Tagging

NLTK Ch. 5. Categorizing and Tagging Words
http://www.nltk.org/book/ch05.html

Outline

- POS (part of speech)
- POS Tagger
- Tagged Corpora
- Korean POS Tagging

POS Tagging and Tagged Corpora

Parts-Of-Speech (POS)

- Word classes, or syntactic categories (Dionysius Thrax, 100 B.C.): noun, verb, pronoun, preposition, adverb, conjunction, participle, article
-
- Knowing POS of a word informs about likely neighboring words (nouns are preceded by determiners and adjectives, verbs by nouns) and syntactic structure word (nouns are generally part of noun phrases)
- Useful features for parsing, labeling named entities (e.g., people, organizations), coreference resolution, speech recognition, or synthesis
    - e.g., the word content is pronounced CONtent when it is a noun and conTENT when it is an adjective

https://web.stanford.edu/~jurafsky/slp3/8.pdf

English Word Classes

Closed classes have fixed membership

- prepositions: on, under, over, near, by, at, from, to, with
- particles: up, down, on, off, in, out, at, by
- determiners: a, an, the
- conjunctions: and, but, or, as, if, when
- pronouns: she, who, I, others
- auxiliary verbs: can, may, should, are
- numerals: one, two, three, first, second, third

Open classes have words that are continually being created or borrowed

- Nouns: **proper nouns** (e.g., iPhone), common nouns (e.g., book)
- Verbs (e.g., to fax)
- Adjectives, adverbs

SLP 8.1

POS tagging

- Assigning a part-of-speech marker to each word in a text
- Tagging is a **disambiguation task**
  - Book as a verb (**book** that flight) or a noun (hand me that **book**)
  - That as a determiner (Does **that** flight serve dinner) or a complementizer (I thought **that** your flight was earlier)
- 6 different parts-of-speech for the word **back**

  earnings growth took a **back/JJ** seat
  a small building in the **back/NN**
  a clear majority of senators **back/VBP** the bill
  Dave began to **back/VB** toward the door
  enable the country to buy **back/RP** about debt
  I was twenty-one **back/RB** then

SLP 8.3

Tag Ambiguity

- Most **word types** (80-86%) are unambiguous
- But the **ambiguous words (14-15%) are very common words** (55-67% of word tokens are ambiguous)

Types:		WSJ	Brown
Unambiguous	(1 tag)	44,432 (**86%**)	45,799 (**85%**)
Ambiguous	(2+ tags)	7,025 (**14%**)	8,050 (**15%**)
Tokens:			
Unambiguous	(1 tag)	577,421 (**45%**)	384,349 (**33%**)
Ambiguous	(2+ tags)	711,780 (**55%**)	786,646 (**67%**)

Figure 8.2 Tag ambiguity for word types in Brown and WSJ, using Treebank-3 (45-tag) tagging. Punctuation were treated as words, and words were kept in their original case.

SLP 8.3

Using a Tagger

- A part-of-speech tagger (POS-tagger) processes a sequence of words, and attaches a part of speech tag to each word
- Tag
  - ***CC***: coordinating conjunction
  - ***RB***: adverbs
  - ***IN***: preposition
  - ***NN***: noun
  - ***JJ***: adjective
- Represent using a tuple (*token, tag)*

```
>>> text = nltk.word_tokenize("And now for something completely different")
>>> nltk.pos_tag(text)
[('And', 'CC'), ('now', 'RB'), ('for', 'IN'), ('something', 'NN'), ('completely', 'RB'), ('different', 'JJ')]

>>> text = nltk.word_tokenize("They refuse to permit us to obtain the refuse permit")
>>> nltk.pos_tag(text)
[('They', 'PRP'), ('refuse', 'VBP'), ('to', 'TO'), ('permit', 'VB'), ('us', 'PRP'),
('to', 'TO'), ('obtain', 'VB'), ('the', 'DT'), ('refuse', 'NN'), ('permit', 'NN')]
```

Penn Treebank tagset (Marcus et al.,1993)

