Stemming

Stemming - standardizing words to their base stem irrespective of their inflections.

Morphemes

Smallest independent units

$$\begin{tikzcd}
 & \text{Smallest independent units} \arrow[dl] \arrow[d] \arrow[dr] & & \\
\text{affix / prefix} & \text{stem / base form} & \text{affix / suffix} \arrow[d] \arrow[dr] & \\
 & & \textbf{derivational} & \textbf{inflectional}
\end{tikzcd}$$

-s	'plural'
-'s	'possessive'
-ed	'past tense'
-s	'3rd person present tense'
-ing	'present participle'
-en	'past participle'
-er	'comparative'
-est	'superlative'

-ment
-ness
-ize

speak -> speaker

V -> N

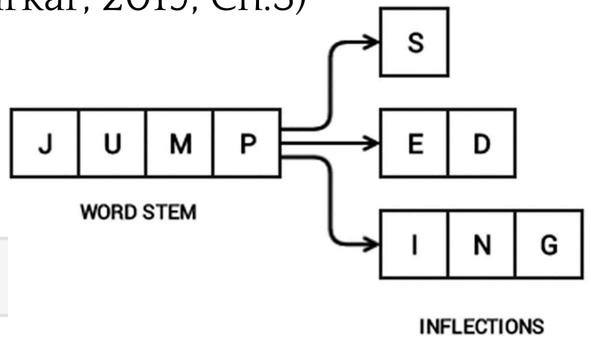
cat -> cats

N = N

https://pages.uoregon.edu/l150web/weblec1.4.html

Stemmers

(Sarkar, 2019, Ch.3)



PorterStemmer
- Strength: known for its simplicity and speed
- Weakness: It does not follow linguistics rather a set of 05 rules to strip suffixes

Porter Stemmer

```
from nltk.stem import PorterStemmer
```

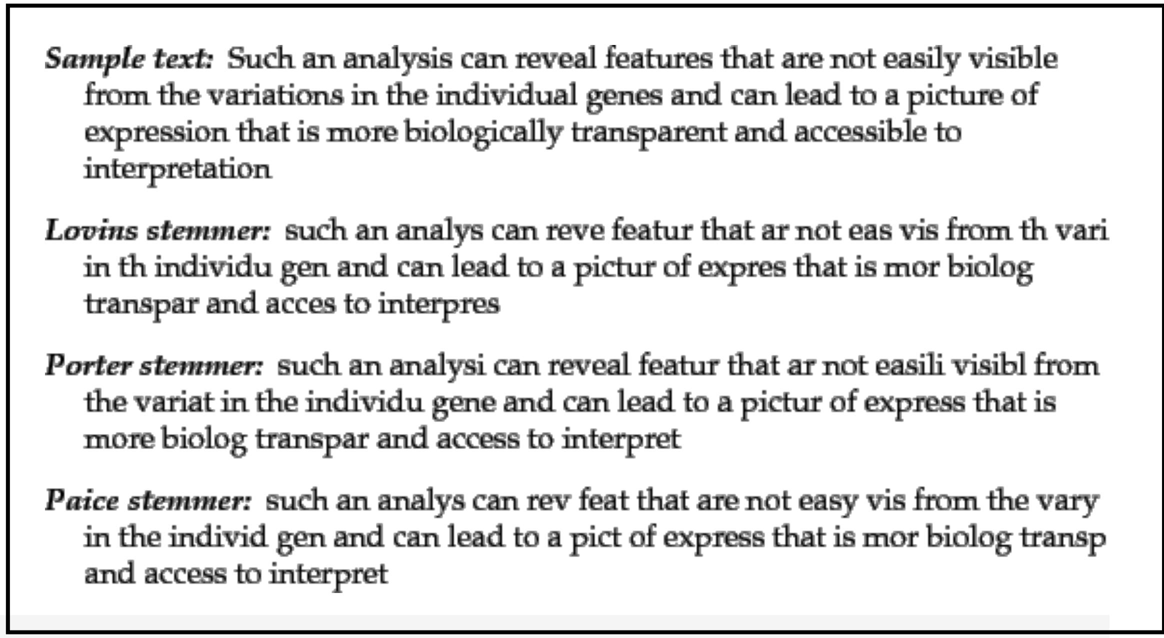
```
ps = PorterStemmer()
ps.stem('jumping'), ps.stem('jumps'), ps.stem('jumped')
```

```
('jump', 'jump', 'jump')
```

```
ps.stem('speak'),ps.stem('speaker')
```

```
('speak', 'speaker')
```

