set of multiple documents and present the most relevant documents in the order .Simple machine learning algorithms are used to accomplish this and the performance examination of the system could the advancement of research activities further to do the same as abstractive summarization using deep neural networks.

Keywords: *Summarization, Machine learning, Tokenization, Algorithms, Spacy*

I. INTRODUCTION

Text Summarization is a strategy used to give shorter outline of a long Text. The significant points and non-redundant content which are huge among the contents recovered from the accessible enormous pool of text [1][3]. In this time of innovation, information gets produced each second in the web. Before the finish of 2025, the information will grow up to 175 ZB (Zettabytes), evaluated by the IDC (International Data Corporation). With such a lot of data streaming in the web, there is a need to create Machine Learning calculations to sum up the substance. Presently individuals don't find a chance to read all the substance in the web. So, summarization plays a significant job in the web by news summarization, meanings of the technical terms, and so forth., The proposed AI framework utilizes Machine Learning for text summarization [2][10]. The following are classifications of Machine Learning Supervised Learning, Unsupervised Learning, Reinforcement Learning, Semi-managed Learning, Feature Learning, Self-Learning and Sparse Dictionary Learning. Supervised Learning model can be developed by historical data (some arrangement of training data). Unsupervised Learning model can have just input data and by gathering the cluster the input data dependent on the model new patterns are created [4][7]. Reinforcement Learning gives experimentation idea by learning itself (trial and error) as what human does. Tkinter is a framework in python that can be utilized to create the GUI for the system. By incorporating the Tkinter and Machine Learning model the Document Summarizing AI System is made. This proposed AI System gives the information in a brief structure by extricating from voluminous measure of information [8][9]. It offers an interesting assistance that can be utilized in the News summarization [5][6], Article Summarization, and so forth., in future.

II. OBJECTIVE

A definitive point of the system is to sum up enormous measure of data. The data could be in any structure, for instance, text, PDF and Webpage URL. The summarized content doesn't miss the huge real factors of the file. It is equipped for working in any platforms. Summarizes the data faster.

ISSN: 2233-7857 IJFGCN

III. TEXT SUMMARIZATION METHODS

(A) SUPERVISED LEARNING

Supervised learning is a manner by which oversight is given to the machine utilizing information that is very much labeled [11][13]. In Supervised learning, input variable X and a yield variable Y are utilized with a calculation to decide the mapping capacity of the contribution to the yield.

$$Y = f(X)$$

The article keeps on assessing the mapping capacity so exact that when you own new information (x) that you can forecast the yield factors (Y) for that information [12]. The Machine taking in design that gains from the old information and makes new forecasts as yield is called directed learning.

Naive Bayes

Naive Bayes utilizes the methodology of AI, the model trains the classifier and predicts the yield depends on the figuring of particular worth decay (SVD). Prior to preparing the model, it needs two ideas of recursive component disposal and SVD-highlight positioning [13][15]. In this strategy, the preparation dataset is utilized as a kind of perspective and the rundown procedure is displayed as a characterization issue: sentences are isolated as outline sentences and non-synopsis sentences dependent on the highlights that they have. The arrangement probabilities are found out factually from the preparation information, utilizing Bayes' standard: where, s is a sentence from the report assortment, F1, F2, F3...FNs are qualities utilized in a grouping. S is the abstract to be designed and P (s< S | F1, F2, F3., FN) is the plausibility that sentences will be separated to shape the frame given that it has highlights F1, F2, F3…., FN.

(B) REINFORCEMENT LEARNING

Reinforcement learning can think and act like human cerebrum. This Reinforcement Learning is tied in with choosing a reasonable activity to expand compensation for a specific circumstance [14][16]. It is planned by different calculations to locate the most ideal conduct or way it should take in a particular event. It is an experimentation. Without the preparation dataset, it is compelled to gain from its experience.

