Application of Big Data in Social Science

week 5

Heidi Hyeseung Choi
Fall 2022
HYSIS
heidichoi@hanyang.ac.kr

Announcements

~~9.1~~	~~1~~	~~Introduction~~
~~9.8~~	~~2~~	~~Web Scraping~~
~~9.15~~	~~3~~	
~~9.22~~	~~4~~	**Natural Language Processing**
9.29	5	
10.6	6	Text Analysis (recorded lecture on week 7)
10.13	7	
10.20	8	Mid term exam (as school schedule)
10.27	9	Social Network Analysis
11.3	10	
11.10	11	Machine Learning: Supervised Learning
11.17	12	
11.24	13	Machine Learning: Unsupervised Learning
12.1	14	
12.8	15	Data Visualization
12.15	16	Final Exam

Recap from last week

Natural Language Processing

- we used nltk library
- tokenization: to break down or split text into smaller pieces
    - word tokenization
    - sentence tokenization
- we made sure to make into lower cases when counting frequencies of words.
- stopwords: words that do not have much contribution to analyzing data
    - eg) a, the, of, and, he, she etc.

- List comprehension
- Regular Expressions

List comprehension

List comprehension with append option

```
>>> new_list= [expression for item in list (if conditional)]
```

```
>>> x= []
>>> my_list = [1,2,3]
>>> for a in my_list:
>>>       x.append(a*2)
```

```
x=[a*2 for a in my_list]
```

Regular Expressions

Regex documentation

Regex Syntax	What it means
[^…]	matches a character not present in the square brackets after the ^ symbol
|	OR operator
+	matches one or more cases of the previous mentioned regex before the + symbol
[A-Za-z]	matches all alphabets (upper and lower cases)
[a-zA-Z0-9_]	matches alpha-numeric characters (₩w)
[^a-zA-Z0-9_]	matches non-alpha-numeric characters (₩W)

re.sub(*pattern*, *repl*, *string*, *count=0*, *flags=0*)

This method is to substitute a specified regex pattern in a string with a replacement string.

Order of NLP

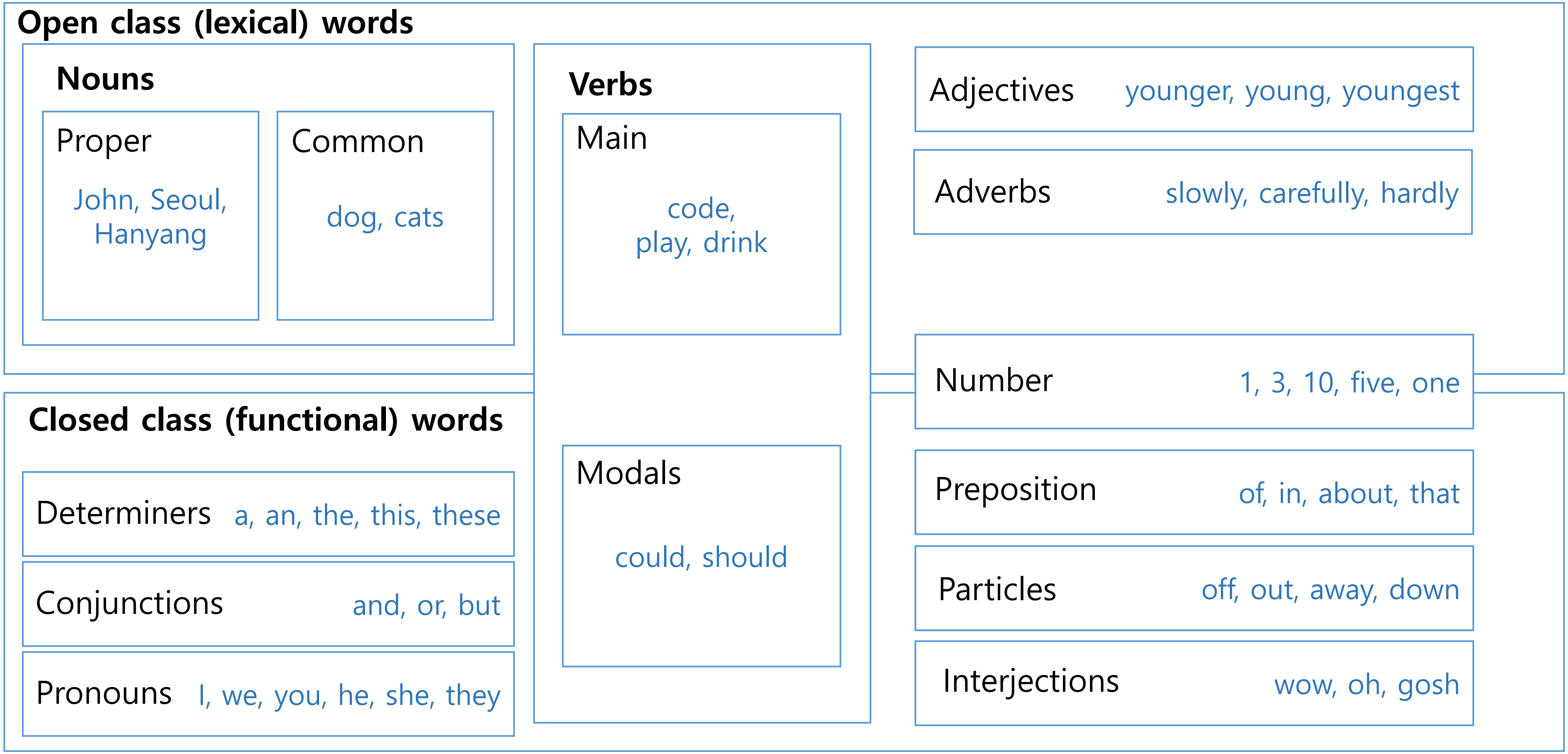
- HTML
- HTML parsing
- sentence segmentation
- word tokenization
- stopwords
- POS tagging
- text lemmatization

