Introduction to Natural Language Processing (NLP)

Yulia Newton, Ph.D.

CS156, Introduction to Artificial Intelligence

San Jose State University

Spring 2021

What is NLP?

- Natural language processing is a collection of technologies used to aid computers to understand and process natural language
- NPL is a branch of AI that deals with interactions that involve natural language
- NLP aims to derive meaning from natural language
- Human interacts with a machine using natural language and the machine “interprets”, “understands”, and responds appropriately according to the meaning of the communicated message
- Considered one of the most difficult AI problems

Examples of NLP tasks

- Audio capture
- Machine translation
- Sentiment analysis
- Grammar checkers
- Interactive voice response (IVR)
- Personal assistants (OK Google, Siri, Cortana, Alexa, etc.)

Understanding natural language can happen at multiple levels

- Morphological - root representation of the word
- Lexical - lexical meaning of the word
- Syntactic - grammar and sentence structure
- Semantic - meaning of the words in context
- Discourse - structure of different types of texts

Why are NLP problems difficult to solve?

- The rules for information passing in natural language are not easy to understand for computers
- Too many abstract concepts
  - Nested abstraction
- Tone and context nuances
  - Sarcasm
  - Connotation
  - Context-specific meaning
- Ambiguities and imprecise characteristics

Corpus

- We call the text we want to analyze “corpus”
- “In linguistics, a corpus or text corpus is a language resource consisting of a large and structured set of texts. In corpus linguistics, they are used to do statistical analysis and hypothesis testing, checking occurrences or validating linguistic rules within a specific language territory” - Wikipedia
- Plural is “corpora”
  - Collection of articles from medical journals is corpora
  - Single medical article is corpus

Major approaches to solving NLP tasks

- Syntax-based
- Semantic analysis

Syntax approach

- Based on the how the words are arranged in the sentence
- Takes grammatical rules into account
- Meaning is derived by applying grammatical rules to groups of words

Some of the syntax-based NLP techniques

- Lemmatization - grouping words into meaning groups (groups of words with the same meaning)
- Morphological segmentation - dividing text into morphemes
  - “A morpheme is the smallest meaningful unit in a language. A morpheme is not necessarily the same as a word. The main difference between a morpheme and a word is that a morpheme sometimes does not stand alone, but a word, by definition, always stands alone.” - Wikipedia
- Word segmentation - dividing text into distinct units from which meaning can be derived
- Part of speech tagging - tagging each word with the “part of speech” tag
  - Sometimes called grammatical tagging
- Parsing - analysis of grammatical structure of the sentence
- Stemming - cutting words down to the root form

Semantics approach

- Semantics means “meaning”
- Semantic analysis has to do with understanding the meaning of words, phrases, sentence parts, sentences and entire corpus

- Semantic analysis is not a solved problem

Some of the semantics based techniques

- Named entity recognition (NER) - finding parts of text that can be categorized into preset groups
  - E.g. names of people and places
- Word sense disambiguation - assigning meaning to words based on the context
- Natural language generation - involve semantic intentions databases (databases containing mappings between natural language and semantic meaning/intentions)

Organizing the corpus

- Paragraphs can be parsed by newline character
- Tokenizing
  - Word tokenizers
  - Sentence tokenizers
- Lexicons - words and their meanings in different contexts

Inflection

- "In linguistic morphology, inflection (or inflexion) is a process of word formation, in which a word is modified to express different grammatical categories such as tense, case, voice, aspect, person, number, gender, mood, animacy, and definiteness. The inflection of verbs is called conjugation, and one can refer to the inflection of nouns, adjectives, adverbs, pronouns, determiners, participles, prepositions and postpositions, numerals, articles etc., as declension." - Wikipedia

- Multiple forms of a word carry the same/similar meaning
- When processing corpus for NLP we want to minimize inflection forms as much as possible as it simplifies the problem

More on stemming an lemmatization

- Stemming - grouping the words based on their root form
  - E.g. drive, drives, driving
  - E.g. nation, national, nationality
  - E.g. playing == play
- Lemmatization - grouping the words based on their meaning
  - E.g. am, are, is == be
  - E.g. cat, cats, cat’s, cats’ == cat
  - E.g. better == good

- Original sentence: *the boy’s dogs are different sizes*
- Sentence after stemming and lemmatization: *the boy dog be differ size*

Stop words

- Words irrelevant to overall meaning of the corpus
  - E.g. “a”, “the”, “and”, etc.