Tag	Description	Example	Tag	Description	Example	Tag	Description	Example
CC	coordinating conjunction	*and, but, or*	PDT	predeterminer	*all, both*	VBP	verb non-3sg present	*eat*
CD	cardinal number	*one, two*	POS	possessive ending	*'s*	VBZ	verb 3sg pres	*eats*
DT	determiner	*a, the*	PRP	personal pronoun	*I, you, he*	WDT	wh-determ.	*which, that*
EX	existential 'there'	*there*	PRP$	possess. pronoun	*your, one's*	WP	wh-pronoun	*what, who*
FW	foreign word	*mea culpa*	RB	adverb	*quickly*	WP$	wh-possess.	*whose*
IN	preposition/ subordin-conj	*of, in, by*	RBR	comparative adverb	*faster*	WRB	wh-adverb	*how, where*
JJ	adjective	*yellow*	RBS	superlatv. adverb	*fastest*	$	dollar sign	*$*
JJR	comparative adj	*bigger*	RP	particle	*up, off*	#	pound sign	*#*
JJS	superlative adj	*wildest*	SYM	symbol	*+,%, &*	"	left quote	' or "
LS	list item marker	*1, 2, One*	TO	"to"	*to*	"	right quote	' or "
MD	modal	*can, should*	UH	interjection	*ah, oops*	(	left paren	[, (, {, <
NN	sing or mass noun	*llama*	VB	verb base form	*eat*	)	right paren	],), }, >
NNS	noun, plural	*llamas*	VBD	verb past tense	*ate*	,	comma	,
NNP	proper noun, sing.	*IBM*	VBG	verb gerund	*eating*	.	sent-end punc	. ! ?
NNPS	proper noun, plu.	*Carolinas*	VBN	verb past part.	*eaten*	:	sent-mid punc	: ; ... – -

Figure 8.1 Penn Treebank part-of-speech tags (including punctuation).

https://web.stanford.edu/~jurafsky/slp3/8.pdf

Penn Treebank tagset example

(8.1) The/DT grand/JJ jury/NN commented/VBD on/IN a/DT number/NN of/IN other/JJ topics/NNS ./.

(8.2) **There/EX** are/VBP 70/CD children/NNS **there/RB**

(8.3) Preliminary/JJ findings/NNS were/VBD **reported/VBN** in/IN today/NN **'s/POS** New/NNP England/NNP Journal/NNP of/IN Medicine/NNP ./.

https://web.stanford.edu/~jurafsky/slp3/8.pdf

Tagged Corpora

- Corpora labeled with parts-of-speech are crucial training (and testing) sets for statistical tagging algorithms
  - **Brown corpus** (87 tags) is a million words of samples from 500 written texts from different genres published in the United States in 1961
  - **WSJ corpus** contains a million words published in the Wall Street Journal in 1989
  - **Switchboard** corpus consists of 2 million words of telephone conversations collected in 1990-1991
- These corpora were created by running an automatic part-of-speech tagger on the texts and then human annotators hand-corrected each tag

https://web.stanford.edu/~jurafsky/slp3/8.pdf

Tagged Corpora

- Construct a list of tagged tokens from a string

```
>>> sent = '''
... The/AT grand/JJ jury/NN commented/VBD on/IN a/AT number/NN of/IN
... other/AP topics/NNS ,/, AMONG/IN them/PPO the/AT Atlanta/NP and/CC
... Fulton/NP-tl County/NN-tl purchasing/VBG departments/NNS which/WDT it/PPS
... said/VBD ``/`` ARE/BER well/QL operated/VBN and/CC follow/VB generally/RB
... accepted/VBN practices/NNS which/WDT inure/VB to/IN the/AT best/JJT
... interest/NN of/IN both/ABX governments/NNS "/" ./.
... '''
>>> [nltk.tag.str2tuple(t) for t in sent.split()]  #convert each of these into a tuple
[('The', 'AT'), ('grand', 'JJ'), ('jury', 'NN'), ('commented', 'VBD'), ('on', 'IN'), ('a', 'AT'), ('number', 'NN'), ...
('.', '.')]
```

