|  |  |
| --- | --- |
| **TITLE** | Text Analytics |
| **PROBLEM STATEMENT/ DEFINITION** | 1. Extract Sample document and apply following document preprocessing methods: Tokenization, POS Tagging, stop words removal, Stemming and Lemmatization.  2. Create representation of document by calculating Term Frequency and Inverse Document Frequency. |
| **OBJECTIVE** | 1. To apply preprocessing steps like tokenization, POS tagging, stop word removal, stemming and Lemmatization on a sample document and bring the text into a form that is predictable and analyzable for a specific task  2. Calculate Term Frequency and Inverse Document Frequency which is used by [search engines](https://en.wikipedia.org/wiki/Search_engine) as a central tool in scoring and ranking a document's [relevance](https://en.wikipedia.org/wiki/Relevance_(information_retrieval)) given a user [query](https://en.wikipedia.org/wiki/Information_retrieval). |
| **S/W PACKAGES AND HARDWARE APPARATUS USED** | **Operating System recommended** :- 64-bit Open source Linux or its derivative  **Programming tools recommended**: - JAVA/Python/R/Scala |
| **REFERENCES** | 1)Chirag Shah, “A Hands-On Introduction To Data Science”, Cambridge University Press,(2020), ISBN : ISBN 978-1-108-47244-9. 2.  2)Wes McKinney, “Python for Data Analysis”, O' Reilly media, ISBN : 978-1-449-31979-3. |
| **STEPS** | **Refer to student activity flow chart if found necessary by subject teacher and relevant to the subjectmanual.**  **Describe steps only.** |
| **INSTRUCTIONS FOR WRITING JOURNAL** | 1. title 2. Problem statement 3. Learning objective 4. Learning outcome 5. Theory (includes methods, libraries and functions, 6. Analysis (as per assignment), 7. conclusion. |

HeadofDepartment SubjectCo-ordinator

**(Dr. M.S.Takalikar) (Dr. S.S.Sonawane)**

P:F:-LTL-UG / 03 /R1

**Theory:**

**Text Analytics:**

Text analytics is the process of extracting meaning out of text. For example, this can be analyzing text written by customers in a customer survey, with the focus on finding common themes and trends. The idea is to be able to examine the customer feedback to inform the business on taking strategic action, in order to improve customer experience.

**Tokenization**

One of the very basic things to do is dividing a body of text into words or sentences, splitting a phrase, sentence, paragraph, or an entire text document into smaller units, such as individual words or terms.The tokens could be words, numbers or punctuation marks. In tokenization, smaller units are created by locating word boundariesThis is called tokenization.

### Word Tokenization

Word Tokenization is the most commonly used tokenization algorithm. It splits a piece of text into individual words based on a certain delimiter. Depending upon delimiters, different word-level tokens are formed.

**Drawbacks of Word Tokenization**

One of the major issues with word tokens is dealing with**Out Of Vocabulary (OOV) words.** OOV words refer to the new words which are encountered at testing. These new words do not exist in the vocabulary. Hence, these methods fail in handling OOV words.

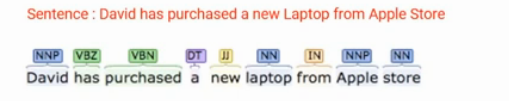
### Character Tokenization

Character Tokenization splits apiece of text into a set of characters. It overcomes the drawbacks we saw above about Word Tokenization.Character Tokenizers handles OOV words coherently by preserving the information of the word. It breaks down the OOV word into characters and represents the word in terms of these character.

**POS Tagging**

* Classifying words in their part of speech and providing them labels according to their part of speech is called part of speech tagging or POS tagging OR POST.  Hence the set of labels/tags is called a tagset Parts of speech tags are the properties of the words, which define their main context, functions, and usage in a sentence. Some of the commonly used parts of speech tags are
* **Nouns**: Which defines any object or entity
* **Verbs**: That defines some action.
* **Adjectives and Adverbs**: This acts as a modifier, quantifier, or intensifier in any sentence.
* In a sentence, every word will be associated with a proper part of the speech tag. For example**,** consider the sentence below

Example: David has purchased a new laptop from Apple store



In this sentence, every word is associated with a part of the speech tag which defines their functions. Here, David has an NNP tag which means it is a proper noun. Further, Has and purchased belong to the verb indicating that they are the actions. The Laptop and Apple store are the nouns. New is the adjective whose role is to modify the context of the laptop.

Parts of speech tags are defined by the relationship of words with the other words in the sentence. We can apply machine learning models and rule-based models to obtain the parts of speech tags of a word.

**Stop Word Removal**

**Stop words**:The words which are generally filtered out before processing a natural language are called **stop words**. These are actually the most common words in any language (like articles, prepositions, pronouns, conjunctions, etc) and does not add much information to the text. Examples of a few stop words in English are “the”, “a”, “an”, “so”, “what”

**Stemming :**

Stemming is an important part of the pipelining process in Natural language processing. The input to the stemmer is tokenized words.

**Errors in Stemming:**   
There are mainly two errors in stemming – 

* over-stemming
* under-stemming

Over-stemming occurs when two words are stemmed from the same root that are of different stems. Over-stemming can also be regarded as false-positives.

Under-stemming occurs when two words are stemmed from the same root that are not of different stems. Under-stemming can be interpreted as false-negatives.

**Applications of stemming :** 

1. Stemming is used in information retrieval systems like search engines.
2. It is used to determine domain vocabularies in domain analysis.

**Lemmatization :**

Lemmatization is the process of converting a word to its base form. The difference between stemming and lemmatization is, lemmatization considers the context and converts the word to its meaningful base form, whereas stemming just removes the last few characters, often leading to incorrect meanings and spelling errors.

Examples of implementing this comes in the following sections. we will see how to implement lemmatization using the following python packages.

1. Wordnet Lemmatizer
2. Spacy Lemmatizer
3. TextBlob
4. CLiPS Pattern
5. Stanford CoreNLP
6. Gensim Lemmatizer
7. TreeTagger

**TF-IDF**

tf-idf, short for term frequency–inverse document frequency, is a numeric measure that is use to score the importance of a word in a document based on how often did it appear in that document and a given collection of documents. The intuition for this measure is : If a word appears frequently in a document, then it should be important and we should give that word a high score. But if a word appears in too many other documents, it’s probably not a unique identifier, therefore we should assign a lower score to that word

## TF Term Frequency

The first part of the formula tf(t,d)tf(t,d) is simply to calculate the number of times each word appeared in each document. Of course, as with common text mining methods: stop words like “a”, “the”, punctuation marks will be removed beforehand and words will all be converted to lower cases.

### Inverse document frequency

The **inverse document frequency** is a measure of how much information the word provides, i.e., if it's common or rare across all documents. It is the [logarithmically scaled](https://en.wikipedia.org/wiki/Logarithmic_scale) inverse fraction of the documents that contain the word (obtained by dividing the total number of documents by the number of documents containing the term, and then taking the logarithm of that quotient)