today's portion!

Part of speech (POS)

- There are eight parts of speech in the English language:
  - Noun… a person, place, thing, or idea (tree, Heidi, notebook, Seoul)
  - Pronoun… word used in place of a noun (she, we, they, it)
  - Verb… expresses action or being (go, write, teach, code)
  - Adjective… describes a noun or pronoun (small, old, blue, smart)
  - Adverb… describes a verb, an adjective or another adverb (extremely, well, carefully)
  - Preposition… placed before a noun to form a phrase modifying another word.. (by, with, about, until, in, on, of,)
  - Conjunction… joins words, phrases, or clauses (and, but, or, while, because)
  - Interjection…used to express emotions (oh! wow! oops!)

Part of speech (POS)



Open class (lexical) words

Nouns

Proper: John, Seoul, Hanyang

Common: dog, cats

Verbs

Main: code, play, drink

Modals: could, should

Adjectives: younger, young, youngest

Adverbs: slowly, carefully, hardly

Number: 1, 3, 10, five, one

Closed class (functional) words

Determiners: a, an, the, this, these

Conjunctions: and, or, but

Pronouns: I, we, you, he, she, they

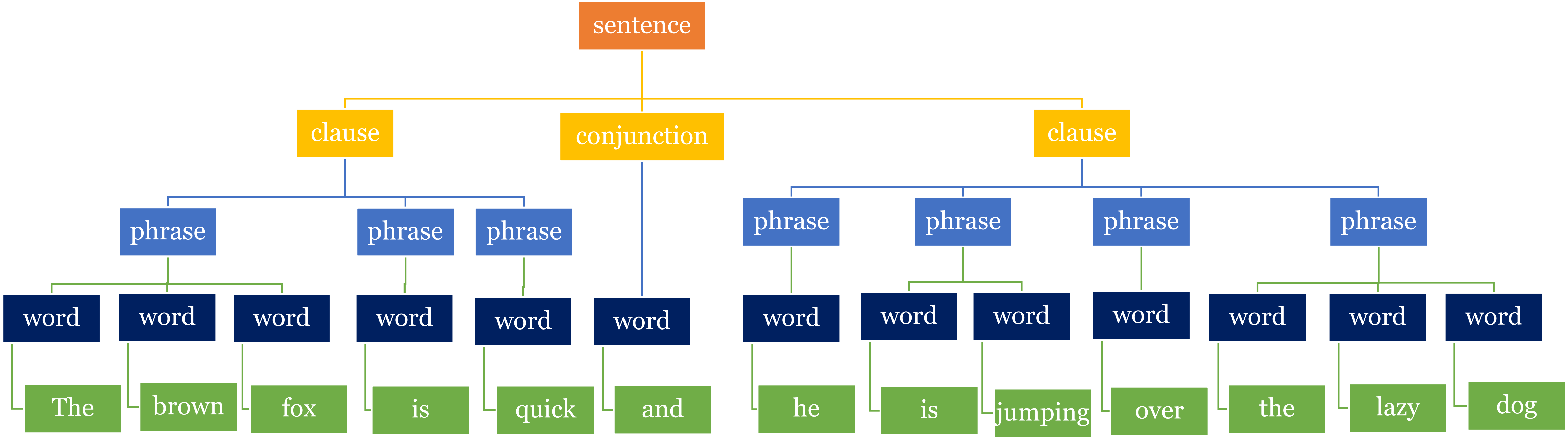
Preposition: of, in, about, that

Particles: off, out, away, down

Interjections: wow, oh, gosh

Language Syntax and Structure

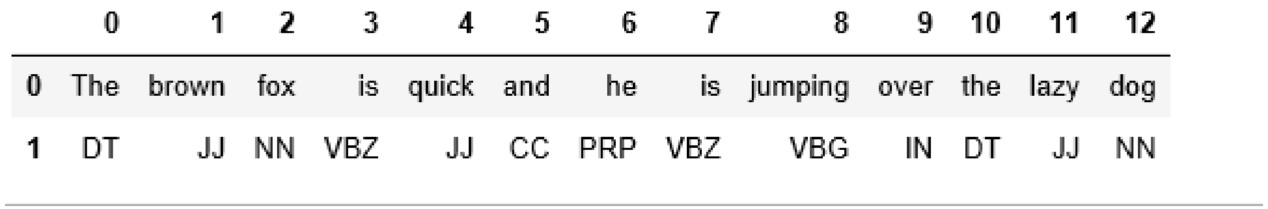
The brown fox is quick and he is jumping over the lazy dog.



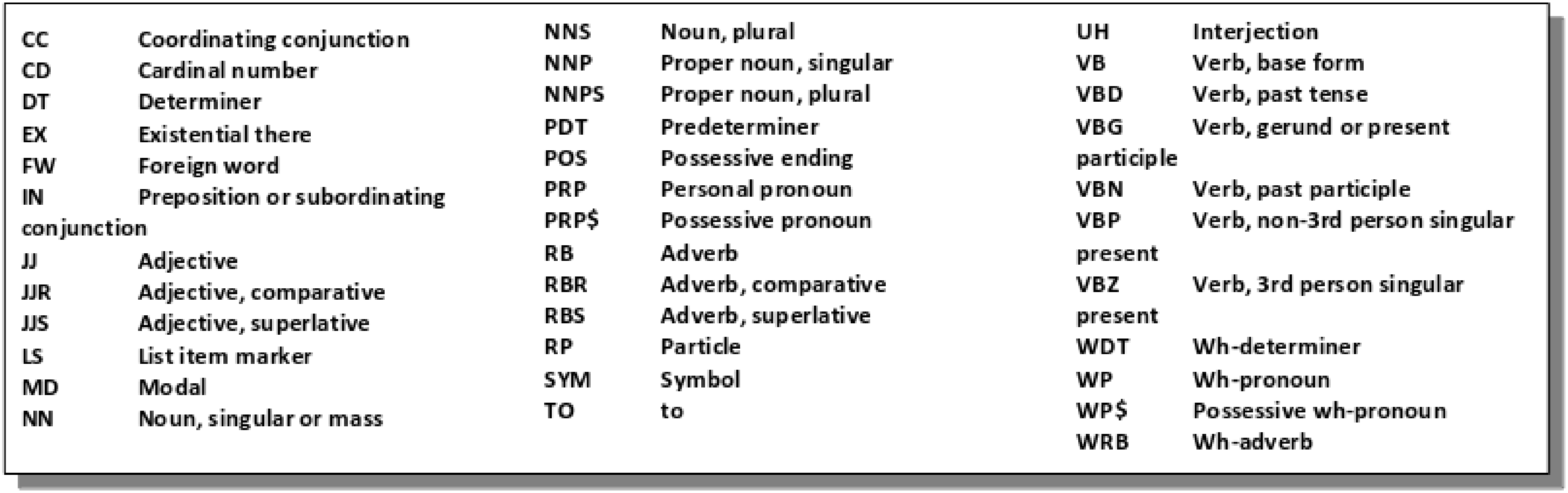
Hierarchical tree:
Sentence -> clauses -> phrases -> words

Words :POS tagging

- Words are the smallest unit in a language.
- It is useful to annotate and tag words then analyze them into their POS to see the major syntactic categories.



