NLTK 자연어 처리 패키지

1692047 원진

설치

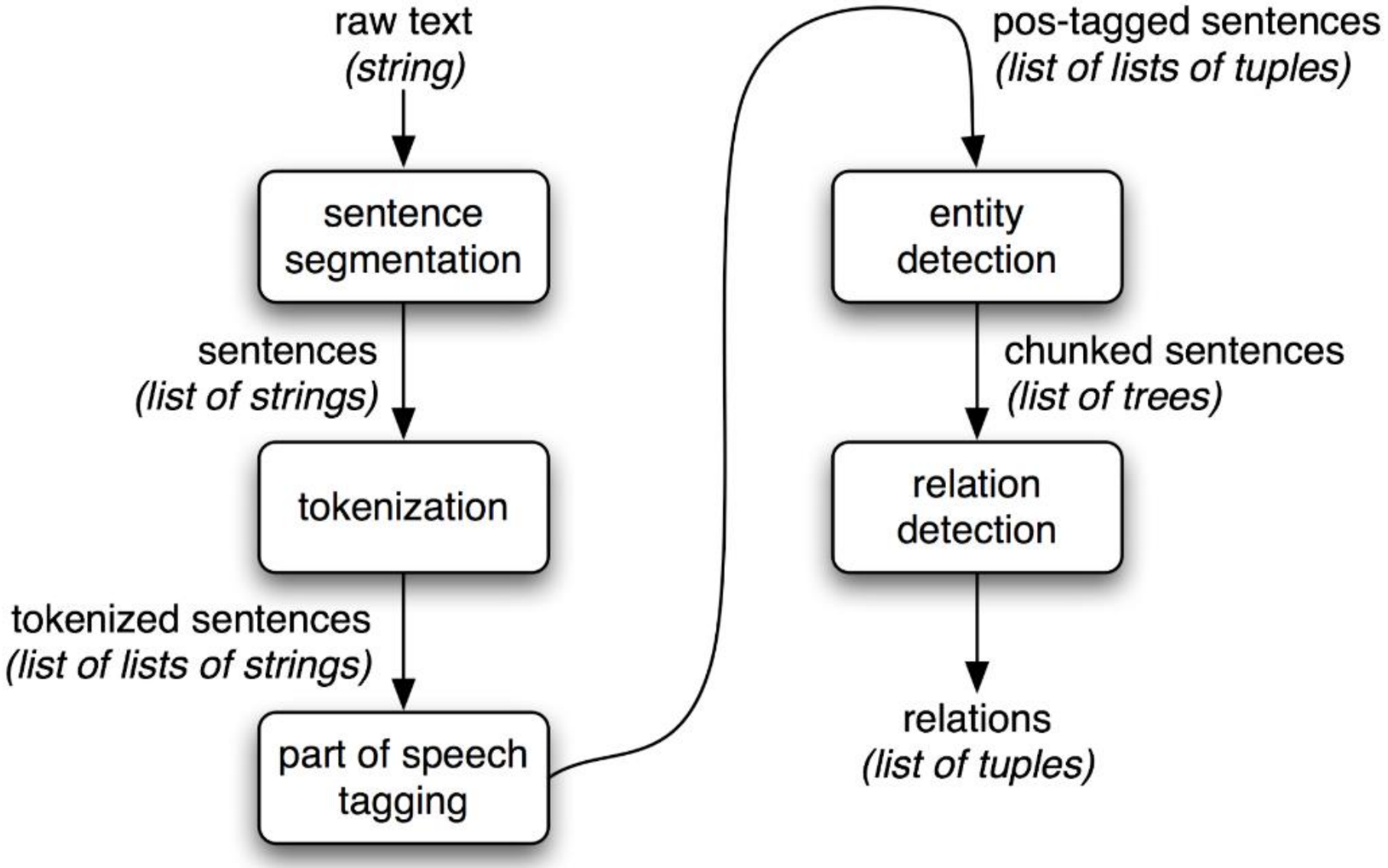
Mac / Unix

1. install NLTK — run `sudo pip install -U nltk`
2. install Numpy (optional) — run `sudo pip install –U numpy`
3. test installation — run `python` then type `import nltk`

Windows

1. install Python 3.7 — http://www.python.org/downloads/ (avoid the 64-bit versions)
2. install Numpy (optional) — https://www.scipy.org/scipylib/download.html
3. install NLTK — http://pypi.python.org/pypi/nltk
4. test installation — `start->Python37`, then type `import nltk`

Information Extraction Architecture



토큰 생성

token 긴 문자열을 분석을 위한 작은 단위로 나눈 것.

tokenizer 문자열을 토큰으로 분리하는 함수, 토큰 문자열의 리스트를 반환한다.

