SCHOOL OF DATA SCIENCE AND FORECASTING

Text Preprocessing

Subject :- Natural Language Processing

In partial fulfillment of the award of the degree

of

MASTER OF TECHNOLOGY

In

DATA SCIENCE

Submitted by

DIKSHANT NARWARE (DS7A-2206)

Submitted to

Prof. Punit Gupta

DEVI AHILYA VISHWAVIDALAYA

Indore (M.P.)

JAN, 2023

STATEMENT OF ORIGINALITY

Following the requirements for the Degree of Master of Technology in DATA SCIENCE in the SCHOOL OF DATA SCIENCE AND FORECASTING, I present this report entitled – Text Preprocessing . This report is completed under the Supervision of:

Dr. Punit Gupta
Designation: Professor

I declare that the work presented in the report is my work except as acknowledged in the text and footnotes, and that to my knowledge this material has not been submitted either in whole or in part, for any other degree at this University or any other such Institution.

Name of the Student:
Dikshant Narware
Date: 06/05/2023

SCHOOL OF DATA SCIENCE AND FORECASTING DEVI AHILYA VISHWAVIDYALAYA
INDORE (M.P)

CERTIFICATE

This is to certify that the dissertation entitled "----Text Preprocessing in Natural Language Processing-----" submitted by –Dikshant Narware --- is approved for the award of Master of Technology in DATA SCIENCE.

ACKNOWLEDGEMENT

We would like to express our gratitude to the **Punit Gupta** Faculty of the **SCHOOL OF DATA SCIENCE AND FORECASTING** Department for guidance and support throughout this work. She has been a constant source of inspiration to us throughout this work. We consider ourselves extremely fortunate for having had the opportunity to learn and work under her guidance over the entire period. I also express my sincere thanks to all the teachers of the **SCHOOL OF DATA SCIENCE AND FORECASTING**, who gave me the golden opportunity to do this wonderful **CASE STUDY** on the topic **"Text Preprocessing in NLP"**, which also helped me in doing a lot of research and I came to know about so many new things I am thankful to them. Last but not least I would like to thank all my friends and family members who were involved directly or indirectly in my work.

Dikshant Narware

Introduction:

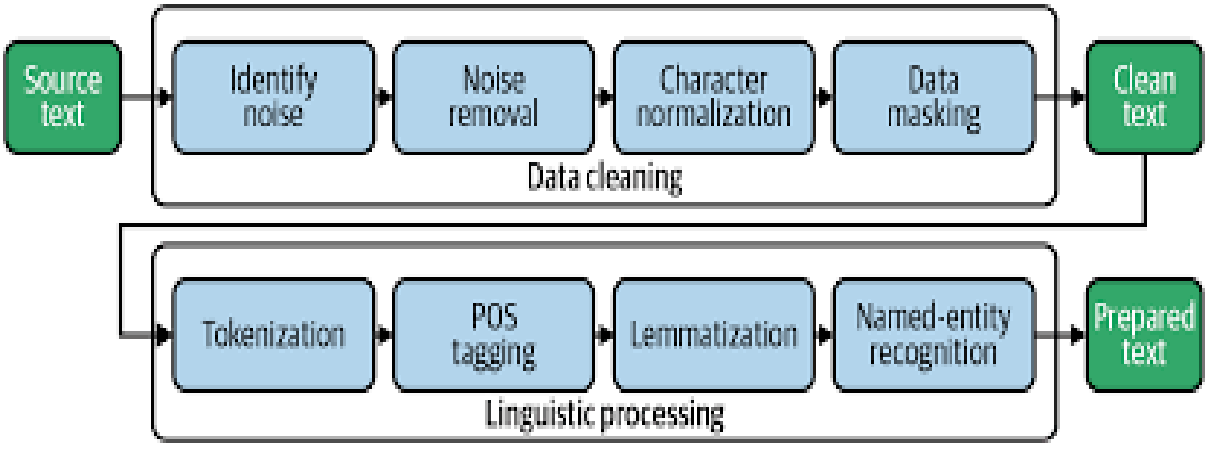
Text pre-processing is a critical step in Natural Language Processing (NLP). It involves cleaning and transforming the raw text data into a format that is suitable for analysis. Pre-processing techniques vary depending on the type of analysis being performed, but some common techniques include tokenization, stop-word removal, stemming, and lemmatization.