Regular expressions

- Regular expressions define a search pattern
  - regex, regexp
- Within python code you usually use regex within raw string notation
  - r”\n”
- In NLP often used to apply additional text filtering

- Common regex patterns:
  - . - match any character except newline
  - \w - match word
  - \d - match digit
  - \s - match whitespace
  - \W - match not word
  - \D - match not digit
  - \S - match not whitespace
  - [abc] - match any of a, b, or c
  - [^abc] - not match a, b, or c
  - [a-g] - match a character between a & g

Bag of words

- Machine learning algorithms cannot work with text directly
  - Therefore, we have to convert text to numeric vectors
- This is a feature extraction technique used in NLP
- This technique discards the information about the text structure, word order and word relationships
- Intuition: similar documents have similar word composition (similar bags of words)

- How does it work?
  - Construct vocabulary of known words (tokens)
  - Define how you will measure whether the vocabulary words are present or not in the text
    - Counts
    - Frequencies
    - TF-IDF (in later slides)

Bag of words (cont’d)

- Beware of complexity of the bag-of-words model (how big is your vocabulary?)
  - Bigger vocabulary means bigger word vector
    - Are some of the words even relevant?

- How to minimize complexity
  - Lemmatization and stemming
  - Case-insensitive approach
  - Punctuation-insensitive approach
  - Remove stop words prior to creating bag-of-words representation
  - Data cleaning (fix misspellings, convert synonyms, etc.)

n-grams

- In NLP:
  - “Unigram” is one words
  - “Bi-gram” - two words
  - “Tri-gram” - three words
  - “n-gram” - a group of n words

- A bag-of-words model can be extended to a bag-of-grams model
  - Reduces complexity of the representation because each element of the feature vector represents a group of words instead of a single word

TF-IDF

- Term frequency-inverse document frequency (TF-IDF) is a statistical measure used to indicate importance of the word to a given corpus
  - Motivation: some of the words that frequently appear may not carry information relevant to the text meaning
  - This metric takes into account not only how often the word appears but also in how many documents it appears

Here the first term of the equation is TF and the second term is IDF

Frequency of token i in document j

Total number of documents being analyzed

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

Frequency of token i across all documents

Token i in document j

$tf_{i,j}$ = number of occurrences of i in j

df_i = number of documents containing i

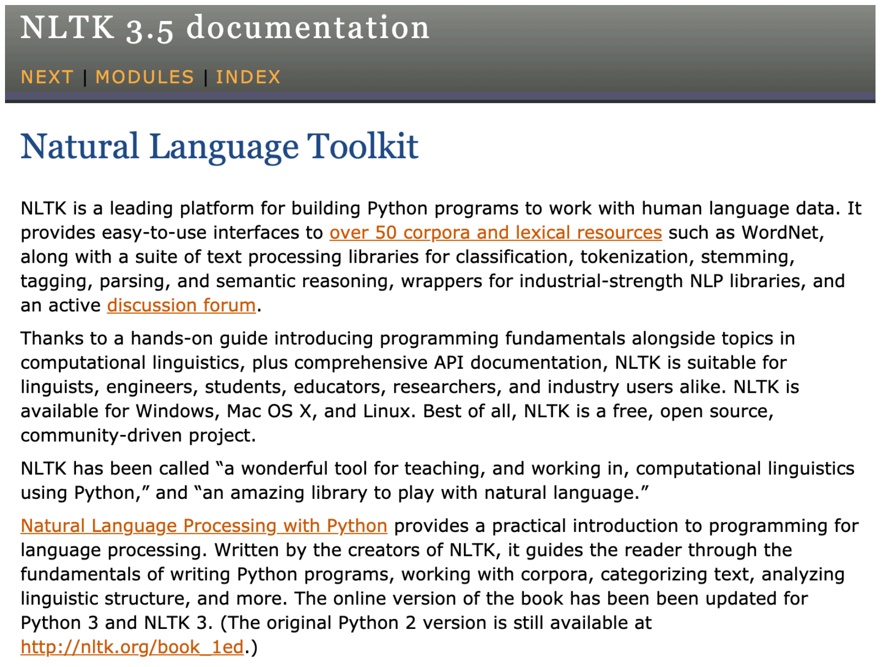
N = total number of documents

searchenginejournal.com

NLTK library

- NLP library for python
  - https://www.nltk.org/
  - https://www.nltk.org/install.html

- Using NLTK in your python code
  - *import nltk*
  - *nltk.download()*



NLTK 3.5 documentation

NEXT | MODULES | INDEX

Natural Language Toolkit

NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active discussion forum.

Thanks to a hands-on guide introducing programming fundamentals alongside topics in computational linguistics, plus comprehensive API documentation, NLTK is suitable for linguists, engineers, students, educators, researchers, and industry users alike. NLTK is available for Windows, Mac OS X, and Linux. Best of all, NLTK is a free, open source, community-driven project.

NLTK has been called "a wonderful tool for teaching, and working in, computational linguistics using Python," and "an amazing library to play with natural language."

Natural Language Processing with Python provides a practical introduction to programming for language processing. Written by the creators of NLTK, it guides the reader through the fundamentals of writing Python programs, working with corpora, categorizing text, analyzing linguistic structure, and more. The online version of the book has been been updated for Python 3 and NLTK 3. (The original Python 2 version is still available at http://nltk.org/book_1ed.)

https://www.nltk.org/

Let’s look at some code examples

- Intro_to_nltk.ipynb
- NLP.IMDB_review_sentiment_analysis_with_logistic_regression.ipynb