	0	1	2	3	4	5	6	7	8	9	10	11	12
0	The	brown	fox	is	quick	and	he	is	jumping	over	the	lazy	dog
1	DT	JJ	NN	VBZ	JJ	CC	PRP	VBZ	VBG	IN	DT	JJ	NN



CC	Coordinating conjunction
CD	Cardinal number
DT	Determiner
EX	Existential there
FW	Foreign word
IN	Preposition or subordinating conjunction
JJ	Adjective
JJR	Adjective, comparative
JJS	Adjective, superlative
LS	List item marker
MD	Modal
NN	Noun, singular or mass
NNS	Noun, plural
NNP	Proper noun, singular
NNPS	Proper noun, plural
PDT	Predeterminer
POS	Possessive ending
PRP	Personal pronoun
PRP$	Possessive pronoun
RB	Adverb
RBR	Adverb, comparative
RBS	Adverb, superlative
RP	Particle
SYM	Symbol
TO	to
UH	Interjection
VB	Verb, base form
VBD	Verb, past tense
VBG	Verb, gerund or present participle
VBN	Verb, past participle
VBP	Verb, non-3rd person singular present
VBZ	Verb, 3rd person singular present
WDT	Wh-determiner
WP	Wh-pronoun
WP$	Possessive wh-pronoun
WRB	Wh-adverb

Phrases

- Group of words make up phrases.
- Phrases are assumed to have at least two or more words.
- However, a phrase can be a single word or a combination of words based on the syntax and position of the phrase in a clause.

Noun phrase (NP): phrases where a noun acts as the head word. NPs at as a subject or object to a verb.

eg) “the lazy dog”, “the brown fox”

Verb phrase (VP): phrases where a verb acts as the head word.

eg) “It has been a while” => ‘has been’

Adjective phrase (ADJP):

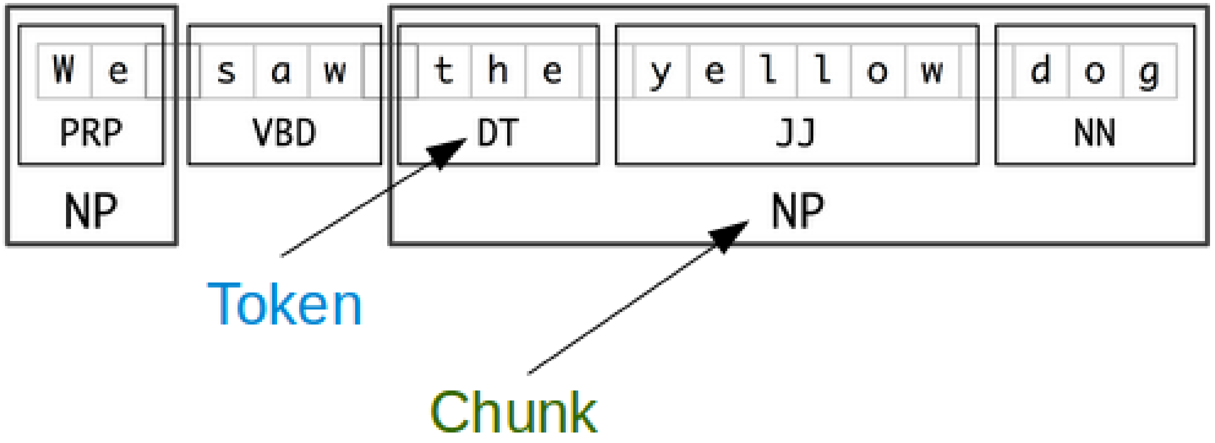
eg) “You are losing weight too fast” => ‘too fast’

Adverb phrase (ADVP):

eg) “I think it is pretty much the same” => ‘pretty much’

