Retraction

Retraction: A Comparative Analysis of Pre-Processing Time in Summary of Hindi Language using Stanza and Spacy (IOP Conf. Ser.: Mater. Sci. Eng. 1110 012019)

Published 10 June 2022

This article has been retracted by IOP Publishing/the authors following an allegation that this article contains tortured phrases [1], masking overlap without citation [2,3,4]. IOP Publishing has investigated in line with the COPE guidelines, and agree this article should be retracted.

IOP Publishing wishes to credit PubPeer commenters [5] and the Problematic Paper Screener for bringing the issue to our attention.

The authors agree to this retraction.

[1] Cabanac G, Labbe C, Magazinov A, 2021, arXiv:2107.06751v1
[2] Desai N and Shah P, 2016, Automatic Text Summarization using Supervised Machine Learning Technique for Hindi Language IJRET, 5 6
[3] Sargule S and Kagalkar R.M, 2016, Strategy for Hindi Text Summarization using Content Based Indexing Approach IJCSE 4 9
[4] Qi P, Zhang Y, Zhang Y, Bolton J, Manning C.D, 2020, Stanza: A Python Natural Language Processing Toolkit for Many Human Languages, Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics
[5] https://pubpeer.com/publications/F00151A8CE7B8D6B4896A6AF306FE9

Retraction published: 10 June 2022
A Comparative Analysis of Pre-Processing Time in Summary of Hindi Language using Stanza and Spacy

Atul Kumar*, Dr Vinodani Katiyar*, Dr Pankaj Kumar*

*Research Scholar, DSMNRU Lucknow, India, atulverma16@gmail.com
bDSMNRU Lucknow, India, drvinodani@gmail.com
cSRMGPC, Lucknow, India, pk79jan@gmail.com

Abstract. Text summarization is a method or say way for converting large texts into smaller ones by keeping all important points of the larger text as it is in the smaller ones and giving the output in the modified form called summary of original text document. This task is very difficult for humans as it required large amount of rigorous analysis of the document. In this proposed paper we are comparing pre-processing time for two tools of natural language processing one is STANZA and the other is SPACY both are based on modern technologies and are examine for HINDI language processing. In this paper we are performing a comparative study of both the tools on the bases of their pre-processing time for processing HINDI language.

Now a days, a lot of text summarizer tools are present in the environment and we are keen to use them but here the point of this paper comes into light that how we know which Summarizer is fast enough to get us the same accuracy.

Keywords— Stanza, Spacy, Pre-processing time, Summarizer, Natural Language Processing, Tokenization, Segmentation, Stemming, Lemmatization.

1. Introduction

In this present scenario information is very important thing exists. Billions of data is floating on the internet every second, but that data also includes a lot of information which are not important in order to make it further use in any domain so, as a solution to this problem TEXT SUMMERIZER came into picture, it helps the people to get the information they required at that time by eliminating unwanted words in that text or document. Text summarization is used by a lot of applications like for instance, scientists require a tool to provide summaries for figuring out whether or not to study the entire record(text) or now no longer and for summarizing facts searched through person on Internet. News corporations can use different document summarizer to group the data from various sources and summarize that. In this proposed paper we are providing a tool by which the user can get the fast output so that it can increase the productivity of their domain.

2. The Related Work

There exist a great deal of summarizers over the web, yet a significant number of them uphold just two dialects that are English and European dialects. The first of the main work in robotized text rundown was given by Luhn [1] in 1958. To produces outlines he utilizes frequencies of words by given regarded contribution with the item to decide the contiguity of a sentence in an info.
P.B. Baxendale [2] in 1958 proposed a novel element that is sentence area or sentence area position in an info report. It is investigated that sentences which are situated toward the start or toward the finish of the report are more significant than different sentences in the report. A sentence position include turned into a basic component for sentence extraction and it is utilized till now.

H.P. Edmundson [3] in 1969 proposed a novel structure for text rundown. He proposed two new highlights. To begin with, prompt words that are the presence of most demonstrative words into a archive, for example, at long last, in a rundown, ultimately, and so forth. Second, title or heading Words that is an extra weight was allocated to a sentence, if sentences have heading words in it.