Tokenization:

Tokenization is the process of breaking down a sentence or a paragraph into individual words or tokens. This step is important because most NLP algorithms work on individual words rather than on the entire text. Tokenization is done by splitting the text based on spaces, punctuation marks, or special characters. For example, the sentence "The cat is on the mat." can be tokenized into ["The", "cat", "is", "on", "the", "mat", "."].

Stop-word Removal:

Stop words are commonly used words like "and", "the", "in", "a", etc. that do not add much value to the text analysis. These words can be removed to reduce the dimensionality of the text data and improve the efficiency of the NLP algorithm. The list of stop words varies depending on the context and the language being analyzed.



Stemming:

Stemming is the process of reducing a word to its base or root form. This is done by removing the suffixes and prefixes from the word. Stemming can help in reducing the dimensionality of the text data and improve the efficiency of the NLP algorithm. For example, the words "running", "runs", and "run" can be stemmed to "run".

Lemmatization:

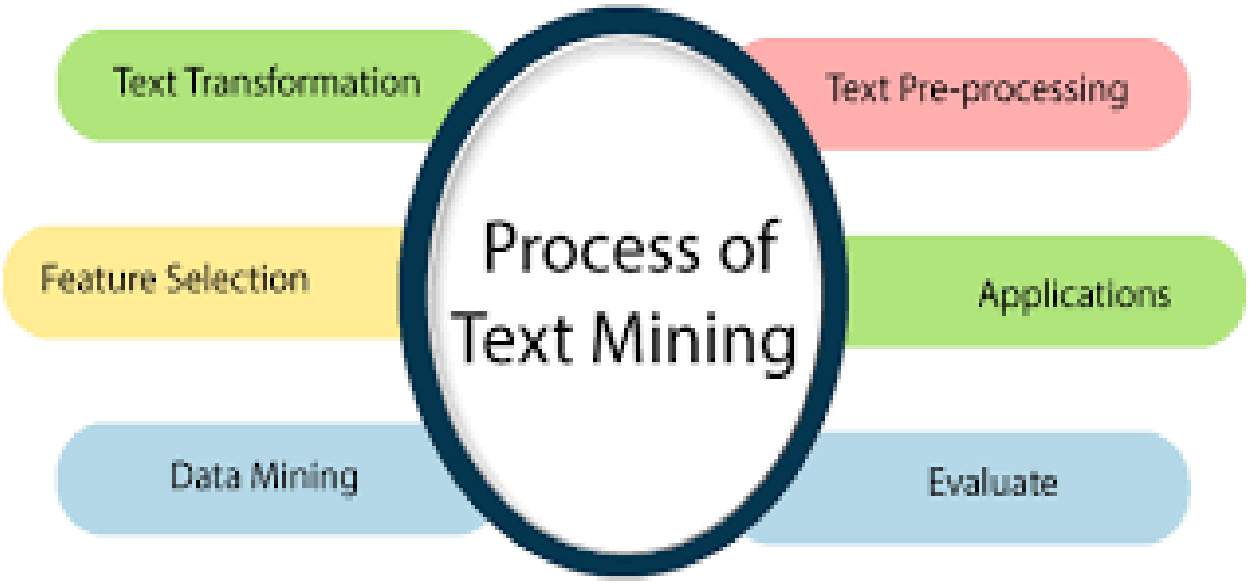
Lemmatization is similar to stemming, but instead of reducing the word to its root form, it transforms the word to its dictionary form or lemma. This is done by considering the part of speech of the word and applying the appropriate transformation. For example, the words "am", "are", and "is" can be lemmatized to "be".

Normalization:

Normalization is the process of transforming the text data to a standard format. This can include converting all the text to lowercase, removing numbers, removing punctuation marks, and removing special characters. Normalization helps in reducing the noise in the text data and making it easier to process.

Text Pre-processing :-

Text pre-processing is the first and crucial step in NLP. It involves cleaning and transforming the raw text data into a format that can be easily analyzed. This section provides a detailed explanation of text pre-processing techniques, including:



Lowercasing:

Lowercasing is a text pre-processing technique that involves converting all the text to lowercase. This step is important because it reduces the dimensionality of the text data by treating words that are capitalized and words that are not capitalized as the same. Lowercasing helps in improving the efficiency of the NLP algorithm and making it easier to analyze the text data.