- sent_tokenize(text, language = 'english')

```
1  import nltk
2
3  corpus = 'hello, my name is jin! nice to meet you.'
4  print(nltk.sent_tokenize(corpus))
```

```
['hello, my name is wonjin!', 'nice to meet you.']
```

- word_tokenize(text, language='english', preserve_line=False)

```
1  import nltk
2
3  corpus = 'hello, my name is jin! nice to meet you.'
4  print(nltk.word_tokenize(corpus))
```

```
['hello', ',', 'my', 'name', 'is', 'jin', '!', 'nice', 'to', 'meet', 'you', '.']
```

토큰 생성

- 영문의 경우 정규 표현식을 쓸 수 있다.

```
1    import nltk
2
3    corpus = 'hello, my name is jin! nice to meet you.'
4
5    retokenize = nltk.tokenize.RegexpTokenizer("[\w]+")
6    retokenize.tokenize(corpus)
7
8    print(retokenize.tokenize(corpus))
```

```
['hello', 'my', 'name', 'is', 'jin', 'nice', 'to', 'meet', 'you']
```

형태소 분석

형태소 (morpheme)

언어학에서 일정한 의미가 있는 가장 작은 말의 단위.
보통 자연어 처리에서는 토큰으로 형태소를 이용함.

형태소 분석 (morphological analysis)

- 어간 추출 (stemming)
- 원형 복원 (lemmatizing)
- 품사 부착 (Part-Of-Speech tagging)

단어로부터 어근, 접두사, 접미사, 품사 등
다양한 언어적 속성을 파악하고,
이를 이용하여
형태소를 찾아내거나 처리하는 작업.

형태소 분석 – 어간 추출 (stemming)

변화된 단어의 접미사나 어미를 제거하여 같은 의미를 가지는 형태소의 기본형을 찾는 방법.
원형을 정확히 찾아주지는 않는다.

- PorterStemmer `A word stemmer based on the Porter stemming algorithm.`

```
import nltk

words = ['lives', 'crying', 'flies', 'dying']

st = nltk.stem.PorterStemmer()
print([st.stem(w) for w in words])
```

```
['live', 'cri', 'fli', 'die']
```

- LancasterStemmer `A word stemmer based on the Lancaster (Paice/Husk) stemming algorithm.`

```
import nltk

words = ['lives', 'crying', 'flies', 'dying']

st = nltk.stem.LancasterStemmer()
print([st.stem(w) for w in words])
```

```
['liv', 'cry', 'fli', 'dying']
```

형태소 분석 - 원형 복원 (lemmatizing)

같은 의미를 가지는 여러 단어를 사전형으로 통일하는 작업.
품사(part of speech)를 지정하는 경우 좀 더 정확한 원형을 찾을 수 있다.

wordnet 패키지를 추가로 다운받아야 함.

```
>>> import nltk
>>> nltk.download('wordnet')
```

```
1  import nltk
2
3  words = ['lives', 'crying', 'flies', 'dying']
4
5  lm = nltk.stem.WordNetLemmatizer()
6  print([lm.lemmatize(w) for w in words])
```

```
['life', 'cry', 'fly', 'dying']
```

```
1  import nltk
2
3  words = ['lives', 'crying', 'flies', 'dying']
4
5  lm = nltk.stem.WordNetLemmatizer()
6  print(lm.lemmatize("dying", pos="v"))
```

```
die
```

형태소 분석 - 품사 부착 (Part-Of-Speech tagging)

낱말을 문법적인 기능이나 형태, 뜻에 따라 구분한 것.
NLTK에서는 Penn Treebank Tagset을 이용함.

TAG	DESCRIPTION	EXAMPLE
CC	conjunction, coordinating	*and, or, but*
CD	cardinal number	*five, three, 13%*
DT	determiner	*the, a, these*
EX	existential there	*there were six boys*
FW	foreign word	*mais*
IN	conjunction, subordinating or preposition	*of, on, before, unless*
JJ	adjective	*nice, easy*
JJR	adjective, comparative	*nicer, easier*
JJS	adjective, superlative	*nicest, easiest*
LS	list item marker	
MD	verb, modal auxillary	*may, should*

형태소 분석 - 품사 부착 (Part-Of-Speech tagging)

TAG	DESCRIPTION	EXAMPLE
NN	noun, singular or mass	*tiger, chair, laughter*
NNS	noun, plural	*tigers, chairs, insects*
NNP	noun, proper singular	*Germany, God, Alice*
NNPS	noun, proper plural	*we met two Christmases ago*
PDT	predeterminer	*both his children*
POS	possessive ending	*'s*
PRP	pronoun, personal	*me, you, it*
PRP$	pronoun, possessive	*my, your, our*
RB	adverb	*extremely, loudly, hard*
RBR	adverb, comparative	*better*
RBS	adverb, superlative	*best*
RP	adverb, particle	*about, off, up*
SYM	symbol	*%*