Afterward, Julian Kupiec [4] in 1995 proposed some new highlights like sentence length, presence of capitalized words, express structure, and furthermore included different highlights created beforehand. Likewise, portrayed another strategy of the rundown with an innocent Bayes classifier, the grouping capacity orders each sentence regarding if it is extraction commendable.

Gupta Vishal and Gurpreet Singh Lehal [5] in 2010, In which extractive summarization method consists of selecting important sentences, paragraphs etc. from the original document and concatenating them into shorter form.

Jawline Yew Lin [6] in 1999 additionally attempted to show a framework for sentence extraction utilizing choice trees for making useful conventional/question situated concentrates. Jawline Yew Lin recognized various highlights; for example, : title, pattern, position, tf-idf, question signature, sentence length, lexical network, citation, first sentences, formal person, place or thing, pronoun and descriptor, mathematical information, non-weekend days, and month. The score for all highlights was joined with programmed learning utilizing choice trees and blend work. Lin directed a profound investigation of various highlights impacts through a glass-box. The test result shows that there was not the single component which performs well for question-based rundown. Highlights like mathematical information give better outcomes for the question which requires an answer in mathematical worth, while non-weekend days and month include gives the best outcome for inquiries like "when?"

Conroy and O’leary [7] made a framework for sentence extraction utilizing two procedures named QR and HMM. In QR procedure the significance of the sentence was determined and the sentences having higher significance were added to the rundown. At that point, an overall score of the excess sentences was modified on the grounds that a portion of the leftover sentences was repetitive. Furthermore, the other procedure was the concealed Markov model (HMM) which is a consecutive model for programmed text outline. What's more, in HMM just three highlights were utilized: sentence position, complete no of terms in the sentence, and the likeness of sentence terms in the given archive. Eventually, the assessment was finished by looking at HMM produced outline with the human-created rundown.

S. P. Yong et al [8] in 2005 portrayed a computerized framework that had the capacity of learning by joining diverse later approaches, for example: a factual methodology, sentences extraction and neural organization.
Ladda Suanmali et al [9] in 2009 proposed a framework based on the fluffy rationale. Fluffy guidelines and fluffy sets were utilized for separating huge sentences dependent on sentence highlights.

Different highlights were utilized to register sentence importance and the worth was given somewhere in the range of „0” and „1” if a specific element was available in the sentence. The element utilized was title word in a sentence, sentence centrality, likeness to initially sentence, watchword, sentence length, sentence position, formal person, place, or thing and numeric information. The estimations of these highlights were given as a contribution to a fluffy framework, which made a yield dependent on highlights and IF-THEN principles characterized by sentence extraction.

Li Chengcheng [10] in 2010 proposed a novel ATS strategy in light of Rhetorical Structure Theory (RST). RST is the documentations of expository connection introduced between two non-overlapping writings called the core (N) and satellite (S). The trial perceptions were utilized to communicate the distinction among N and S, which was communicated by N that what is more imperative to the writer’s reason than S and the Logical core was intelligible autonomous of the S. The whole content was separated into little units called sentences dependent on delimiter like a full stop, commas, any accentuation mark found between sentences.

The whole cycle was based on a diagram, sentences which were less significant was taken out from the diagram and remaining sentences were summed up.

S. Harabagiu and F. Lacatusu [11] propose a method that, given a catchphrase question, on the fly creates new pages, called formed pages, which contain all inquiry watchwords. The created pages are produced by getting and assembling related pieces from hyperlinked Web pages and holding connects to the first Web pages. To number the created pages, we consider both the hyperlink structure of the first pages and the relationship between the catchphrases inside each page. Besides, they introduce and tentatively assess heuristic calculations to productively create the top formed pages.

In R. Varadarajan, V. Hristidis, and T. Li [12] proposed an assessment mining and outline strategy utilizing various methodologies and assets, assessing every one of them thusly. Their work incorporates the improvement of the extremity arrangement segment by utilizing AI over-explained corpora and different strategies, for example, an anaphora goal.

