Text Processing in Python

Deepesh Hada

Shikhar Bharadwaj

Contents

- Tokenization
- Stop words
- Word Normalization
  - Stemming
  - Lemmatization
- POS Tagging
- Chunking
- NER Tagging

Tokenization

- Process of segmenting a piece of text into smaller units called tokens.
- The tokens can later be used to create a dictionary of words.
- **Example:** I have a can opener, but I can’t open these cans.
  - ○ **Word Token:** An occurrence of a word (11 tokens).
  - ○ **Word Type:** *Unique* tokens (10).
- **Issues:**
  1. What’re, I’m, shouldn’t → What are, I am, should not?
  2. San Francisco → one token or two?
  3. m.p.h. → three tokens or one?

Stop words

- **Stop words** are a set of commonly used words.
- Often removed from the corpus before training models as they occur in abundance, providing little to no unique information that can be used for classification or clustering.
- In English, ***the***, ***is*** and ***and*** would easily qualify as stop words.
- These words, just like punctuations, help just in maintaining the structure and do not contribute much to the meaning of a sentence.

Word Normalization

- For grammatical reasons, a corpus usually contains different forms (inflections) of the same root. It is desirable that the search for one of these words returns the other words in the set.
- **Goal:** to reduce words to their inflectional and sometimes derivationally related forms.
- **Examples:**
  1. am, are, is → be
  2. car, cars, car's, cars’ → car
- Stemming and Lemmatization are used for Word Normalization.

Stemming

- **Stemming** is the process of reducing inflection in words to their root forms, such as mapping a group of words to the same stem even if the stem itself is not a valid word in the language.
- Hence, stemming words in a sentence may result in words that are not actual English words: a drawback.
- However, stemming is much faster than **Lemmatization** as it has a rule-based algorithm (Porter's algorithm).
- **Example:** *argue*, *argued*, *argues* and *arguing* are reduced to the stem ***argu***.

Lemmatization

- **Lemmatization**, unlike Stemming, reduces the inflected words to their respective roots (lemmas), while also ensuring that the root word belongs to the language.
- A lemma (root word) is the dictionary form of a set of words.
- Because lemmatization returns an actual word of the language, the algorithm is a bit more complex and consequently, slower than stemming.
- **Example:** *runs*, *running*, *ran* are all forms of the word ***run***, and hence, *run* is the lemma of all these words.

POS Tagging

- The **part of speech** explains how a word is used in a sentence – nouns, pronouns, adjectives, verbs, conjunctions, etc.
- POS tagging is a supervised learning solution that tags/classifies each word into one of the categories (parts of speech).
- Works on top of word tokens.

Chunking

- Chunking is a process of extracting **phrases** from unstructured text.
- Instead of using simple tokens which may not represent the actual meaning of the text, we can use phrases such as “**San Francisco**” as a single word instead of ‘**San**’ and ‘**Francisco**’ separate words.
- Works on top of POS tagging: uses POS tags as input and provides chunks as output like Noun and Verb Phrases.
- Very useful for extracting information from text for **Named Entity Extraction**.

NER Tagging

- NER (Named Entity Recognition) Tagging seeks to locate and classify **named entities** in text into a set of pre-defined categories such as person names, organizations, locations, monetary values, percentages, etc.
- Useful for analyzing unstructured text.
- **Example: [Jim]$_{\textbf{Person}}$** bought 300 shares of **[Acme Corp.]$_{\textbf{Organization}}$** in **[2006]$_{\textbf{Time}}$**.

Thank You!