Reading Tagged Corpora

- NLTK corpora are tagged for their part-of-speech
  - a file from the Brown Corpus
- POS tags are converted to uppercase, since this has become standard practice since the Brown Corpus was published

```
>>> nltk.corpus.brown.tagged_words()
[('The', 'AT'), ('Fulton', 'NP-TL'), ...]
>>> nltk.corpus.brown.tagged_words(tagset='universal')
[('The', 'DET'), ('Fulton', 'NOUN'), ...]
>>> print(nltk.corpus.nps_chat.tagged_words())
[('now', 'RB'), ('im', 'PRP'), ('left', 'VBD'), ...]
>>> nltk.corpus.conll2000.tagged_words()
[('Confidence', 'NN'), ('in', 'IN'), ('the', 'DT'), ...]
>>> nltk.corpus.treebank.tagged_words()
[('Pierre', 'NNP'), ('Vinken', 'NNP'), (',', ','), ...]
>>> nltk.corpus.treebank.tagged_words(tagset='universal')
[('Pierre', 'NOUN'), ('Vinken', 'NOUN'), (',', '.'), ...]
```

Universal POS Tagset

- POS-tagging **classifies** words into their parts of speech and **labeling** them accordingly
- 장르별로 자주 쓰이는 tag 확인

```
>>> from nltk.corpus import brown
>>> brown_news_tagged = brown.tagged_words(categories='news', tagset='universal')
>>> tag_fd = nltk.FreqDist(tag for (word, tag) in brown_news_tagged)
>>> tag_fd.most_common()
[('NOUN', 30640), ('VERB', 14399), ('ADP', 12355), ('.', 11928), ('DET', 11389),  ('ADJ', 6706), ('ADV', 3349), ('CONJ', 2717), ('PRON', 2535), ('PRT', 2264), ('NUM', 2166), ('X', 106)]
```

Tag	Meaning	English Examples
ADJ	adjective	*new, good, high, special, big, local*
ADP	adposition	*on, of, at, with, by, into, under*
ADV	adverb	*really, already, still, early, now*
CONJ	conjunction	*and, or, but, if, while, although*
DET	determiner, article	*the, a, some, most, every, no, which*
NOUN	noun	*year, home, costs, time, Africa*
NUM	numeral	*twenty-four, fourth, 1991, 14:24*
PRT	particle	*at, on, out, over per, that, up, with*
PRON	pronoun	*he, their, her, its, my, I, us*
VERB	verb	*is, say, told, given, playing, would*
.	punctuation marks	*. , ; !*
X	other	*ersatz, esprit, dunno, gr8, univeristy*

Nouns

- people, places, things, or concepts
- Nouns can appear after determiners and adjectives, and can be the subject or object of the verb
- ***N* for common nouns**, and ***NP* for proper nouns**
- To see what parts of speech occur before a noun, with the most frequent ones first
  - construct a list of bigrams whose members are themselves word-tag pairs such as (('The', 'DET'), ('Fulton', 'NP'))
  - construct a FreqDist from the tag parts of the bigrams

```
>>> word_tag_pairs = nltk.bigrams(brown_news_tagged)
>>> noun_preceders = [a[1] for (a, b) in word_tag_pairs if b[1] ==
'NOUN']
>>> fdist = nltk.FreqDist(noun_preceders)
>>> [tag for (tag, _) in fdist.most_common()]
['NOUN', 'DET', 'ADJ', 'ADP', '.', 'VERB', 'CONJ', 'NUM', 'ADV',
'PRT', 'PRON', 'X']

# Nouns occur after determiners and adjectives, including
numeral adjectives (tagged as NUM).
```