Markov Decision Process

The numerical methodology for mapping an answer in support Learning is recon as a Markov Decision Process or (MDP). This methodology can likewise be called as a discrete time stochastic control process. It tends to be applied in different fields, for example, data building, creation, analyzing and financial aspects [17][19]. By utilizing this methodology, we can be tackled by utilizing dynamic programming and fortification learning.

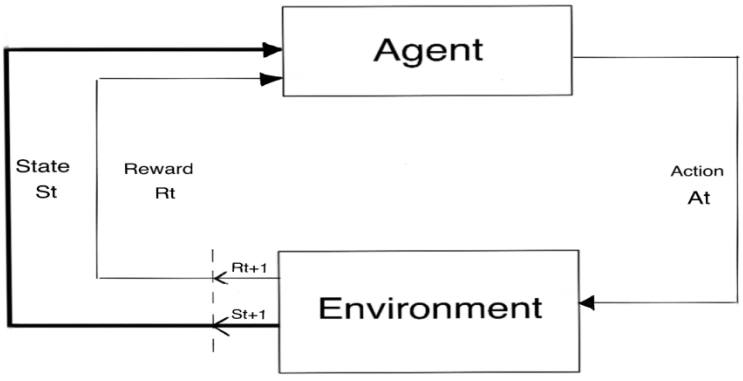

Figure 1: Markov Decision Process

ISSN: 2233-7857 IJFGCN

(C) UNSUPERVISED LEARNING

Unsupervised learning is a sort of AI calculation used to draw designs from input sets. Bunch investigation is the most well-known technique to distinguish concealed examples or gathering in the information [18][20]. The bunch model distinguishes the comparable list by estimating the Euclidean or probabilistic separation self-sorting out maps utilize neural systems that gain proficiency with the topology and dispersion of the information [22][24]. In Hidden Markov models, observed information is utilized to recuperate the succession of states.

K-Means Clustering

Unsupervised learning k-implies is probably the least complex calculation that takes care of the notable grouping issue [21][23]. This calculation essentially arranges the given information into various groups (accept k bunches) which is to be given earlier. The system thought is to characterize k loci, one for each cluster. These centroids ought to be put in the right manner on the grounds that a distinctive area causes a diverse outcome. In this way, putting them as much as far perhaps will improve the outcomes [25]. The subsequent stage from a piece of given information takes each point and partners it to the closest centroid. To this time when no point is pending, the fundamental step is achieved and an advanced groupage is finished.

IV. PROPOSED SYSTEM

(A) SYSTEM IMPLEMENTATION

The Summarizer system developed accepts three kinds of inputs. One is direct text input; the second way is web URL and the third way is Files in text format. It processes the input by spacy summarizer algorithmic means and produce the summarized text as output. The output is of two forms. One is direct text and the other is the result stored in the files in the text format. The Spacy summarizer implementation screenshots are attached below.