Removing Punctuations and Special Characters:

Removing punctuations and special characters is another important text pre-processing technique. Punctuations and special characters do not add much value to the analysis and can create noise in the text data. Removing them can help in reducing the dimensionality of the text data and making it easier to analyze.

Removing Stop Words:

Stop words are commonly used words like "and", "the", "in", "a", etc. that do not add much value to the text analysis. These words can be removed to reduce the dimensionality of the text data and improve the efficiency of the NLP algorithm.

Stemming and Lemmatization:

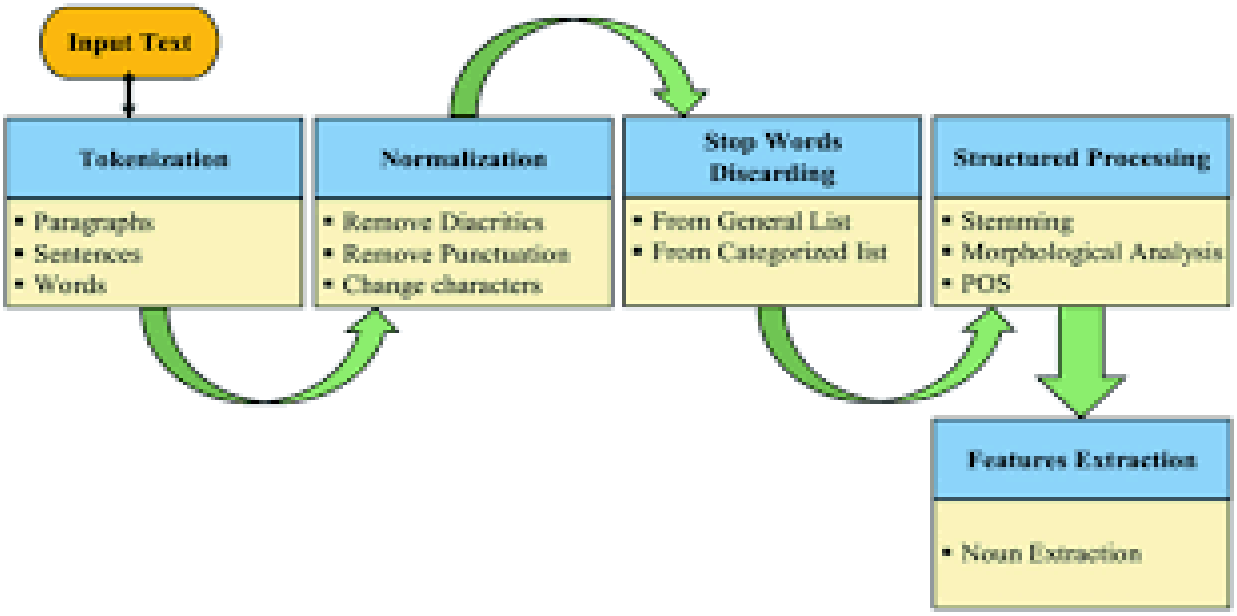
Stemming and lemmatization are text pre-processing techniques that involve reducing words to their base or root form. Stemming involves removing the suffixes and prefixes from the words, while lemmatization transforms the word to its dictionary form or lemma. These techniques help in reducing the dimensionality of the text data, improving the efficiency of the NLP algorithm, and making it easier to analyze.

Removing HTML Tags:

Removing HTML tags is another important text pre-processing technique. HTML tags are used to format the text on web pages but can create noise in the text data. Removing them can help in reducing the dimensionality of the text data and making it easier to analyze.

Text representation:

is the process of transforming raw text data into a numerical format that can be used for analysis by machine learning algorithms. There are several text representation techniques used in natural language processing, including:



Bag of Words (BoW):

BoW is a simple text representation technique that involves creating a matrix of word occurrences in a given text corpus. The rows of the matrix represent the documents in the corpus, and the columns represent the words in the corpus. Each cell in the matrix represents the count of the number of times a particular word appears in a particular document. BoW ignores the order of the words and only considers the frequency of each word in the document.

