Interdisciplinary Journal of Linguistics
Volume [15] 2022, pp. 195-201

HINDI TEXT PRE-PROCESSING: PROCEDURAL STEPS OR TECHNIQUES FOR NLP TOOL DEVELOPMENT

Kapil V Gawande*
Chandrakant S. Ragit**

Abstract

Text pre-processing plays an important role in the development of NLP applications. Text pre-processing is the process of transforming unstructured and noisy text data into a clean and consistent format. Linguists create tools and software that aim to facilitate the lexical goals and technical understanding of a language. The available text data of any language cannot be completely improved, which is why many applications of natural language processing do not work optimally, becoming a bottleneck and not achieving the goal. Many techniques are used in text pre-processing, and this technique brings improvements to the cleanliness, accuracy, and correctness of the text.

Keywords: Segmentation, Regular Expression, Stemming, Lemmatization, NLP

Introduction

Text pre-processing plays a vital role in the development of natural language processing applications. Text pre-processing is done in many different ways, including stemming, regular expression (regex) detection and removal, lemmatization, and stop word remover techniques. These techniques are used to clean and organize the text so that further application of natural language processing does not involve any problems in programming, processing data, applying rules, or logic.

Review

1. Need of Text Pre-processing
2. Text Pre-Processing Steps
3. Experiment
4. Applications
5. Finding and Discussion

Need of Text Pre- processing

Pre-processing text data is one of the most difficult tasks in natural language processing because there are no specific

* Mahatma Gandhi Antarrashtriya Hindi ViswaVidyalaya, Wardha (Maharashtra)

** Mahatma Gandhi Antarrashtriya Hindi ViswaVidyalaya, Wardha (Maharashtra)

statistical guidelines available (AyishaTabassum, 2020). The available text data of any language cannot be completely improved; that is why many applications of natural language processing do not work better; it becomes a hindrance and cannot achieve the goal. For this reason, text pre-processing is generally used in NLP applications. Text pre-processing is the basic need for developing NLP applications. By using pre-processing, the accuracy of the text increases and one can get good results. Many techniques are used in this for the software depending on its goals. Each technique plays its own important role, such as Stop words are the most common words found in any natural language that carry very little or no significant semantic context in a sentence (Jaideep sinha K. Raulji, 2016). Stemming is the process of removing affixes (prefixes and suffixes) from features, i.e., the process derived from reducing inflected (or sometimes derived) words to their stems (Kadhim, 2018). Stop-word elimination and stemming are commonly used methods in indexing. Stop words are high frequency words that have little semantic weight and are thus unlikely to help the retrieval process (Siddiqui). Lemmatization has traditionally been a standard pre-processing technique for linear text (Jose Camacho- Collados, 2018). But text pre-processing reads "Impact on the text" (Muhittin IŞIK, 2020).

Text Pre-processing Steps

Linguists create tools and software for the purpose of text goals and to understand the language in an easy way by technicalizing the language. That's why NLP text pre-processing is done according to the target text. Basically used text pre-processing techniques are:

1. Sentence Segmentation
2. Tokenization
3. Regular Expression (regex) Detection and Removal
4. Stop Word Removal
5. Stemming
6. Lemmatization

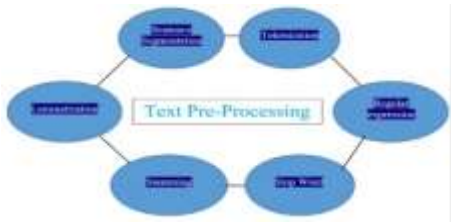

Fig. 1. Text pre-processing

Sentence Segmentation

Sentence segmentation is also called sentence boundary detection or sentence tokenization. (AyishaTabassum, 2020) It is the action of breaking the given text string. It is used to separate punctuation marks in Hindi or other languages like a full stop or comma, semicolon.

Tokenization

Tokenization refers to the splitting of sentences into words, characters, and punctuation, all of which are called "tokens." Splitting criteria are primarily based on the occurrence of a space or punctuation mark. This step helps in filtering out unwanted words in further processing steps.

Example.

"प्रकृतिकभाषासंसाधनमेंकईक्षेत्रशामिलहैं।"

Converting this sentence into a token:

"प्रकृतिक", "भाषा", "संसाधन", "में", "कई", "क्षेत्र", "शामिल", "हैं", "।"

Regular Expression (regex) Detection and Removal

Regular expressions are most commonly used in text pre-processing. Regular expressions are most commonly used in text pre-processing. Since any word is a symbol for the machine, regular expressions are not understood by the machine. This regular expression becomes a hindrance to processing the text, so it is necessary to remove it. Regular expressions can include punctuation marks, symbols, and special symbols.

Example:

Numbers, Extra space, symbols like ([?,/,\,@,,,;,#,!, %,$,*,&,।,{,},(,)])

Stop Word Removal

Stop words are unnecessary words that are used in sentences (Jashanjot Kaur, 2018). These words are used only to enhance the glory of the sentence. Many stop words are used in the Hindi language. The deletion of which has no effect on the sentence. It is used in many NLP applications. In Hindi, common words include 'मे', 'का', 'के', 'को', 'और' and so on.

Stemming

Stemming is the process of reducing a word by removing suffixes and prefixes to obtain the root word. Although the

semantic meaning does not change after removing the suffix or prefix of the root word, the suffix is brought to the base root word by truncating it, while meaning of all the different forms remains the same. However, this does not always provide good results as the word loses its meaning (Ayisha Tabassum, 2020).

