International Journal of Engineering and Technical Research (IJETR)
ISSN: 2321-0869 (O) 2454-4698 (P) Volume-7, Issue-8, August 2017

Synonym or similar word detection in assignment papers

Gayatri Behera

***Abstract*— Natural Language Processing (NLP) is one of the domains that hold the potential to transform and improve the human-computer connect. It helps provide a medium to analyze and decode human generated text and speech (in a text-based format) to glean useful insight from it. NLTK is a Python-based module that comes in handy while trying to solve problems belonging to the domain of NLP. It can find immense use in trying to detect a recurring, common pattern in a text stream.**

***Index Terms*— NLP, NLTK, Corpus, Word Tokens**

I. INTRODUCTION

Natural Language Processing is possible by utilizing NLTK (Natural Language Tool Kit) which comprises of a set of libraries available under Python that helps to not just tokenize words but also sentences separately. It also allows to group elements of a sentence together on the basis of appearance of a particular type of word, and also on the basis of arrangement of particular sequence of words. It allows 'chunking' or grouping together of data, which belongs to one common type i.e. grouping of nouns, verbs, adjectives etc. together. Additionally, it also allows to determine the word stem of the words, that in turn helps in better judgment and to gain insight about the actual context of the word.

NLTK comprises primarily of a vast body of work that encompasses a corpora as well as commonly used lexicon pertaining to different fields i.e. separate lexicon usage for medical, legal, financial, actuarial sciences etc. It is these specialties that make it an essential tool and a helpful prerequisite for anyone trying to get a better understanding and gain a foothold in the domain of NLP.

II. WORKING

The initial steps would involve procuring sample assignment responses or thesis material from different sets of students. These can be suspected to be having unoriginal content, with the "key" words or phrases being replaced by words similar in meaning with the intent of making it appear like an original piece of work; with the words being merely interchanged with their synonyms. It can be achieved by making use of Natural Language Tool Kit (NLTK) - a package available with Python that helps in word processing, cleaning, segregation and pattern-matching to derive insight from a large body of text.

The steps would involve:-

1. Fetching of the corpus or body of text pertinent to this problem.
2. Separating the text files into training and test set. These have to be determined randomly.
3. Implement initial filtering on this body of text to separate text into chunks.
4. Separation of the sample text as per the parts of their speech into specific components i.e. as nouns, verbs, adjectives etc.
5. Identify the specific parts of speech that may contain the most information at any given set time i.e. be it noun or adjective etc.
6. Perform comparison in these respects between the training and test set data to determine a correlation.

III. SYSTEM FLOW:

NLTK provides a host of features as well as functions that make the task easier. Given below are some of the key components and commonly used terminologies.

Corpora - This is a collection of a large body of text that can be used while performing insight gathering.

Lexicon – Jargon or peculiar text or verbiage pattern which is specific to a particular community or group of people. This is commonly observed for any particular trade such as in the fields of medicine, law, finance, physical or environmental sciences etc.

Token – An entity or part of a sentence.

Stemming – A manner of normalizing various words that exist in different tense formats but convey the same meaning.

Chunking – Grouping of text together that belong to a particular word group i.e. nouns, adjectives, pronouns etc. This would help in further filtering.

Lemmatizing – Identification of word root of a particular word irrespective of its current state (i.e. whether it is in past participle form, adverb form of the word etc.) It is considered to be more effective than stemming.

This problem is primarily focused on highlighting instances of plagiarism that tend to be rampant in academic circles at the graduate or undergraduate level. It involves segregating the text by zoning down on those parts that carry relevance. Separating these and identifying the repeated occurrences of these words that might come across in subsequently submitted assignment material.

First the sentences are broken down into small parts or tokens. This is done with the help of **sent_tokenize()** function belonging to tokenize library of NLTK. Using the **PorterStemmer** algorithm we identify the common stem of the word i.e. words such as abruptly, abruptness etc. that actually "stem" from the common root word - abrupt. Post this, the parts of speech of the words in the passage is identified. It is achieved with the help of the **PunktSentenceTokenizer** which needs to be trained separately on the training samples and test samples of text. Next, using Wordnet a lexical corpus available in NLTK, the meanings as well as synonyms and antonyms of the selected words can be found. **Wu Palmer method** is used to find the degree of similarity between two selected words.

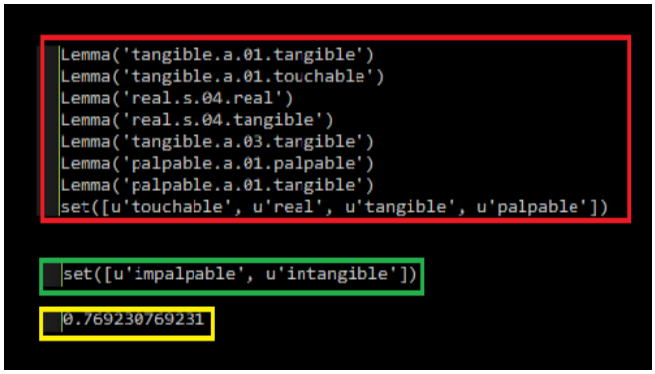

Fig (i): The section in red shows the similar words to the base word 'tangible'. The section in green shows the antonyms for the word. The section in yellow shows the percent of similarity using Wu Palmer method of two words namely – mountain and hillock.

IV. Results

Utilizing this approach helped identify such submitted material that was portrayed as original, authentic content, but was in fact lifted from another person's work. Few word usages had been replaced by their equivalent words in a bid to present the work as unique. Going by this approach, we were able to bring down instances of plagiarism and ensure authenticity of the work was maintained.

References

[1] http://www.nltk.org/
[2] Bird, Steven, Edward Loper and Ewan Klein (2009), *Natural Language Processing with Python*. O'Reilly Media Inc.
[3] https://tartarus.org/martin/PorterStemmer/
[4] https://pythonprogramming.net/
[5] https://en.wikipedia.org/wiki/Natural_Language_Toolkit