Verbs

- Events and actions
- Express a relation involving the referents of noun phrases
- Code shows the most common verbs in news

```
>>> wsj = nltk.corpus.treebank.tagged_words(tagset='universal')
>>> word_tag_fd = nltk.FreqDist(wsj)
>>> [wt[0] for (wt, _) in word_tag_fd.most_common() if wt[1] == 'VERB']
['is', 'said', 'are', 'was', 'be', 'has', 'have', 'will', 'says', 'would', 'were', 'had', 'been', 'could',
"'s", 'can', 'do', 'say', 'make', 'may', 'did', 'rose', 'made', 'does', 'expected', 'buy', 'take',
'get', 'might', 'sell', 'added', 'sold', 'help', 'including', 'should', 'reported', ...]
```

Adjectives and Adverbs

- **Adjectives** describe nouns, and can be used as modifiers (e.g. large in the large pizza), or in predicates (e.g. the pizza is large)
- **Adverbs** modify verbs to specify the time, manner, place or direction of the event described by the verb (e.g. quickly in the stocks fell quickly). Adverbs may also modify adjectives (e.g. really in Mary's teacher was really nice)
- English has several **categories of closed class** words in addition to prepositions, such as articles (also often called determiners) (e.g., the, a), modals (e.g., should, may), and personal pronouns (e.g., she, they). Each dictionary and grammar classifies these words differently.

Tagged Corpora

- Words that follow *often*

```
>>> brown_learned_text = brown.words(categories='learned')
>>> sorted(set(b for (a, b) in nltk.bigrams(brown_learned_text) if a == 'often'))
[',', '.', 'accomplished', 'analytically', 'appear', 'apt', 'associated', 'assuming',
'became', 'become', 'been', 'began', 'call', 'called', 'carefully', 'chose', ...]

>>> brown_lrnd_tagged = brown.tagged_words(categories='learned', tagset='universal')
>>> tags = [b[1] for (a, b) in nltk.bigrams(brown_lrnd_tagged) if a[0] == 'often']
>>> fd = nltk.FreqDist(tags)
>>> fd.tabulate()
 PRT  ADV  ADP    .  VERB  ADJ
   2    8    7    4    37    6
```

Summary

- Words can be grouped into classes, such as nouns, verbs, adjectives, and adverbs
- Assigning parts of speech to words is called part-of-speech tagging, **POS tagging**, or just tagging
- Some linguistic corpora, such as the Brown Corpus, have been POS tagged.
- Taggers can be trained and evaluated using tagged corpora

Korean POS Tagging

Korean NLP resources

- 한글 처리 파이썬 패키지
  - koNLPy: http://konlpy-ko.readthedocs.io/ko/v0.4.3/#start
- 한글 corpus
  - 국립국어원 언어정보나눔터 : https://ithub.korean.go.kr/user/main.do
  - koNLP corpora link: http://konlpy.org/ko/latest/references/#corpora

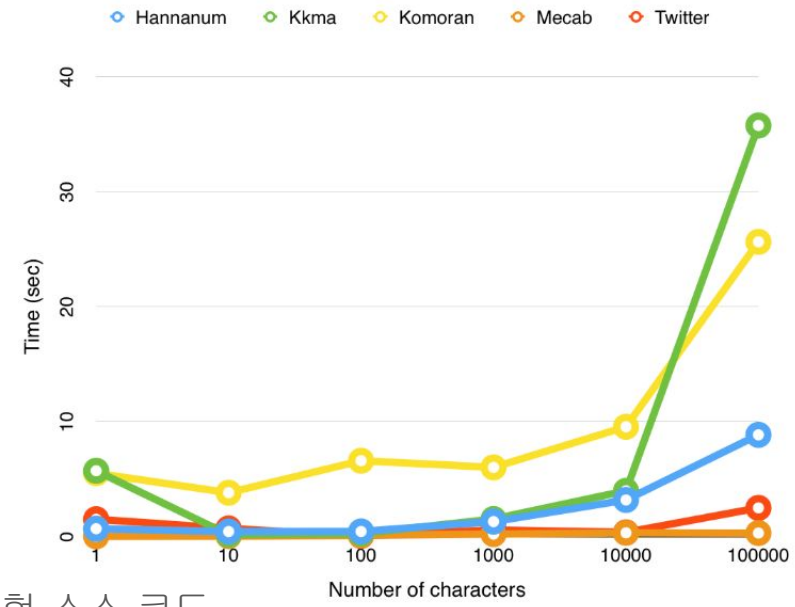
```
from konlpy.corpus import kolaw  # 법률 말뭉치. constitution.txt
from konlpy.corpus import kobill # 국회의안. 190890.txt ~ 1809899.txt
kolaw.open('constitution.txt').read()
kobill.open(1809890.txt').read()
```