Example 1:गाड़ीवाला = गाड़ी +वाला here,वाला =**suffix** and गाड़ी=**root word**

Example 2: घोड़ेवाला=घोड़े +वाला here,वाला =**suffix** and घोड़े =**root word**

In Example 1, if the suffix "वाला" is removed, the root word is "गाड़ी", but similarly in Example 2, if the suffix " वाला " is removed, the root word is "घोड़े" but the root word is "घोडा".

Lemmatization

Lemmatization is the act of removing or replacing a prefix or a suffix. In this action, an attempt is made to find the root word of a word. The root word, also known as lemma, is a meaningful word in and of itself (Muhittin IIK, 2020). Many words are formed by taking root words and adding prefixes and suffixes to them. The root words to which suffixes (prefixes and suffixes) are added are called root words because they form the basis of the new word.

Example: 'अंतरराष्ट्रीयता'it uses **root word**'राष्ट्र', **prefix**'अंतर'and **Suffix**'ईयता'

Lemmatization is commonly used in POS taggers. Due to this, the accuracy of the POS tagger gives good results.

More techniques or steps can be added to text pre-processing, such as POS tagging, chunking, parsing, adding or replacing synonyms and antonyms, etc.

Experiment

Clustering methods have been commonly used for text pre-processing. Data collection for stop word removal is included, as are data on root words or affixes for the stemming and lemmatization processes. If there is data on the root word in it, then by matching the root word with the word, the root word is obtained by removing the affixes. Similarly, if there is a database of affixes, the affixes are removed by matching the suffix of the word.

Front End

The front end for this online text pre-processing project uses JavaScript. The front end for this online text pre-processing project uses JavaScript. This project includes the processes of sentence segmentation, tokenization, regular expression detection and removal, and stop word removal.

Fig. 2: Application Screenshot

Back End

In the back end, a database of stop words used in Hindi was used. Some stop words are also included in this section to help identify the word and place it in front of or behind it using a recursive technique.

Applications

Almost all language related NLP applications are used for text pre-processing techniques. such as information retrieval (Ruby Rania, 2018), text summarization, domain recognition (Roman Sergienko, 2016), text classification, document similarity (Urvashi Garg, 2014), keyword matching, keyword recognition, POS tagger, name entity identifier, language translation, and many more (Jaideepsinh K. Raulji, 2016).

Findings & Discussion

It was discovered that several text preparation stages or techniques alter the text, which results in a lexical and semantic loss in Hindi language text. The meaning of the text may appear incorrect or altered as a result of the addition or deletion of any Hindi word.

Antonyms and Synonyms, as an illustration, think about adding and deleting words.

- ✓ As with "karana (करना)" which in English means "do" the word "kar(कर)" is a Hindi grammatical category verb.
- ✓ In English, the Hindi grammatical category noun"kar(कर)" it means as "tax."
- ✓ In Hindi, the word "kar(कर)" is a stThe Hindi stop words category includes a large number of words that have grammatical category verbs.

Example. ["kiya(किया)", "karane(करने)", "rahati(रहती)" etc.]

Conclusion

The use of text pre-processing plays an important role in the application of NLP. It compresses the text by giving it a well-defined structure. This is the first step in improving language text in NLP applications.

Works Cited

Ayisha, Tabassum and Rajendra R. Patil. "A Survey on Text Pre-Processing & Feature Extraction Techniques in Natural Language Processing." *International Research Journal of Engineering and Technology*, 2020, pp. 4864-4867.

Camacho-Collados, Jose and Mohammad Taher Pilehvar. "On the Role of Text Preprocessing in Neural Network Architectures:An Evaluation Study on Text Categorization and Sentiment Analysis". 2018.

Garg, Urvashi and Vishal Goyal. "Effect of Stop Word Removal on Document Similarity for Hindi Text." *International Journal of Engineering Sciences*, 2014, pp. 161-163.

Kadhim, Ammar Ismael. "An Evaluation of Preprocessing Techniques for Text Classification". *International Journal of Computer Science and Information Security*, 2018, pp. 22-32.

Kaur, Jashanjot and Preetpal Kaur Buttar. "A Systematic Review on Stopword Removal Algorithms". *International Journal on Future Revolution in Computer Science & Communication Engineering*, 2018, pp. 207-210.

Muhittin IŞIK, Hasan DAĞ. " The Impact of Text Preprocessing on the Prediction of Review Ratings". *Turkish Journal of Electrical Engineering & Computer Sciences*, 2020, pp. 1405-1421.

Rania, Ruby and D.K.Lobiyala. "Automatic Construction of Generic Stop Words List for Hindi Text." *International Conference on Computational Intelligence and Data Science (ICCIDS)*, Elsevier. 2018, pp. 362-370.

Raulji, Jaideepsinh K and Jatinderkumar R. Saini. "Stop-Word Removal Algorithm and its Implementation ." *International Journal of Computer Applications*, 2016, pp. 15-17.

Sergienko, Roman et al. "A Comparative Study of Text Preprocessing Approaches for Topic Detection of User Utterances." *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. European Language Resources Association (ELRA), 2016, pp. 1826−1831.

Siddiqui, Amaresh Kumar Pandey and Tanvver J. "Evaluating Effect of Stemming and Stop-word Removal on Hindi Text Retrieval" (n.d.), 2009, pp. 316-317.