A RESULTANT OF STANZA AND SPACY FOR HINDI TEXT SUMMARIZATION

1st Durgesh Pandey
Department of Computer Engineering
Ambalika Institute of Management and Technology
Lucknow, INDIA
durgeshpandey733@gmail.com

2nd Kamal Srivastava
Department of Computer Engineering
Ambalika Institute of Management and Technology
Lucknow, INDIA
2007.srivastava@gmail.com

3rd Nitin Kushwaha
Department of Computer Engineering
Ambalika Institute of Management and Technology
Lucknow, INDIA
strangenk66@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 preform ing 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, Hindi text summarizer, Text summarizer, Natural Language Processing(NLP).

1. INTRODUCTION

In this present scenario information is very important thing exists . Billions of data is floating on the internet every seconds 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 produce summaries for deciding whether to read the full document(text) or not and for summarizing data searched by user on Internet. News groups can use different document summarizer to group the data from various sources and summarize that. In this proposed paper we are providing by which tool the user can get the fast output so that can increase the productivity of their domain.

2. TYPES OF TEXT SUMMARIZATION

We have two technique in text summarization which are as follows:

2.1. 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 point features influencing the relevance of sentences are decided and calculated and then weights are assigned to these features using weight learning method. Final score of each sentence is determined 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.

2.2. Abstractive Text Summarization:

Abstractive text summarization provide the summary after rigorous analysis of the provided document and designing 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 which are tree technique, template technique, ontology technique, lead and body phrase technique and rule technique.

Semantic technique use techniques which are Multi-modal Semantic model, Information item technique, and semantic graph technique.

3. 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:

3.1. Pre-Processing:

In the pre-processing stage of these tools, the text is firstly divided into list of sentences, then that sentences are further divided into words and then common words are eliminated.

Pre-processing method includes three steps:

1) Segmentation
2) Tokenization
3) Stop words elimination.

3.1.1. Segmentation:

In this stage 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 sentence boundary, the sentences are broken and place into list. The final output of this stage is the list of sentence and this list is send for next level processing.

3.1.2. Tokenization:

At this stage, all the sentences are divided into words. In Hindi language, sentences are tokenized by finding out the space separation and commas between the words. So the list is formed, which has elements as words and are called tokens. And this list is send for next level processing.

3.1.3. Stop-Words Elimination:

Generally, most commonly used words are called stopwords. These common words are not necessary for the text and hence are removed in this stage.

So in this way these common words should be removed from the original text because if we do not perform this stage then the weight of these words grows maximum as can also affect the final output. From having prior observations that every Hindi text data has minimum 25% to 35% or even more stopwords.

Example of stop-words are "के", "है", "और", "नहीं" etc.

3.2. Processing phase:

Processing phase is the most important phase in text summarization in any respective language. In processing stage, value of feature for every sentence is calculated. And on bases of certain functions the final summary is being generated.

4. STANZA

Stanza is an open-source Python natural language processing toolkit which had a wide range of 66 human speaking languages. Stanza toolkit involves a language-agnostic fully neural pipeline for text examining, it has tokenization, multiword token expansion, lemmatization, part-of speech and morphological feature tagging, dependency parsing, and named entity recognition (NER).

4.1 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.

4.1.1 Neural Multilingual NLP Pipeline:

Stanza's neural pipeline has models that can perform operation from tokenizing raw text to performing syntactic analysis on entire 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 a single module and is further named as tagging problem over character sequences, where the model predicts whether a given character is the end of a token, end of a sentence, or end of a multi-word token predicts 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 ensure that frequently observed expansions in the training set are always robustly expanded while 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 each word in a sentence to get back its canonical form. Lemmatization is obtained as an ensemble of a dictionary-based lemmatizer and a neural seq2seq lemmatizer.

Dependency Parsing: Here each word in the sentence is assigned 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 CoreNLP Client:

Stanford's Java CoreNLP tookit provides a set of NLP tools but the drawback is that these tools are not smoothly accessible from python programming language , which is the majority choice of data scientists. But by applying certain methods we can it flexible for python use too.