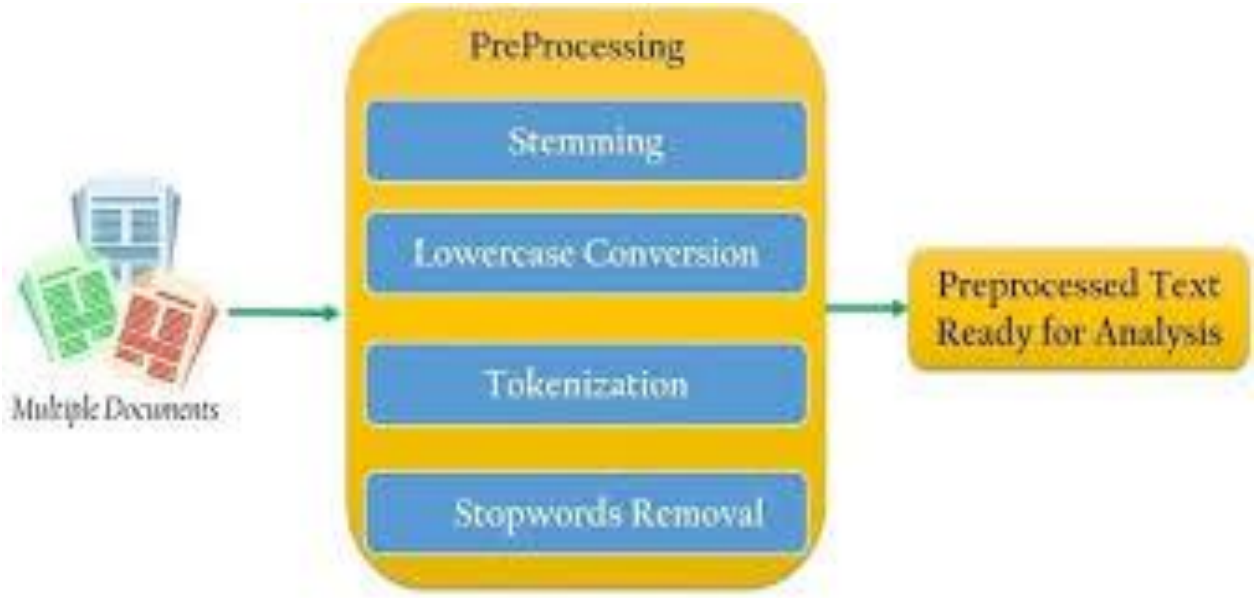
Term Frequency-Inverse Document Frequency (TF-IDF):

TF-IDF is a text representation technique that gives more weight to the words that are important to the document and less weight to the words that are common across the corpus. The TF-IDF score of a word is calculated as the product of its term frequency and inverse document frequency. The term frequency is the number of

times the word appears in the document, and the inverse document frequency is the logarithm of the total number of documents in the corpus divided by the number of documents that contain the word.

Word Embeddings:

Word embeddings are a family of text representation techniques that involve representing words as dense vectors in a high-dimensional space. Each dimension in the vector represents a different feature of the word, and similar words are represented by vectors that are close to each other in the space. Word embeddings are typically learned using neural network models such as Word2Vec and GloVe.



Character-level Representations:

Character-level representations involve representing words as sequences of characters rather than individual words. This allows the representation to capture the spelling and morphological features of the words, which can be useful for tasks such as named entity recognition and sentiment analysis.

Subword-level Representations:

Subword-level representations involve representing words as sequences of smaller units such as characters, character n-grams, or word pieces. This can help to capture

the morphological and semantic information of the words and can improve the performance of the models on rare and out-of-vocabulary words.

One-Hot Encoding:

One-hot encoding is a simple binary text representation technique that involves representing each word in a vocabulary as a binary vector of 1s and 0s. In this technique, each word in a vocabulary is assigned a unique index, and the corresponding binary vector has a value of 1 at the index corresponding to the word and 0s elsewhere. One-hot encoding is a simple and effective text representation technique for tasks such as text classification and sentiment analysis.

Word2Vec:-

Word2Vec is a widely used word embedding technique that represents words in a high-dimensional vector space. This section provides a step-by-step guide to training and using Word2Vec models.

Text Classification:

Text classification is a popular NLP task that involves classifying text data into different categories based on its content. This section provides an in-depth explanation of text classification, including:

Types of Text Classification:

There are several types of text classification, including:

- **Binary Classification**: In binary classification, text data is classified into two categories, such as spam or not spam.

