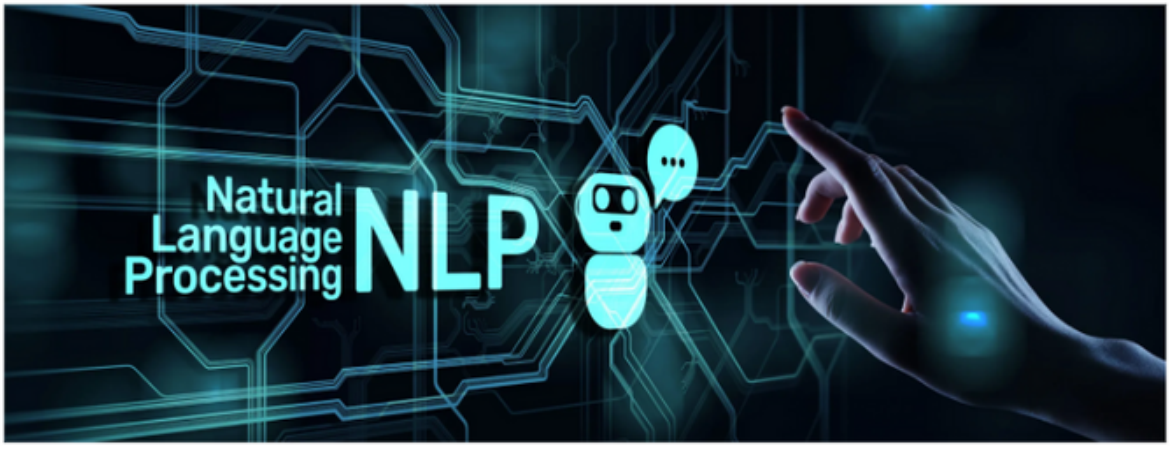
ACTIVIDAD #2

Tipo actividad: Reading comprehension activity, Multiple choice activity.

Reading comprehension activity #3: "NLP: Tokenization, Stemming, Lemmatization and Part of Speech Tagging"

NLP: Tokenization, Stemming, Lemmatization and Part of Speech Tagging

In this blog post, I'll talk about Tokenization, Stemming, Lemmatization, and Part of Speech Tagging, which are frequently used in Natural Language Processing processes. We'll have information about how to use them by reinforcing them with applications. Enjoyable readings.



Resource: https://www.asksid.ai/resources/what-is-natural-language-processing/

After downloading the library and importing it, let's define a text.

In order to do tokenization, we can access tokens by calling words from the TextBlob object. As a result, you will see that the text we have is allocated to tokens as below.

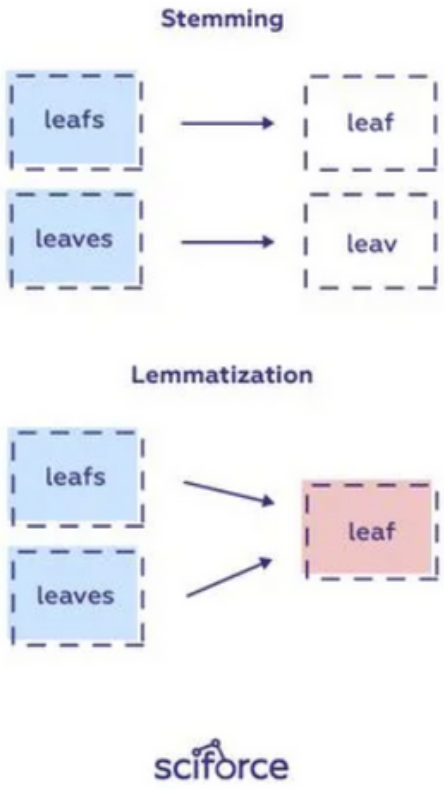
As you can see, we were able to split it into tokens quite simply. Let's do this with the NLTK (Natural Language Toolkit) library.

As you can see, we have called word_tokenize and sent_tokenize objects from the NLTK library. With sent_tokenize we'll be able to split the text into sentences. We'll split it into the same text words with word_tokenize.

The sent_tokenize function uses an instance of PunktSentenceTokenizer from the nltk.tokenize.punkt module, which has already been trained and thus very well knows to mark the end and beginning of sentence at what characters and punctuation.

word_tokenize() function is a wrapper function that calls tokenize() on an instance of the TreebankWordTokenizer class.

Thus, we can do the split into tokens in a very practical way with two different libraries.



Resource: https://tr.pinterest.com/pin/706854104005417976/

Stemming

Stemming is the process of finding the root of words. Let's examine a definition made about this.

Stemming is definitely the simpler of the two approaches. With stemming, words are reduced to their word stems. A word stem need not be the same root as a dictionary-based morphological root, it just is an equal to or smaller form of the word.

Overstemming occurs when words are over-truncated. In such cases, the meaning of the word may be distorted or have no meaning.

Understemming occurs when two words are stemmed from the same root that is not of different stems.