ISSN: 2233-7857 IJFGCN

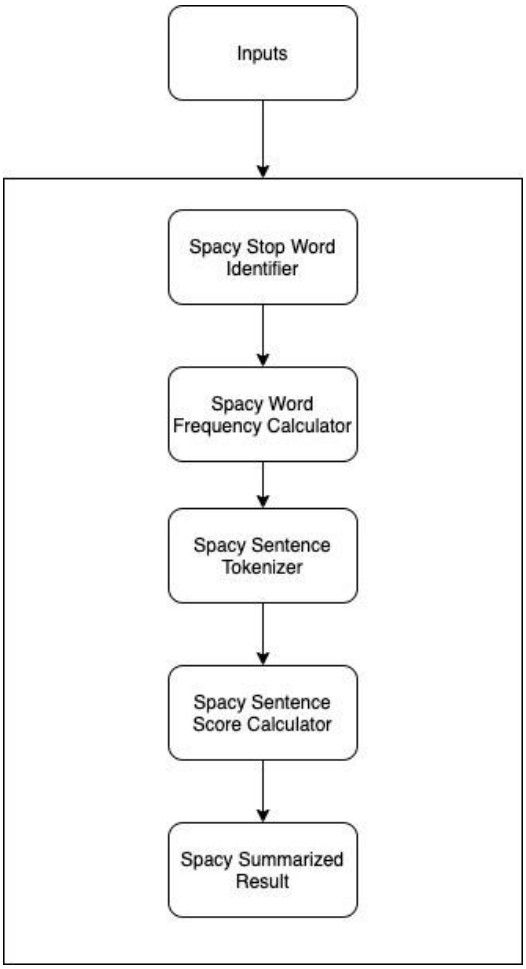

Figure 2: Architecture of Spacy Summarizer System

The content has a great deal of stop words and it should be disposed of in light of the fact that it isn't required while summing up a book. The python bundles have a rundown of stop words and those prevent words from these contents are expelled. At that point the word recurrence should be resolved. A word reference is made where the words are keys and whose qualities are refreshed with the no of times it happens in the content. At that point the given content contains sentences and those sentences are tokenized. At that point dependent on the recurrence of the words the sentence scores are resolved. In this way sentences are tokenized and their scores are determined. At that point dependent on the score of the sentence the outlines are created. The sentence having the greatest score are considered for summing up the content. At that point the summed-up result is given as yield to the yield window.

Summarization Module

The synopsis module gets the content as information. The procedures can be classified as,
(i) Stop Word Identification
(ii) Word Frequency Determination
(iii) Sentence Tokenization
(iv) Sentence Score Determination
(v) Spacy summarized Result

The content has a great deal of stop words and it should be disposed of as it isn't required while summing up a book. The python bundles have a rundown of stop words and those prevent words from these

ISSN: 2233-7857 IJFGCN

contents are expelled. At that point the word recurrence should be resolved. A word reference is made where the words are keys and whose qualities are refreshed with the no of times it happens in the content. At that point the given content contains sentences and those sentences are tokenized. At that point dependent on the recurrence of the words the sentence scores are resolved. In this manner sentences are tokenized and their scores are determined. At that point dependent on the score of the sentence the synopses are created. The sentence having the greatest score are considered for summing up the content. At that point the summed-up result is given to the applet window.

Results

The summed-up text is acquired as an applet from the spacy module. At that point the summarized result is made accessible to the clients in two modes and they are as per the following.

(i) Summed up Text Output in the applet window

(ii) Summed up Text File

The summed-up text is shown in the output window to the people as output. At that point the summed-up text can be spared as a document. We use document tasks to compose the summed-up text yield to the records. The records are named dependent on the report number, date and time.

TABLE I: Comparison of Algorithms

ALGORITHM	**ORIGINAL TEXT** (in number of words)	**SUMMARIZED TEXT** (in number of words)
SPACY	1032	280
GENSIM	1032	336
SUMY	1032	310

From the above table named as Table I, we see that there are four calculations which are looked at dependent on their synopsis. A book of length 1032 words is considered for rundown. The outline is finished utilizing four calculations and every calculation delivers a summed-up text. Among the four calculations we discovered Spacy is best which delivers a shorter summed up text of length 280 words and it is significant also.

(B) LIBRARIES USED IN THE AI SYSTEM

Spacy

Progressed in NLP (Natural language handling) [3] written in python and cython (Open-source library). It is explicitly intended for creation use programming and used to assemble this NL (Natural Language) Artificial System. Spacy offers a few highlights are free, more adaptable than other measurable models. It gives an assortment of phonetic explanations to give you text's linguistic structure from bits of knowledge. This AI framework utilizes unaided Learning with the assistance of spacy.

Tkinter

It is the standard library (GUI) for python used as a front-end applet provider in the Document Summarizing AI System. Tkinter provide a powerful OO (object-oriented) interface to Tk (tkinter) GUI tk (toolkit) and include with standard Microsoft windows, Linux, mac installs of python. Tk (tkinter) provides text boxes, buttons, labels (In GUI application) and these are commonly called as widgets which can be implemented as a front-end applet in the AI system.

