



형태소 분석

● 형태소 분석 개념

형태소(Morpheme)란?

자연어에서 **최소**의 의미 단위

형태소 분석(Morphological Analysis)이란?

자연어 문장에서 의미를 가진 최소 단위인 **형태소** 분석



명사, 동사, 형용사, 부사, 조사, 어미 등

자연어 처리를 위해 수행해야 하는 가장 첫 단계의 분석

● 형태소 분석 개념





● 자연어 처리 분석 과정

형태소 분석: 자연어 처리를 위한 분석 과정 중 가장 기본적인 시작 과정

자연어 처리는 일반적으로 형태소 분석, 구문 분석, 의미분석, 담화 분석 등의 과정으로 수행





● 자연어 처리 분석 과정

■ 형태소 분석(Morphological Analysis) 과정

형태소 분석기를 이용하여 형태소 분석 수행

파이썬을 이용한 형태소 분석



영문 형태소 분석기

NLTK(<http://www.nltk.org>)

한글 형태소 분석기

KoNLPy:(<http://konlpy.org/ko/latest>)

● 자연어 처리 분석 과정

▬ 형태소 분석(Morphological Analysis) 과정

▪ 형태소 분석 예

형태소 분석기

형태소 분석 문장

형태소
분석 결과

```
from eunjeon import Mecab  
tagger = Mecab()
```

```
pos = tagger.pos('나는 하늘을 나는 비행기를 보았다')
```

pos

```
[('나', 'NP'),  
( '는', 'JX'),  
( '하늘', 'NNG'),  
( '을', 'JKO'),  
( '나', 'NP'),  
( '는', 'JX'),  
( '비행기', 'NNG'),  
( '를', 'JKO'),  
( '보', 'VV'),  
( '았', 'EP'),  
( '다', 'EC')]
```



● 자연어 처리 분석 과정

■ 형태소 분석(Morphological Analysis) 과정

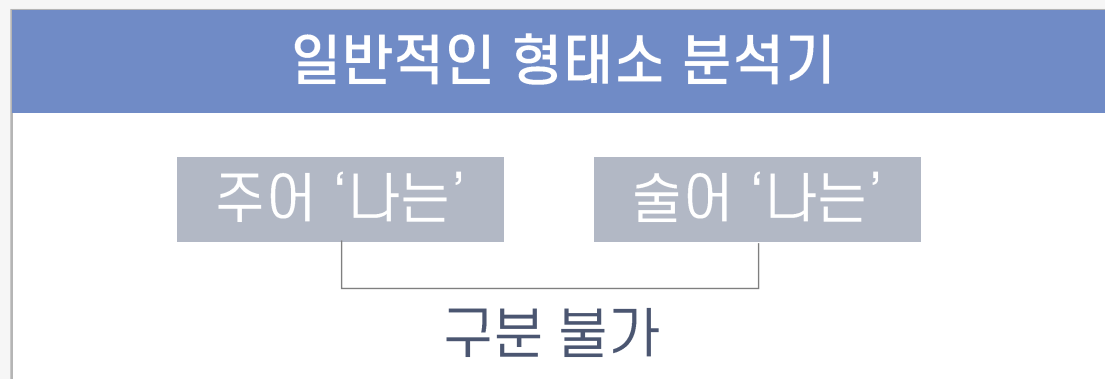
■ 품사 태깅표 예시(Mecab 형태소 분석기)

NNG	일반 명사	MAG	일반 부사	JX	보조사
NNP	고유 명사	MAJ	접속 부사	EP	선어말어미
NNB	의존 명사	IC	감탄사	EF	종결 어미
NNBC	단위를 나타내는 명사	JKS	주격 조사	EC	연결 어미
NR	수사	JKC	보격 조사	ETN	명사형 전성 어미
NP	대명사	JKG	관형격 조사	ETM	관형형 전성 어미
VV	동사	JKO	목적격 조사	XPN	체언 접두사
VA	형용사	JKB	부사격 조사	XSN	명사 파생 접미사
VX	보조 용언	JKV	호격 조사	XSV	동사 파생 접미사
VCP	긍정 지정사	JKQ	인용격 조사	XSA	형용사 파생 접미사
VCN	부정 지정사	JC	접속 조사	XR	어근
MM	관형사				