visible, vis, visibl
features, feature, featur, feat



Sample text: Such an analysis can reveal features that are not easily visible from the variations in the individual genes and can lead to a picture of expression that is more biologically transparent and accessible to interpretation

Lovins stemmer: such an analys can reve featur that ar not eas vis from th vari in th individu gen and can lead to a pictur of expres that is mor biolog transpar and acces to interpres

Porter stemmer: such an analysi can reveal featur that ar not easili visibl from the variat in the individu gene and can lead to a pictur of express that is more biolog transpar and access to interpret

Paice stemmer: such an analys can rev feat that are not easy vis from the vary in the individ gen and can lead to a pict of express that is mor biolog transp and access to interpret

SnowBall Stemmer

```
from nltk.stem import SnowballStemmer
print('Supported Languages:', SnowballStemmer.languages)
```

```
Supported Languages: ('arabic', 'danish', 'dutch', 'english', 'finnish', 'french', 'german', 'hungarian', 'italian', 'norwegian',
'porter', 'portuguese', 'romanian', 'russian', 'spanish', 'swedish')
```

https://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html

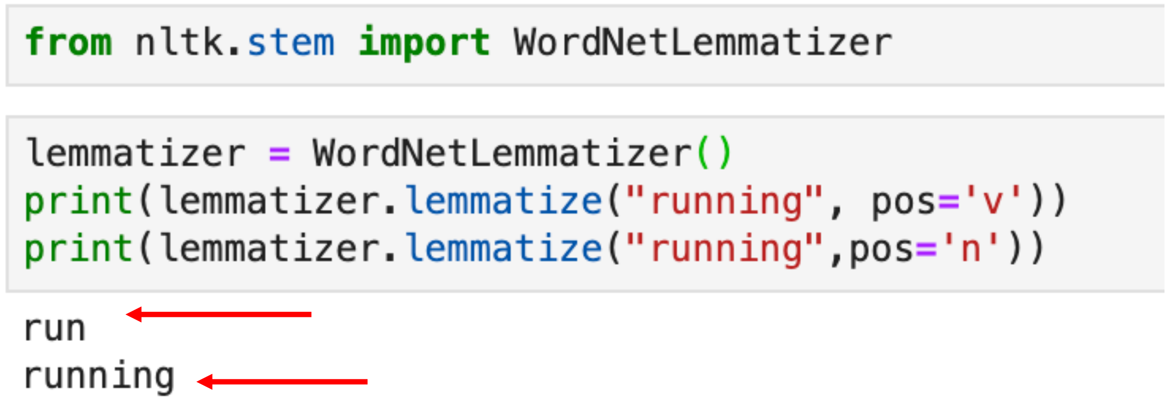
Lemmatization

- What is the POS?
- What affixes should be removed?
- Is the word in the dictionary after the removing affixes?

WordNet

a semantic lexicon for the English language

WordNet - a key component in IBM's Jeopardy-
playing Watson computer system



```
from nltk.stem import WordNetLemmatizer
```

```
lemmatizer = WordNetLemmatizer()
print(lemmatizer.lemmatize("running", pos='v'))
print(lemmatizer.lemmatize("running",pos='n'))
```

```
run
running
```

https://www.cbsnews.com/news/ibm-watson-defeats-humans-in-jeopardy/

http://mentalfloss.com/article/574505/oxford-english-dictionary-facts

Stemming and Lemmatization

Stemming – a base form

Lemmatization – a root word

- Stemming increases recall and decreases precision
- Lemmatization increases precision and decreases recall