Tokenizer

In the AI system using tokenizer to split large contents into smaller parts like sentences to words, paragraphs to sentences such as words, keywords, phrases, symbols and other elements called tokens [23][25]. Two types of tokenizer can be used, one is tokenizer for words and other is tokenizer for sentences. A few characters like punctuation are expelled. In parsing and text mining have input from tokens. In a words or sentence tokenizer break text into tokens whenever sees any whitespace. Finally, the tokenized content will be stored in a separate list.

ISSN: 2233-7857 IJFGCN

V. CONCLUSION

This system targets changing over the bigger content documents into a shorter summed up text which contains the significant and important data. This undertaking helps the understudies and clients to sum up the sites. So, their riding time will be less and the learning time will get improved. Embracing proficient philosophies in the everyday tasks will build their profitability. We can concentrate more on significant perspective on task while utilizing productive systems. It encourages us to search for progressively important data in the midst of the exceptionally enormous assortment of information in the Internet which will set aside tremendous effort to peruse every one of them. The time taken to record the summed-up text will be off the image since we have the choice to spare the summed-up text as a document. We discovered Spacy calculation is best since it produces shorter and important rundowns.

VI. FUTURE WORK

In future updates, we add the productivity of the AI system and it tends to be exceptionally valuable in the field of programmed news synopsis. It can likewise use in the API to give indexed lists in a summed-up structure. The efficiency of the calculations can be improved dependent on the comprehension of the area explicit information. The future progressions in the NLP which is an exploration region will permit us to make outlines which are increasingly similar to the words conveyed by a human. It will be linguistically right and significant simultaneously. We can stretch out this usefulness to different record designs having text content.

REFERENCES

[1] Darling, W.M. and F. Song, 2011. Probabilistic document modeling for syntax removal in text summarization. Proceedings of the 49th Annual Meeting of the Association for computational linguistics, (CL' 11), ACM Press, Stroudsburg, PA., pp: 642-647.
[2] Goldstein, J., V. Mittal, J. Carbonell and M. Kantrowitz, Multi-document summarization by sentence extraction. Proceedings of the NAACL-ANLP Stroudsburg, PA, USA., pp: 40-48.DOI: 10.3115/1117575.1117580.
[3] M. Haque et al. "Literature Review of Automatic Multiple Documents Text Summarization", International Journal of Innovation and Applied Studies, Vol. 3, pp. 121-129, 2013.
[4] https://web.stanford.edu/~jurafsky/slp3/23.pdf
[5] Gu¨nes Erkan and Dragomir R. Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. J. Artif. Int. Res. 22(1):457–479.
[6] Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into texts. In Dekang Lin and Dekai Wu, editors, Proceedings of EMNLP 2004. Association for Computational Linguistics, Barcelona, Spain, pages 404–411. http://www.aclweb.org/anthology/W04- 3252.
[7] Kai Hong and Ani Nenkova. 2014. Improving the estimation of word importance for news multi-document summarization. In Proceedings of the 14th Conference of the Euro- pean Chapter of the Association for Computational Linguistics. Association for Com- putational Linguistics, Gothenburg, Sweden, pages 712–721.
[8] Hui Lin and Jeff Bilmes. 2011. A class of submodular functions for document summariza- tion. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1. Association for Computational Linguistics, Stroudsburg, PA, USA, HLT '11, pages 510–520.
[9] Dan Gillick and Benoit Favre. 2009. A scalable global model for summarization. In Pro- ceedings of the Workshop on Integer Linear Programming for Natural Langauge Pro- cessing. Association for