- **Multi-class Classification**: In multi-class classification, text data is classified into more than two categories, such as sentiment analysis where text data is classified into positive, negative, or neutral.

- **Multi-label Classification**: In multi-label classification, text data can belong to multiple categories. For example, a news article can be classified into politics, sports, and entertainment.

Text Feature Extraction:

- The first step in text classification is to extract relevant features from the text data. Common text feature extraction techniques include:

- Bag of Words: Bag of words is a simple technique that involves representing text data as a bag of words, ignoring the order of the words and only considering the frequency of each word in the text.

- Term Frequency-Inverse Document Frequency (TF-IDF): TF-IDF is a more sophisticated technique that takes into account the importance of each word in the text and the corpus as a whole.

- **Word Embeddings:** Word embeddings represent words as dense vectors in a high-dimensional space, capturing the semantic meaning of the words.

Model Selection and Training:

After feature extraction, the next step is to select a suitable machine learning algorithm and train the model on the extracted features. Popular machine learning algorithms used for text classification include:

Naive Bayes: Naive Bayes is a probabilistic algorithm that works well for high-dimensional sparse data such as text data.

Support Vector Machines (SVM): SVM is a linear classifier that works well for both linearly and non-linearly separable data.

Convolutional Neural Networks (CNN): CNN is a deep learning algorithm that has been shown to be effective for text classification tasks.

Evaluation Metrics:

- To evaluate the performance of the text classification model, various evaluation metrics are used, including:
- Accuracy: Accuracy measures the proportion of correctly classified data points.
- Precision: Precision measures the proportion of true positives out of the total number of predicted positives.
- Recall: Recall measures the proportion of true positives out of the total number of actual positives
- F1-score: F1-score is the harmonic mean of precision and recall and provides a balanced measure of the performance of the classifier.

POS Tagging:

POS (part-of-speech) tagging is the process of marking each word in a text with its corresponding part of speech. This section provides an introduction to POS tagging, including:

also known as part-of-speech tagging, is a natural language processing technique that involves assigning grammatical tags to words in a text corpus based on their part of speech. These tags can help in identifying the syntactic structure of the text and can be used in many NLP tasks such as text classification, sentiment analysis, and information retrieval.

POS tag sets:

There are various tag sets used for POS tagging, but some of the most commonly used tag sets are:

Brown Corpus tag set: This tag set was developed for the Brown Corpus, which is a large collection of American English texts. It contains 87 tags, including nouns, verbs, adjectives, adverbs, prepositions, pronouns, conjunctions, and interjections.

Penn Treebank tag set: This tag set was developed for the Penn Treebank corpus, which is a large corpus of written and spoken English. It contains 45 tags, including nouns, verbs, adjectives, adverbs, prepositions, pronouns, conjunctions, and interjections.

Universal tag set: This tag set is a simplified version of the Penn Treebank tag set, containing only 17 tags. It is designed to be language-independent and can be used for POS tagging in many languages.

POS tagging algorithms:

There are several algorithms used for POS tagging, including:

Rule-based tagging: Rule-based tagging involves manually creating a set of rules based on linguistic rules and heuristics. These rules are then used to assign POS tags to words in the text corpus.

Hidden Markov Models (HMM): HMM is a statistical approach that models the probability of a sequence of words and their corresponding POS tags. This approach involves learning the probability distribution of words and their tags from a training corpus and then using this information to tag new text data.

Conditional Random Fields (CRF): CRF is a machine learning algorithm that models the conditional probability of a sequence of words given their corresponding POS tags. It uses a training corpus to learn the feature weights that are used to make predictions about the POS tags of new text data.

Conclusion:

Text pre-processing is a critical step in NLP. Lowercasing, removing punctuations and special characters, removing stop words, stemming and lemmatization, and removing HTML tags are some common text pre-processing techniques. These techniques help in reducing the dimensionality of the text data, improving the efficiency of the NLP algorithm, and making it easier to analyze.

This repository provides a comprehensive guide to NLP text processing and analysis, covering the essential topics in the field. Whether you're a beginner **or an** experienced NLP practitioner, this repository will help you improve your skills and knowledge.