Text Mining Applications

- text categorization
- text clustering
- concept/entity extraction
- production of granular taxonomies
- sentiment analysis
- document summarization
- entity relation modeling

Source: https://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html
https://queryunderstanding.com/stemming-and-lemmatization-6c086742fe45

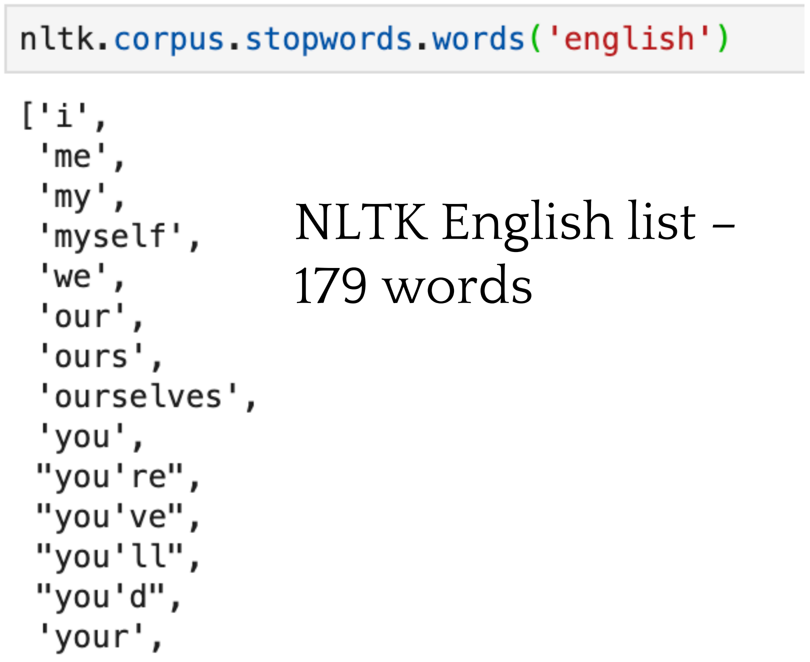
Stopwords

Functions Words

Prepositions (of, in)
Conjunctions (and)
Articles (a, the)
Auxiliary verbs (to be)

Lexical Words

Noun
Adjective
Verbs
Adverbs



```
nltk.corpus.stopwords.words('english')
```

```
['i',
 'me',
 'my',
 'myself',
 'we',
 'our',
 'ours',
 'ourselves',
 'you',
 "you're",
 "you've",
 "you'll",
 "you'd",
 'your',
```

NLTK English list – 179 words

Things to Consider:

- There is no universal stoplist
- General strategy: take the frequent words and filter for semantic content **relative to task and domain**

For example, consider whether you need to remove "not" and "no"

Sentiment analysis

I am happy versus I am NOT happy

Stopwords

```
from nltk.corpus import stopwords
stop_words = stopwords.words('english')
filtered_words = [w for w in words if not w in stop_words]
```

```
word_counts = collections.Counter(filtered_words)
word_counts.most_common(10)
```

```
[('said', 462),
 ('alice', 398),
 ('little', 128),
 ('one', 104),
 ('know', 88),
 ('like', 85),
 ('would', 83),
 ('went', 83),
 ('could', 77),
 ('queen', 75)]
```

```
word_counts = collections.Counter(words)
word_counts.most_common(10)
```

```
[('the', 1642),
 ('and', 872),
 ('to', 729),
 ('a', 632),
 ('it', 595),
 ('she', 553),
 ('i', 543),
 ('of', 514),
 ('said', 462),
 ('you', 411)]
```

```
extended_words = [w for w in words if not w in stop_words]
word_counts = collections.Counter(extended_words)
word_counts.most_common(10)
```

```
[('alice', 398),
 ('little', 128),
 ('know', 88),
 ('went', 83),
 ('queen', 75),
 ('thought', 74),
 ('time', 71),
 ('see', 67),
 ('well', 63),
 ('king', 63)]
```