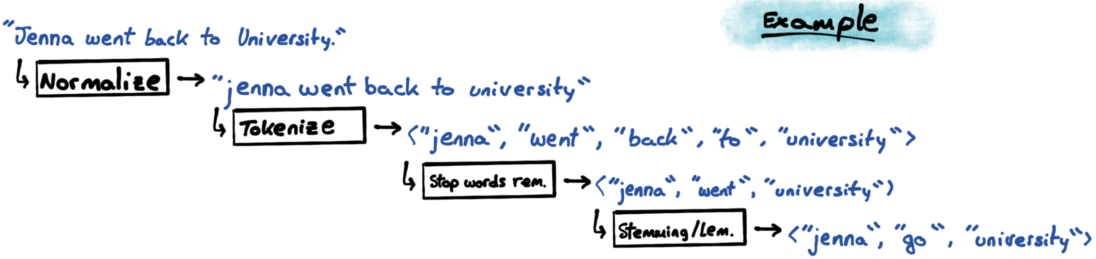
Example



"Jenna went back to University."

↳ Normalize → "jenna went back to university"

↳ Tokenize → <"jenna", "went", "back", "to", "university">

↳ Stop words rem. → <"jenna", "went", "university">

↳ Stemming/Lem. → <"jenna", "go", "university">

1. Capturing text data

- ↳ Plain text: With open
- ↳ Tabular data: Pandas
- ↳ Online resource: Requests

2. Cleaning

Regex woul need lots of arguments because of HTML, JS,...

- ↳ BeautifulSoup

find_all, select_one, get_text, strip

3. Normalization

- ↳ Lowercase
- ↳ Punctuation removal

Text Processing Steps

4. Tokenization

- ↳ Token: Individual words (generally)
- ↳ Text to tokens:

Use <word_tokenize / sent_tokenize> from NLTK

5. Stop word removal

- ↳ words that don't add a lot of meaning to a sentence (are, the, in, at, ...)

nltk.corpus.stopwords.words("english")

- ↳ Reduces the size of the input

6. Part-of-Speech tagging

- ↳ Identify nouns, pronouns, verbs, ...

↓

To better understand what is being said.

nltk.pos_tag(sentence) — sentence: word_tokenize("...")

- ↳ For custom grammar:

nltk.ChartParser(nltk.CFG.fromstring("""..."""))

- ↳ Named Entity Recognizion:

Use ne_chunk to identify named entities

7. Stemming and Lemmatization

Stemming is less memory intensive

↓

But is common to apply both stemming (first) and lemmatization

- ↳ Used to simplify text data
- ↳ Stemming: Reduce a word to its stem or root form.

Branchi~~ng~~
Branch~~ed~~
Branch~~es~~ } Branch ← nltk.stem.porter.PorterStemmer().stem(word)

- ↳ Lemmatization: Uses a dictionary to get the stem.

Is
Was
Were } Be ← nltk.stem.wordnet.WordNetLemmatizer().lemmatize(word, pos = "v")

The PoS parameter indicates the form of the converted word (v= verb, n = noun (default), ...)