In E. Lloret, A. Balahur, M. Palomar, and A. Montoyo [13] present the main report of programmed estimation rundown in the legitimate space. This work depends on preparing legitimate inquiries with a framework containing a self-loader Web blog search module and Fast Sum, a total programmed extractive multi-record assumption synopsis framework. They give quantitative assessment aftereffects of the rundowns utilizing legitimate master analysts. They report benchmark assessment results for question-based opinion synopsis.

In J.G. Conrad, J.L. Leidner, F. Schilder, and R. Kondadadi [14], the creator proposed a novel calculation for sentiment outline that assesses substance and cognizance, all the while. They think about a synopsis as a succession of sentences and straightforwardly get the ideal grouping from numerous survey archives by pulling and requesting the sentences. They accomplish this with a novel Integer Linear Programming (ILP) definition. The framework in this paper is a ground-breaking combination of the Maximum Coverage Problem and the Traveling Salesman Problem and is generally pertinent to message age and outline.
In H. Nishikawa, T. Hasegawa, Y. Matsuo, and G. Kikui [15] writers characterize an assignment called subject life structures, which sums up and relates the centrepieces of a theme transiently with the goal that per users can comprehend the substance without any problem. The proposed model, called TSCAN, determines the significant subjects of a point from the eigenvectors of a fleeting square affiliation framework.

3. Types of Text Summarization

We have two types of text summarization which are as follows:

3.1 Extractive Summarization:

Extractive summarization can be proposed as a classification subject. Its main target is to take out the most relevant sentences and paragraph from that respective text or data and ranked them high. The highest ranked regions from all text (data) can then combined and re-rank using same aspects and append them into smaller form. Extractive summarization uses statistical way or say technique to select important sentences or keyword from text (data).

Extractive summarization involves mentioned points:

A. Pre-processing step
B. Processing step.

Pre-processing is a well arranged representation of the original document. It involves three sub processes:

1. Sentence segmentation: In this method sentence’s boundary are find out and it is find out with the presence punctuation marks.

2. Stop-Word Removal: Stop-words and the words which do not provide relevant information to the subject are removes.

3. Stemming: Reason for stemming is to get the root word by eliminating prefix and suffix.

In processing phase, characteristic influencing the significance of sentences are decided and calculated, and after that weights are assigned to these characteristics using weight learning method. Final score of every sentence is calculated using Feature-weight equation and those sentences are chosen for the final summary whose scores are highest among all the sentences in the original document.

3.2 Abstractive Text Summarization:

Abstractive text summarization provides the summary after rigorous analysis of the provided document and design the summary using only required words and eliminate the rest of the unwanted words in order to make the text sort and easy to read and fast to understand.

In this method summarization of every text of the sentence is implemented and have different way as compared to the original document.

Abstractive summarization method can be divided into two approaches:

A. Structured approach
B. Semantic approach.

Structure approach uses various techniques to give the required result that are tree technique, template technique, ontology technique, lead and body phrase technique and rule technique.
Semantic technique uses techniques that are Multi-modal Semantic model, Information item technique, and semantic graph technique.

[1] Summarization Technique
As we are performing Abstractive method in pre-processing time-based comparative study on STANZA and SPACY here are the common techniques which both tools follows:

a. Pre-Processing:
   In the pre-processing phase of these tools, the text is firstly divided into list of sentences and then, that sentences are additionally divided into words and after that common words are eliminated.
   Pre-processing method includes three steps:
   1) Segmentation
   2) Tokenization
   3) Stop words elimination.

1. Segmentation:
   Here sentences are segmented, based on sentence boundary which can be predefined or as well as user defined too. In Hindi language, sentence boundary is identified by “|” which is known as full stop in English language. On every full stop, the sentences are fragmented and place into a list. The final yield of this phase is the list of sentence and this list is send for next level processing.

2. Tokenization:
   In this phase, all the sentences are divided into words. In Hindi language, this process is performed by finding out the space separation and commas between the words. After that a list of words is formed and are called tokens. And this list is send for next level processing.