KoNLPy installation

1. install ipykernel
   >> pip (혹은 pip3) install ipykernel
2. install jpype
   >> pip install JPype1-py3
3. install konlpy
   >> pip install konlpy

POS Tagger Evaluation

http://konlpy-ko.readthedocs.io/ko/v0.4.4/morph/#comparison-between-pos-tagging-classes



KoNLPy의 5개의 형태소 분석기중 Twitter(Okt)가 품사를 쉽게 읽을 수 있고 속도가 빨라 대용량 데이터 분석에 유용

Hannanum	Kkma	Komoran	Mecab	Twitter
아버지가방에 들어가 / N	아버지 / NNG	아버지가방에 들어가신다 / NNP	아버지 / NNG	아버지 / Noun
이 / J	가방 / NNG		가 / JKS	가방 / Noun
시ㄴ다 / E	에 / JKM		방 / NNG	에 / Josa
	들어가 / VV		에 / JKB	들어가신 / Verb
	시 / EPH		들어가 / VV	다 / Eomi
	ㄴ다 / EFN		신다 / EP+EC	

실험 소스 코드
https://github.com/konlpy/konlpy/blob/master/docs/morph.py

Twitter(Okt-Open Korean Text) POS Tag

- Noun 명사 (Nouns, Pronouns, Company Names, Proper Noun, Person Names, Numerals, Standalone, Dependent)
- Verb 동사
- Adjective 형용사
- Determiner 관형사 (ex: 새, 헌, 참, 첫, 이, 그, 저)
- Adverb 부사 (ex: 잘, 매우, 빨리, 반드시, 과연)
- Conjunction 접속사
- Exclamation 감탄사 (ex: 헐, 어머나, 얼씨구)
- Josa 조사 (ex: 의, 에, 에서)
- PreEomi 선어말어미 (ex: 었)
- Eomi 어미 (ex: 다, 요, 여, 하댘ㅋㅋ)

https://docs.google.com/spreadsheets/d/1OGAjUvalBuX-oZvZ_-9tEfYD2gQe7hTGsgUpiiBSXI8/edit#gid=0

Twitter examples

```
>>> from konlpy.tag import Okt
>>> okt = Okt()
>>> print(okt.morphs(u'단독입찰보다 복수입찰의 경우')) #형태소 단위로 분리
['단독', '입찰', '보다', '복수', '입찰', '의', '경우', '가']
>>> print(okt.nouns(u'유일하게 항공기 체계 종합개발 경험을 갖고 있는 KAI는')) # 명사 추출
['유일하', '항공기', '체계', '종합', '개발', '경험']
>>> print(okt.phrases(u'날카로운 분석과 신뢰감 있는 진행으로')) # 어절 추출
['분석', '분석과 신뢰감', '신뢰감', '분석과 신뢰감 있는 진행', '신뢰감 있는 진행', '진행', '신뢰']
>>> print(okt.pos(u'이것도 되나욬ㅋㅋ')) # 품사 태깅
[('이', 'Determiner'), ('것', 'Noun'), ('도', 'Josa'), ('되나욬', 'Noun'), ('ㅋㅋ', 'KoreanParticle')]
>>> print(okt.pos(u'이것도 되나욬ㅋㅋ', norm=True))
[('이', 'Determiner'), ('것', 'Noun'), ('도', 'Josa'), ('되', 'Verb'), ('나요', 'Eomi'), ('ㅋㅋ', 'KoreanParticle')]
>>> print(okt.pos(u'이것도 되나욬ㅋㅋ', norm=True, stem=True))
[('이', 'Determiner'), ('것', 'Noun'), ('도', 'Josa'), ('되다', 'Verb'), ('ㅋㅋ', 'KoreanParticle')]
```

```
from konlpy.tag import Okt
okt = Okt()

korea=u'아시아 대륙의 동쪽 끝 한반도에 있는 나라로서,
최초의 국가인 고조선은 BC 108년까지 존재했다. 고구려, 백제,
신라의 삼국시대를 거쳐 중세에는 고려가 세워졌으며, 이후
조선이 건립되어 근대까지 이어졌다. 현대 들어 35년의
일제강점기를 거쳐 제2차 세계대전 뒤 미국과 소련 군대의
한반도 분할 주둔으로 남북으로 나뉘었고 1948년 대한민국이
수립되었다. 이후 6·25전쟁이 일어나 휴전중이며, 현재까지
분단국가로 남아 있다'

sentences = korea.split(".")

for sentence in sentences:
    tagged = okt.pos(sentence, norm=True, stem=True)
    print(tagged)
```