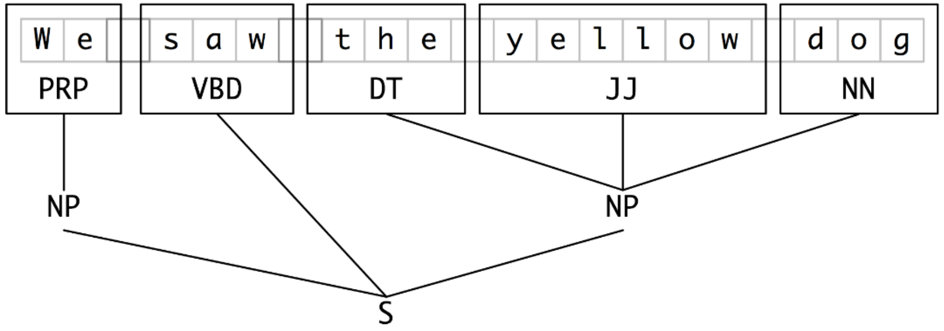
Prepositional phrase(PP):

eg) “Go down the hallway” => ‘down’



Chunking

- In part of speech tagging, we tag individual words.
- **Chunking** works on top of POS tagging and it chunks together set of tokens like Verb phrase or Noun phrase. The process of extracting phrases, or ‘chunks’ of texts from unstructured text.
- It is a very important concept if you are working with unstructured data and you want to obtain information from it.
- A common group of chunking is the noun phrase chunk (NP chunk).
- POS tags are used to create chunk grammar.
- Regex is used to make chunk grammar rules.



Chunking

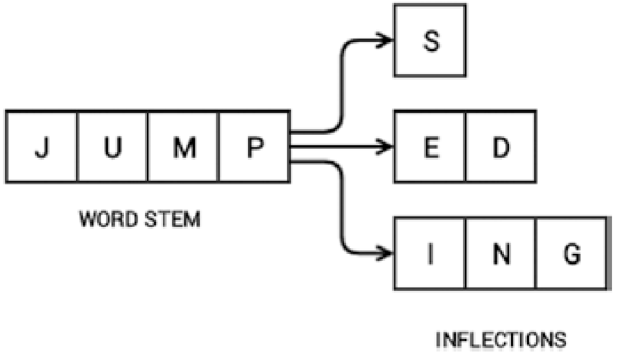
- Noun Phrase Chunk:
- It follows a rule which determines if the context it takes into consideration represents a Noun phrase.
- If the function finds a Determiner followed by an Adjective and then a Noun then the chunk will be tagged as a Noun Phrase.

```
grammar = "NP: {<DT>?<JJ>*<NN>}"
```

Your chunks:

1. Start with an optional (?) determiner ('DT')
2. Can have any number (*) of adjectives (JJ)
3. End with a noun (<NN>)

Word stemming



- Word stem is the base form of a word where we can create new words by attaching affixes to them.
- There are different algorithms, such as Porter Stemmer, Snowball Stemmer, Lancaster Stemmer etc.
- **Porter Stemmer:** The Porter stemming algorithm

  (or "Porter stemmer") uses suffix-stemming to produce stems.
- **Snowball Stemmer:** Upgraded version, a.k.a. Porter2 Stemmer.
- **Lancaster Stemmer:** Compared to snowball and porter stemming, lancaster is the most aggressive stemming algorithm because it tends to over-stem a lot of words. It tries to reduce the word to the shortest stem possible.

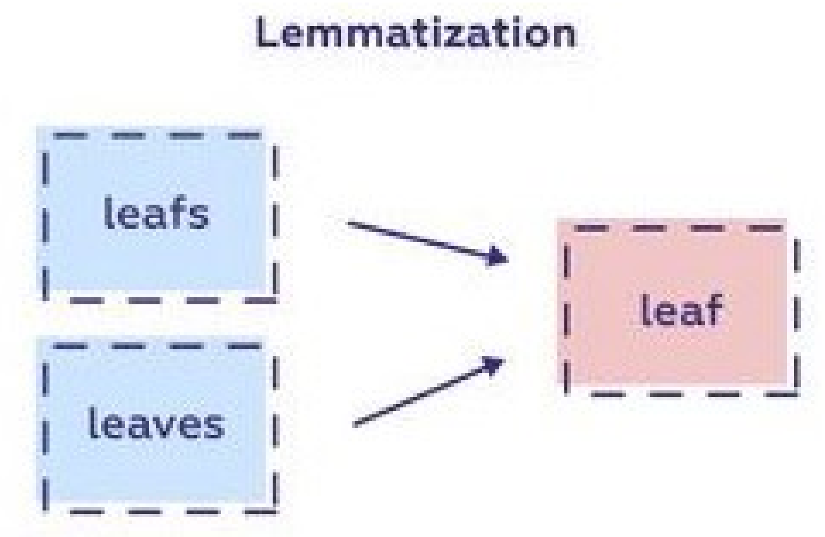
=> Stemming is fast, it is not 100% accurate.