ISSN: 2233-7857 IJFGCN

Computational Linguistics, Stroudsburg, PA, USA, ILP '09, pages 10–18. http://dl.acm.org/citation.cfm?id=1611638.1611640.
[10] Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. In Proceedings of the 54th Annual Meeting of the Association for Computa- tional Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, Berlin, Germany, pages 484–494.
[11] Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In Proceed- ings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.. pages 3075–3081.
[12] Daraksha Parveen and Michael Strube. 2015. Integrating importance, non-redundancy and coherence in graph-based extractive summarization. In Proceedings of the 24th Interna- tional Conference on Artificial Intelligence. AAAI Press, IJCAI'15, pages 1298–1304.
[13] Xun Wang, Masaaki Nishino, Tsutomu Hirao, Katsuhito Sudoh, and Masaaki Nagata. 2016. Exploring text links for coherent multi-document summarization. In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Tech- nical Papers. The COLING 2016 Organizing Committee, Osaka, Japan, pages 213–223.
[14] Saranyamol C S and Sindhu L, "A Survey on Automatic Text Summarization", International Journal of Computer Science and Information Technologies, Vol. 5(6), pp. 7889-7893, 2014.
[15] James Clarke and Mirella Lapata. 2006. Models for sentence compression: A compari- son across domains, training requirements and evaluation measures. In Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics. Association for Computa- tional Linguistics, Stroudsburg, PA, USA, ACL-44, pages 377–384.
[16] James Clarke and Mirella Lapata. 2008. Global inference for sentence compression: An in- teger linear programming approach. Journal of Artificial Intelligence Research 31:399– 429.
[17] Katja Filippova and Michael Strube. 2008. Sentence fusion via dependency graph com- pression. In Proceedings of the Conference on Empirical Methods in Natural Lan- guage Processing. Association for Computational Linguistics, Stroudsburg, PA, USA, EMNLP '08, pages 177–185. http://dl.acm.org/citation.cfm?id=1613715.1613741.
[18] Katja Filippova. 2010. Multi-sentence compression: Finding shortest paths in word graphs. In Proceedings of the 23rd International Conference on Computational Linguistics. As- sociation for Computational Linguistics, Stroudsburg, PA, USA, COLING '10, pages 322–330. http://dl.acm.org/citation.cfm?id=1873781.1873818.
[19] Florian Boudin and Emmanuel Morin. 2013. Keyphrase extraction for n-best reranking in multi-sentence compression. In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics, Atlanta, Georgia, pages 298– 305. http://www.aclweb.org/anthology/N13-1030.
[20] Siddhartha Banerjee, Prasenjit Mitra, and Kazunari Sugiyama. 2015. Multi-document ab- stractive summarization using ilp based multi-sentence compression. In Proceedings of the 24th International Conference on Artificial Intelligence. AAAI Press, IJCAI'15, pages 1208–1214. http://dl.acm.org/citation.cfm?id=2832415.2832417.
[21] Elaheh ShafieiBavani, Mohammad Ebrahimi, Raymond K. Wong, and Fang Chen. 2016. An efficient approach for multi-sentence compression. In Proceedings of The 8th Asian Conference on Machine Learning. PMLR, The University of Waikato, Hamilton, New Zealand, volume 63 of Proceedings of Machine Learning Research, pages 414–429. http://proceedings.mlr.press/v63/ShafieiBavani24.html.
[22] Dung Tran Tuan, Nam Van Chi, and Minh-Quoc Nghiem. 2017. Multi-sentence Com- pression Using Word Graph and Integer Linear Programming, Springer International Publishing, Cham, pages 367–377. https://doi.org/10.1007/978-3-319-56660-332.

ISSN: 2233-7857 IJFGCN

[23] Rafael Ferreira et al. “Assessing Sentence Scoring Techniques for Extractive Text Summarization”, Elsevier Ltd., Expert Systems with Applications 40 (2013) 5755-5764.
[24] Vimal Kumar K, Divakar Yadav “An Improvised Extractive Approach for Hindi TextSummarization” Springer India 2015
[25] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vec- tors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, Doha, Qatar, pages 1532–1543. http://www.aclweb.org/anthology/D14- 1162.

ISSN: 2233-7857 IJFGCN