실행 결과

```
[('아시아', 'Noun'), ('대륙', 'Noun'), ('의', 'Josa'), ('동쪽', 'Noun'),
('끝', 'Noun'), ('한반도', 'Noun'), ('에', 'Josa'), ('있다', 'Adjective'),
('나라', 'Noun'), ('로서', 'Noun'), (',', 'Punctuation'), ('최초',
'Noun'), ('의', 'Josa'), ('국가', 'Noun'), ('인', 'Josa'), ('고조선',
'Noun'), ('은', 'Josa'), ('BC', 'Alpha'), ('108', 'Number'), ('년',
'Noun'), ('까지', 'Josa'), ('존재', 'Noun'), ('하다', 'Verb')]

[('고구려', 'Noun'), (',', 'Punctuation'), ('백제', 'Noun'), (',',
'Punctuation'), ('신라', 'Noun'), ('의', 'Josa'), ('삼국시대', 'Noun'),
('를', 'Josa'), ('거치다', 'Verb'), ('중세', 'Noun'), ('에는', 'Josa'),
('고려', 'Noun'), ('가', 'Josa'), ('세워지다', 'Verb'), (',',
'Punctuation'), ('이후', 'Noun'), ('조선', 'Noun'), ('이', 'Josa'),
('건립', 'Noun'), ('되어다', 'Verb'), ('근대', 'Noun'), ('까지', 'Josa'),
('이어지다', 'Verb')]

[('현대', 'Noun'), ('들다', 'Verb'), ('35', 'Number'), ('년', 'Noun'),
('의', 'Josa'), ('일제강점기', 'Noun'), ('를', 'Josa'), ('거치다',
'Verb'), ('제', 'Noun'), ('2', 'Number'), ('차', 'Noun'), ('세계대전',
'Noun'), ('뒤', 'Noun'), ('미국', 'Noun'), ('과', 'Josa'), ('소련',
'Noun'), ('군대', 'Noun'), ('의', 'Josa'), ('한반도', 'Noun'), ('분할',
'Noun'), ('주둔', 'Noun'), ('으로', 'Josa'), ('남북', 'Noun'), ('으로',
'Josa'), ('나뉘다', 'Verb'), ('1948', 'Number'), ('년', 'Noun'),
('대한민국', 'Noun'), ('이', 'Josa'), ('수립', 'Noun'), ('되어다',
'Verb')]

[('이후', 'Noun'), ('6', 'Number'), ('·', 'Foreign'), ('25', 'Number'),
('전쟁', 'Noun'), ('이', 'Josa'), ('일어나다', 'Verb'), ('휴전', 'Noun'),
('중', 'Suffix'), ('이며', 'Josa'), (',', 'Punctuation'), ('현재', 'Noun'),
('까지', 'Josa'), ('분단국가', 'Noun'), ('로', 'Josa'), ('남아', 'Noun'),
('있다', 'Adjective')]
```

Examples

http://konlpy-ko.readthedocs.io/ko/v0.4.3/examples/