5. SPACY

Spacy is used for advanced natural language processing and is also open source so anyone can use it with in just few clicks. Spacy perform 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.

5.1 Implementation of Spacy:

Its implementation phrase in python programming language consists of following mentioned 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) Analyzing top sentences,
g) Generating final summary.

6. PERFORMING COMPARISION

In this phase we are going to perform 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 is quit 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 pre-processing step in order to compute the pre-processing time.

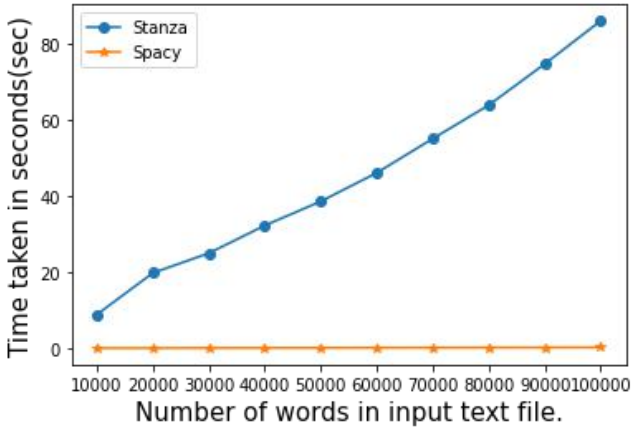
After that separate readings of the text is taken for both the tools starting from ten thousand to one lakh word length. The pre-processing time is analyzed in unite of seconds i.e. the time required in the execution of pre-processing of the various text documents. After performing this operation on the entire dataset of text documents the following readings are found as shown in the table mentioned below:

Table 1. Comparison of pre-processing time in Stanza & Spacy

Text size (words)	Stanza (in sec)	Spacy (in sec)
10000	8.906458	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

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 11 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.

8. REFERENCES

[1]. A Review Paper on Text Summarization for Indian Languages by Bijal Dalwadi , Nikita Patel , Sanket Suthar.

[2]. N. R. Kasture1, Neha Yargal "A Survey on Methods of Abstractive Text Summarization" International Journal For Research In Emerging Science And Technology, Volume-1, Issue-6, November-2014.

[3]. Vishal Gupta, Gurpreet Singh Lehal,"A Survey of text summarization of extractive techniques" Journal of Emerging Technologies In Web Intelligence, Vol. 2, No. 3, August 2010.

[4]. Sonia Haiduc1, Jairo Aponte,"On the Use of Automated Text Summarization Techniques for Summarizing Source Code " 2010 17th Working Conference on Reverse Engineering.

[5]. Nikita Munot Sharvari S. Govilkar "Comparative Study of Text Summarization Methods International Journal of Computer Applications" (0975 – 8887) Volume 102– No.12, September 2014.

[6]. Richa Sharma, Prachi Sharma "A Survey on Extractive Text Summarization, International Journal of Advanced Research in Computer Science and Software Engineering", Volume 6, Issue 4, ISSN: 2277 128X, April 2016.

[7]. Khan, Atif, and Naomie Salim. "A review on abstractive summarization methods." Journal of Theoretical and Applied Information Technology 59, 64-72, no. 1 , 2014.

[8]. Dawinder Kaur, Rajbhupinder Kaur, Automatic Summarization of Text Documents Written in Hindi Language, International Journal of Computer Science and Mobile Computing, ISSN 2320–088X IJCSMC, Vol. 3, Issue. 10, pg.320 – 323, October 2014.

[9]. Deepali P kadam, Mrs. Nita Patil Mrs. Archana Gulathi "A Comparative Study Of Hindi Text Summarization Techniques", Genetic Algorithm And Neural Network, International Journal of Innovations & Advancement in Computer Science IJIACS ISSN 2347 – 8616 Volume 4, Special Issue March 2015.

[10]. Sarkar, Kamal. "An approach to summarizing Bengali news documents." In proceedings of the International Conference on Advances in Computing, Communications and Informatics, pp. 857-862. ACM, 2012.

[11]. Natural Language Processing and Computational Linguistics: A Practical guide to text analysis with Python, Gensim, Spacy and Keras by Bhargav Srinivasa-Desikan.