3. Stop-words Elimination:
   Generally, maximum commonly used words are referred to as stop words and they are not required for the text and hence are removed in this stage.
   So in this way these common words should be eliminated from the original text because if we do not perform at this stage then the weight of these words grows maximum as can also affect the final output. From having prior observations that each Hindi textual content statistics has minimal 25% to 35% or maybe extra stop words. A Sample of stop-words are "के", "है", "और", "नहीं" etc.

b. Processing phase:
   Processing phase is the most important phase in text summarization in any respective language. In this phase, feature value for each sentence is determined. And on bases of certain functions the final summary is being generated.

Stanza
Stanza is an open-source Python NLP toolkit which had a huge variety of sixty six human talking languages.). Stanza toolkit entails a language-agnostic completely neural pipeline for textual content examining, it has tokenization, multiword token expansion, lemmatization, part-of speech and morphological function tagging, dependency parsing, and named entity recognition (NER).
4. Design and Architecture:

Stanza have of two components:
1. Fully neural multilingual NLP pipeline,
2. Python client interface to the Java Stanford Core NLP software.

4.1 Neural Multilingual NLP Pipeline:

Stanza’s neural pipeline has models that can perform operation from tokenizing raw text to accomplishing syntactic examination on complete sentences.

Segmentation and Tokenization:
Tool performs segmentation and then tokenization and makes a list respectively, Stanza combines tokenization and sentence segmentation from given input into one module and is additional named as tagging problem over character sequences, where the model guesses whether a given character is the end of a token, end of a sentence, or end of a multi-word token.

Multi-word Token Expansion:
It is obtained with an ensemble of a frequency lexicon and a neural sequence-to-sequence model, to make sure that regularly ascertained expansions within the training set are invariably robustly enlarged whereas maintaining flexibility to model unseen words statistically.

POS and Morphological Feature Tagging:
Stanza assigns it part of speech to every word mentioned in the list, and observes its universal morphological features.

Lemmatization:
Stanza also performs lemmatization on every word in a sentence to retrieve its recognized form. Lemmatization is obtained as a collaboration of a dictionary-based lemmatizer and a neural seq2seq lemmatizer.

Dependency Parsing: Here every word in the sentence is allotted a syntactic head as Stanza parses every sentence of its syntactic structure.

Named Entity Recognition (NER):
Stanza also recognizes named entities in the given input in order to generate the required text summary.

4.2 Core NLP Client:

Stanford’s Java Core NLP toolkit offers a collection of NLP tools but the drawback is that they are not smoothly accessible in python language, which is the majority choice of data scientists. But by applying certain methods we can make it flexible for python use too.

Spacy
Spacy is used for advanced natural language processing and is also open source so anyone can use it within just a few clicks. Spacy performs excellent for named entity recognition As time passes developers made spacy to train on more than one language especially in version 2.0 and after.
5. Implementation of Spacy

Its implementation phrase in python programming language consists of the following points.

a. Import libraries like (numpy, pandas, spacy, sklearn, string),
b. Providing input document,
c. Defining boundaries and performing segmentation,
d. Performing Tokenization,
e. Performing TF-IDF functionality,
f. Analysing top sentences,
g. Generating final summary.

6. Performing Comparison

In this phase we are going to perform a comparison of pre-processing time of both the tools, for this we considered 10 different Hindi Text Documents of size starting from 10,000 to 1,00,000 words. So, in this way we have 10 different observations and are quite sufficient to test whose pre-processing time is fast.

The processing of comparison involves, firstly import the respected dataset into the program, then for computing pre-processing time we imported time python library. We implemented this function at starting and end of the preprocessing step in order to compute the pre-processing time. After that separate readings of the text is taken for both the tools starting from ten thousand to one lakh word length. The pre-processing time is analysed in units of seconds i.e. the time required in the execution of pre-processing of the various text documents. After performing this operation on the entire dataset of text documents the following readings are found as shown in the table mentioned below:

| Text size (words) | Stanza (in sec) | Spacy (in sec) |
|-------------------|----------------|----------------|
| 10000             | 8.908458       | 0.049776       |
| 20000             | 19.81356       | 0.062253       |
| 30000             | 24.997198      | 0.081768       |
| 40000             | 32.280505      | 0.109722       |
| 50000             | 38.622927      | 0.127685       |
| 60000             | 46.119452      | 0.153165       |
| 70000             | 55.102151      | 0.171214       |
| 80000             | 63.901981      | 0.201384       |
| 90000             | 74.714428      | 0.221147       |
| 100000            | 86.042275      | 0.246014       |
Fig 5.1: Comparison Graph