형태소 분석 - 품사 부착 (Part-Of-Speech tagging)

TAG	DESCRIPTION	EXAMPLE
TO	infinitival to	*what to do?*
UH	interjection	*oh, oops, gosh*
VB	verb, base form	*think*
VBZ	verb, 3rd person singular present	*she thinks*
VBP	verb, non-3rd person singular present	*I think*
VBD	verb, past tense	*they thought*
VBN	verb, past participle	*a sunken ship*
VBG	verb, gerund or present participle	*thinking is fun*
WDT	*wh*-determiner	*which, whatever, whichever*
WP	*wh*-pronoun, personal	*what, who, whom*
WP$	*wh*-pronoun, possessive	*whose, whosever*
WRB	*wh*-adverb	*where, when*
.	punctuation mark, sentence closer	*.;?**

형태소 분석 - 품사 부착 (Part-Of-Speech tagging)

TAG	DESCRIPTION	EXAMPLE
,	punctuation mark, comma	,
:	punctuation mark, colon	:
(	contextual separator, left paren	(
)	contextual separator, right paren	)

- pos_tag 단어 토큰에 품사를 부착하여 튜플로 출력함.
  averaged_perceptron_tagger 패키지를 추가로 다운받아야 함.

```
>>> import nltk
>>> nltk.download('averaged_perceptron_tagger')
```

```
1  import nltk
2
3  sentence = "Emma refused to permit us to obtain the refuse permit"
4  tagged_list = nltk.tag.pos_tag(nltk.word_tokenize(sentence))
5
6  print(tagged_list)
```

```
[('Emma', 'NNP'), ('refused', 'VBD'), ('to', 'TO'), ('permit', 'VB'), ('us', 'PRP'), ('to', 'TO'), ('obtain', 'VB'),
```

형태소 분석 - 품사 부착 (Part-Of-Speech tagging)

- 품사 태그 정보를 사용하면 명사인 토큰만 선택할 수 있다.

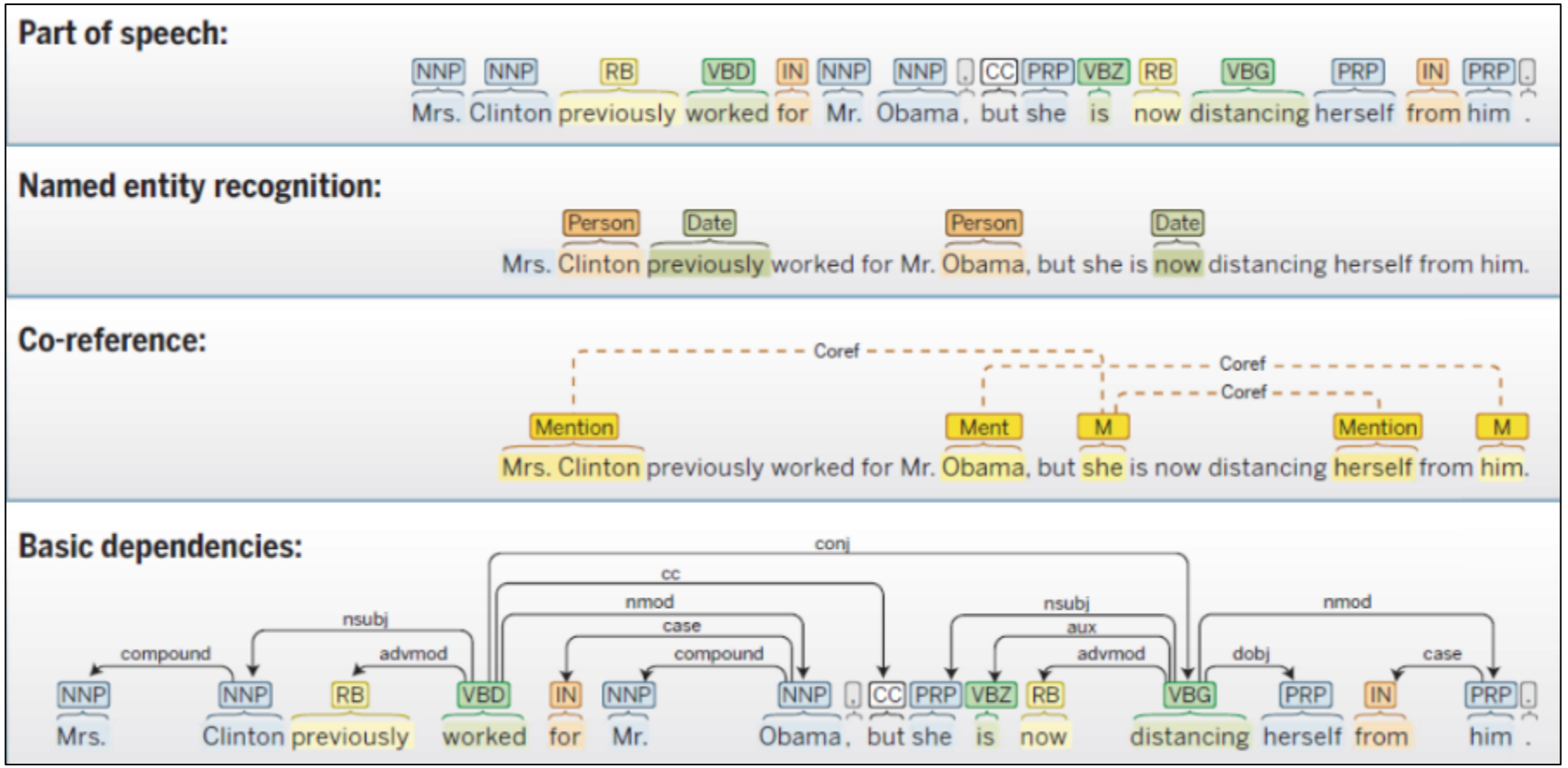
```
import nltk

sentence = "Emma refused to permit us to obtain the refuse permit"
tagged_list = nltk.tag.pos_tag(nltk.word_tokenize(sentence))

nouns_list = [t[0] for t in tagged_list if t[1] == "NN"]
print(nouns_list)
```

```
['refuse', 'permit']
```

