• Lexicon Normalization

Lexicon normalization considers another type of noise in text. It reduces derivationally related forms of a word to a common root word.

① **Stemming**: It is a process of linguistic normalization, which reduces words to their word root word or chops off their derivational affixes.

For ex: connection, connected, connecting redue to 'connect'

Code

```
# Stemming
from nltk.stem import PorterStemmer
from nltk.tokenize import sent_tokenize, word_tokenize.
ps = PortStemmer()      ★★★
stemmed_words = []
for w in filtered_sent:
        stemmed_words.append(ps.stem(w))

print('Filtered Sent:', fi_sen)
print('stemmed sent:', stemmed_words)
```

O/P:

```
Filtered Sentence : ['Hello', 'Mr.', 'Smith', ',', 'today', '?']
Stemmed Sentence : ['hello', 'mr.', 'smith', ',', 'today', '?']
```

② **Lemmantization**: ★★★ (It reduces word to their base word, which is linguistically correct lemmas / meaningful meaningful.)

Stemmer works on on individual word without knowledge of context. This thing is miss by stemming because it requires dictionary look-up.
For ex: The word ~~good~~ better has ~~'better'~~ good as it's lemma.

Code: from nltk.stem.wordnet import WordNetLemmantizer

```
★★ lem = WordNetLemmatizer()
     word = 'flying'                        ★★
     print('Lemmatized':,                    lem.lemmatize(word, 'v'))
                                                 ↳ fly
     print('stem':,                          ps. ~~stem.~~ stem(word))
                                                 ↳ fli
```