*The results may slightly vary on systems with different configurations of hardware.*

7. Conclusion

The final conclusion of this paper is that the pre-processing time of Stanza is very high as compared to Spacy, as it is clear from the above observation. But we can’t forget that Stanza supports 66 languages whereas Spacy supports only 61 languages as of now. So, if the language you want to work on is supported by Spacy then it’s better to go with spacy as it performs about 300 times faster as compared to Stanza. The pre-processing time in stanza increases with increase in the number of words in the input text. Both have their merits and demerits depending upon the place they are used but it’s a better option to go with Spacy wherever it is applicable.

References

[1] Hans Peter Luhn. “The automatic creation of literature abstracts,” IBM Journal of research and development, 2(2):159–165, 1958.

[2] P. B. Baxendale, “Machine-made Index for Technical Literature - An Experiment,” Journal of Research and IBM Development, vol. 2, no. 4, pp. 354-361, October 1958.

[3] Harold P Edmundson. “New methods in automatic extracting,” Journal of the ACM (JACM), 16(2):264–285, 1969.

[4] Julian Kupiec, Jan Pedersen and Francine Chen, “A trainable document summarizer,” Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval, ACM, pp. 68-73, 1995.

[5] Gupta Vishal and Gurpreet Singh Lehal. “A survey of text summarization extractive techniques,” Journal of Emerging Technologies in Web Intelligence, vol. 3, pp.258-268, 2010.

[6] Chin-Yew Lin, “Training a selection function for extraction,” Proceedings of the eighth international conference on Information and knowledge management, ACM, pp. 55-62, 1999.

[7] M.J Conroy and O'leary, “Text summarization via hidden markov models,” In Proceedings of SIGIR '01, pp.406-407, 2001.

[8] Md Haque, Suraiya Pervin and others, “Literature Review of Automatic Single Document Text Summarization Using NLP,” International Journal of Innovation and Applied Studies, vol.3, no.3, pp. 857-865, 2013.

[9] Ladda Suanmali, Mohammed Salem Binwahlan and Naomie Salim, “Sentence features fusion for text summarization using fuzzy logic,” Ninth International.
[10] Li Chengcheng, “Automatic Text Summarization Based On Rhetorical Structure
Theory,” International Conference on Computer Application and System Modeling
(ICCASM), vol. 13, pp. 595-598, October 2010.
[11] S. Harabagiu and F. Lacatusu, “Using Topic Themes for MultiDocument
Summarization,” ACM Trans. Information Systems, vol. 28, pp. 13:1-13:47, July
2010.
[12] R. Varadarajan, V. Hristidis, and T. Li, “Beyond Single-Page Web Search Results,”
IEEE Trans. Knowledge and Data Eng., vol. 20, no. 3, pp. 411-424, Mar. 2008.
[13] E. Lloret, A. Balahur, M. Palomar, and A. Montoyo, “Towards Building a Competitive
Opinion Summarization System: Challenges and Keys,” Proc. Human Language
Technologies: The 2009 Ann. Conference of the North Am. Ch. Assoc. for
Computational Linguistics, Companion Vol.: Student Research Workshop and
Doctoral Consortium, pp. 72-77, 2009.
[14] J.G. Conrad, J.L. Leidner, F. Schilder, and R. Kondadadi, “Query-Based Opinion
Summarization for Legal Blog Entries,” Proc. 42th Int’l Conf. Artificial Intelligence
and Law, pp. 167-176, 2009.
[15] H. Nishikawa, T. Hasegawa, Y. Matsuo, and G. Kikui, “Opinion Summarization
with Integer Linear Programming Formulation for Sentence Extraction and Ordering,”
Proc. 23rd Int’l Conf. Computational Linguistics: Posters, pp. 910-918, 2010.