We'll examine the stemming example with two different algorithms.

- Porter Stemmer (Algorithm details are in this link.)
- Snowball Stemmer (Algorithm details are in this link.)

First, let's import the PorterStemmer.

Then, let's define a ps an object that will implement PorterStemmer. After defining the word to be stemming, all that remains is to run the code.

Let's do a similar process with SnowballStemmer. For this, we do import the SnowballStemmer.

Then, we define the stemmer object. Here, in addition to PorterStemmer, we can also choose in which language we will be stemming in SnowballStemmer.

Lemmatization

Lemmatization is the process of finding the form of the related word in the dictionary. It is different from Stemming. It involves longer processes to calculate than Stemming. Let's examine a definition made about this.

The aim of lemmatization, like stemming, is to reduce inflectional forms to a common base form. As opposed to stemming, lemmatization does not simply chop off inflections. Instead, it uses lexical knowledge bases to get the correct base forms of words.

NLTK provides WordNetLemmatizer class which is a thin wrapper around the wordnet corpus. This class uses the morphy()function of the WordNet CorpusReaderclass to find a lemma.

First, let's do import NLTK and WordNetLemmatizer.

Let's see how the lemmatizer works in a single word.

Then we have a text. Let's break this text down to tokens first. Then let's apply the lemmatizer one by one on these tokens.
In the first example of Lemmatizer, we used WordNet Lemmatizer from the NLTK library. Let's do similar operations with TextBlob. As a result, we will reach similar results.

When we apply the 'lemmatize' process to the word 'stripes', it deletes the 's' suffix and reaches the word 'stripe', which is the dictionary form of the word. Now let's do the same on a sentence.

Thus, we examined how the 'lemmatization' process is implemented on both sentences and a single word with two different libraries.

Part of Speech Tagging
Part of Speech Tagging (POS-Tag) is the labeling of the words in a text according to their word types (noun, adjective, adverb, verb, etc.). Let's look at how it is explained in a definition.

Resource:https://blog.aaronccwong.com/2019/building-a-bigram-hidden-markov-model-for-part-of-speech-tagging/

It is a process of converting a sentence to forms – list of words, list of tuples (where each tuple is having a form (word, tag)). The tag in case of is a part-of-speech tag, and signifies whether the word is a noun, adjective, verb, and so on.

Let's examine the most used tags with examples.

- Noun (N)- Daniel, London, table, dog, teacher, pen, city, happiness, hope
- Verb (V)- go, speak, run, eat, play, live, walk, have, like, are, is
- Adjective(ADJ)- big, happy, green, young, fun, crazy, three
- Adverb(ADV)- slowly, quietly, very, always, never, too, well, tomorrow
- Preposition (P)- at, on, in, from, with, near, between, about, under

- Conjunction (CON)- and, or, but, because, so, yet, unless, since, if
- Pronoun(PRO)- I, you, we, they, he, she, it, me, us, them, him, her, this
- Interjection (INT)- Ouch! Wow! Great! Help! Oh! Hey! Hi!

So, how does POS Tagging work?

POS tagging is a supervised learning solution that uses features like the previous word, next word, is first letter capitalized etc. NLTK has a function to get pos tags and it works after the tokenization process.

Let's understand Part of Speech Tagging with an application. Let's import the NLTK library and word_tokenize object. When applying this, we first need to split a sentence into tokens. Tagging works after splitting to tokens.

After separating the words in a sentence into tokens, we applied the POS-Tag process. For example, the word 'The' has gotten the tag 'DT'. The word 'feet' has been labeled 'NNS'. You can review this link to investigate in detail what these tags are.

Taken from: https://medium.com/mlearning-ai/nlp-tokenization-stemming-lemmatization-and-part-of-speech-tagging-9088ac068768

7) Multiple choice activity.

1. What is the primary purpose of Tokenization in Natural Language Processing (NLP)?
   - A. To find the roots of words
   - B. To reduce words to their word stems
   - C. To break down a text into the smallest units (tokens)
   - D. To label words according to their word types

2. What is the main challenge associated with Stemming in NLP?
   - A. Overstemming and understemming
   - B. Lemmatization errors
   - C. Tokenization inconsistencies
   - D. Part of Speech Tagging complexities

3. How does Lemmatization differ from Stemming in NLP?
   - A. Lemmatization involves finding the root of words
   - B. Lemmatization is a rule-based process
   - C. Lemmatization aims to reduce inflectional forms to a common base form
   - D. Lemmatization uses lexical knowledge bases

4. What is the purpose of Part of Speech Tagging (POS-Tag) in NLP?
   - A. To find the roots of words
   - B. To reduce words to their word stems
   - C. To label words according to their word types
   - D. To break down a text into the smallest units (tokens)

5. Which Python library is mentioned in the text as a valuable tool for NLP tasks, including tokenization and part-of-speech tagging?

- A. TensorFlow
- B. PyTorch
- C. NLTK (Natural Language Toolkit)
- D. Scikit-learn