어휘 분석 (Lexical Analysis)



Named Entity Recognition

개체 명 인식 (NER)

인명, 지명 등 고유 명사를 분류하는 방법론.

NE Type	Examples
ORGANIZATION	*Georgia-Pacific Corp., WHO*
PERSON	*Eddy Bonte, President Obama*
LOCATION	*Murray River, Mount Everest*
DATE	*June, 2008-06-29*
TIME	*two fifty a m, 1:30 p.m.*
MONEY	*175 million Canadian Dollars, GBP 10.40*
PERCENT	*twenty pct, 18.75 %*
FACILITY	*Washington Monument, Stonehenge*
GPE	*South East Asia, Midlothian*

NLTK에서 지원하는 기본 모듈로는 정확한 결과를 얻을 수 없다.
따라서 추가적인 데이터 셋을 확보해야 하며,
일반적으로 Stanford NER tagger가 사용된다.

Named Entity Recognition

Stanford NER Tagger

- Java JRE 8 이상 설치.
- JAVAHOME 환경변수 지정.
- dataset 경로 지정.

```
import nltk
import os
from nltk.tag.stanford import StanfordNERTagger

os.environ['JAVAHOME'] = "C:/Program Files/Java/jre1.8.0_191/bin/java.exe"
```

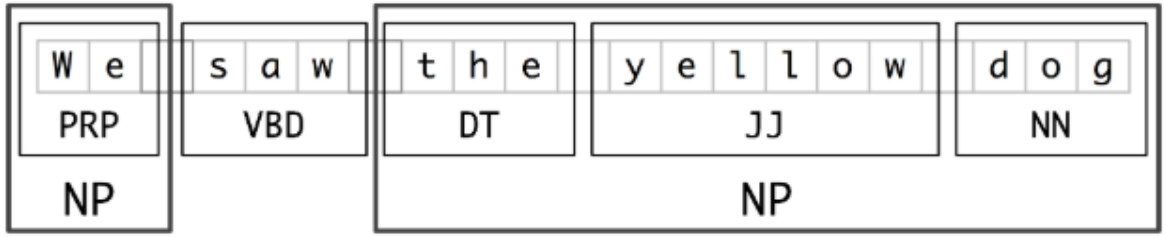
```
jar = './stanford-ner-3.9.2.jar'
model = './english.all.3class.distsim.crf.ser'
```

```
# Prepare NER tagger with english model
ner_tagger = StanfordNERTagger(model, jar, encoding='utf8')
```

Chunking

Chunk

구(phrase)로 묶어서 레이블을 지정함.



TAG	DESCRIPTION	WORDS	EXAMPLE
NP	noun phrase	DT+RB+JJ+NN + PR	*the strange bird*
PP	prepositional phrase	TO+IN	*in between*
VP	verb phrase	RB+MD+VB	*was looking*
ADVP	adverb phrase	RB	*also*
ADJP	adjective phrase	CC+RB+JJ	*warm and cosy*
SBAR	subordinating conjunction	IN	*whether or not*
PRT	particle	RP	*up the stairs*
INTJ	interjection	UH	*hello*