Note!

The purpose of the Porter stemmer is not to produce complete words but to find variant forms of a word.

Lemmatization

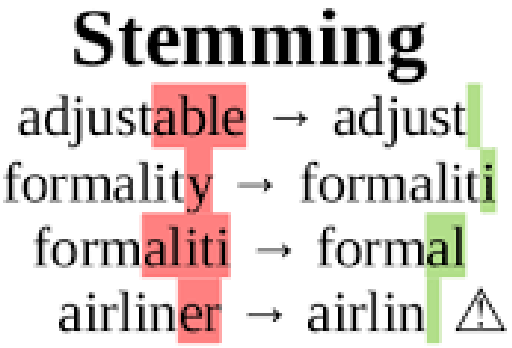
- Lemmatization is similar to stemming, but base form for lemmatization is the root word.
- In Lemmatization, the parts of speech(POS) will be determined first, unlike stemming which stems the word to its root form without considering the context.



- Lemmatization always considers the context and converts the word to its meaningful root/dictionary(WordNet) form called Lemma.
- It is important to do POS tagging before using this algorithm or it would assume every word as a noun.
- we will use NLTK package module 'WordNet' for lemmatization.

Word stemming & lemmatization

- **Stemming** usually refers to a crude heuristic process that chops off the ends of words in the hope of achieving this goal correctly most of the time, and often includes the removal of derivational affixes.
- **Lemmatization** usually refers to doing things properly with the use of a vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the lemma.

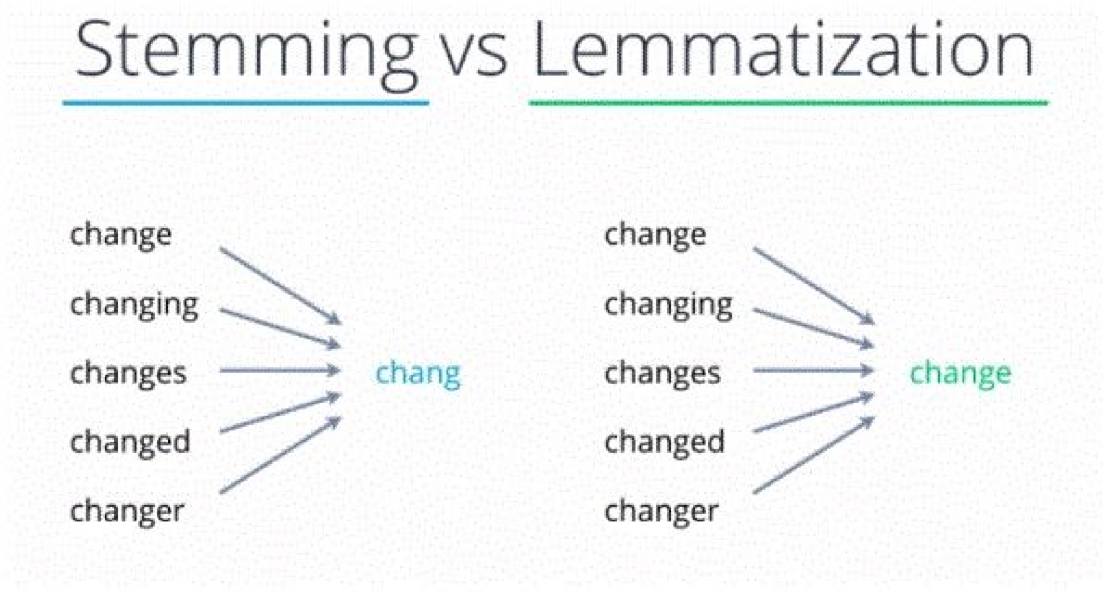


Stemming

adjustable → adjust
formality → formaliti
formaliti → formal
airliner → airlin ⚠

Lemmatization

was → (to) be
better → good
meeting → meeting



Let’s go to Jupyter notebook ☺