● 자연어 처리 분석 과정

■ 형태소 분석(Morphological Analysis) 과정

■ 형태소 분석기의 한계



정확도 높은 형태소 분석



딥러닝 '학습' 방법 사용을 통한 개선가능

● 자연어 처리 분석 과정

■ 형태소 분석(Morphological Analysis) 과정

■ 형태소 분석기의 한계

나	는	하늘	을	나	는	비행기	를	보	았	다
NP	JX	NNG	JKO	NP	JX	NNG	JKO	VV	EP	EC

나	는	하늘	을	나	는	비행기	를	보	았	다
NP	JX	NNG	JKO	VV	ETM	NNG	JKO	VV	EP	EC

● 자연어 처리 분석 과정

■ 구문분석(Syntactic Analysis) 과정

문장의 구조적 성질을
규칙화한 문법



문장의 구조를 분석하는
과정

나	는	하늘	을	나	는	비행기	를	보	았	다
NP	JX	NNG	JKO	VV	ETM	NNG	JKO	VV	EP	EC
주어		목적어		술어		목적어		술어		

● 자연어 처리 분석 과정

▬ 의미분석(Semantic Analysis) 과정

구문 분석결과에
해석을 가미



문장의 의미를
분석하는 과정

나	는	하늘	을	나	는	비행기	를	보	았	다
NP	JX	NNG	JKO	VV	ETM	NNG	JKO	VV	EP	EC
주어		목적어		술어		목적어		술어		
행동주 (Agent)		피동주 (Agent)		술어 (날다, Fly)		피동주 (Agent)		술어 (보다, See)		



● 자연어 처리 분석 과정

■ 담화분석(Discourse Analysis) 분석

대화의 맥락(Context)과
의도(Intent) 파악



문맥과 의도에 맞게
문장의 의미를 분석하는 과정

나는 하늘을 나는 비행기를 보았다

→ 나는 하늘에서 날고 있는 비행기를 보았다

나는 아메리카노 그녀는 라떼

→ 나는 아메리카노를 주문했고 그녀는 라떼를 주문했다

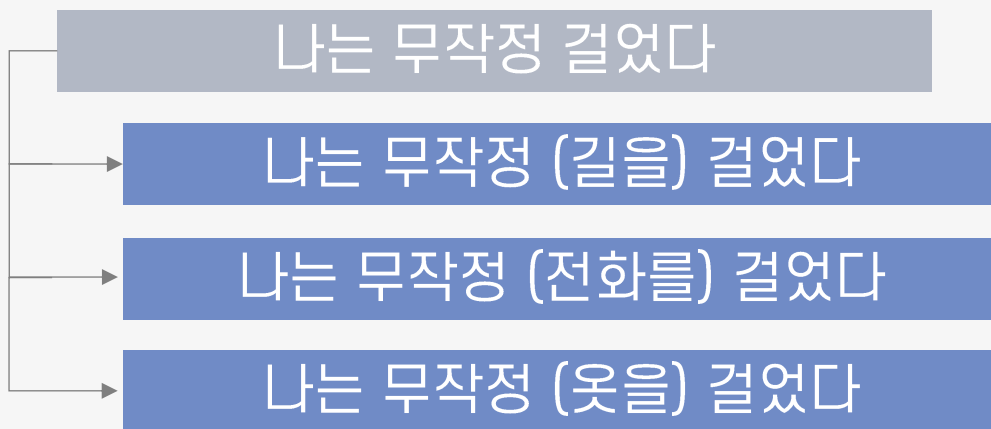
● 자연어 처리 분석 과정

▬ 담화분석(Discourse Analysis) 분석

대화의 맥락(Context)과
의도(Intent) 파악



문맥과 의도에 맞게
문장의 의미를 분석하는 과정



- 영문 형태소 분석기 소개 및 설치(Python, NLTK)

NLTK 패키지란?

교육용으로 개발된 자연어 처리 및 문서 분석용 파이썬 패키지



- 영문 형태소 분석기 소개 및 설치(Python, NLTK)

- ▬ NLTK 패키지 주요 기능



텍스트 분석에 유용한 다양한 메서드를 포함하는 Text 클래스 제공



- 영문 형태소 분석기 사용방법
 - ▬ NLTK 연구용 코퍼스 다운로드

저작권이 말소된 문학작품을 포함하는 gutenbergl 코퍼스 다운로드

```
import nltk
nltk.download('gutenberg')
```

gutenberg 코퍼스 목록

```
from nltk.corpus import gutenberg
print(gutenberg.fileids())
```


● 영문 형태소 분석기 사용방법

▬ NLTK 연구용 코퍼스 다운로드

저작권이 말소된 문학작품을 포함하는 gutenbergr 코퍼스 다운로드

```
import nltk  
nltk.download('gutenberg')
```

gutenbergr 코퍼스 목록

```
from nltk.corpus import gutenberg  
print(gutenberg.fileids())
```

```
['austen-emma.txt',  
'austen-persuasion.txt',  
'austen-sense.txt',  
'bible-kjv.txt',  
'blake-poems.txt',  
'bryant-stories.txt',  
'burgess-busterbrown.txt',  
'carroll-alice.txt',  
'chesterton-ball.txt',  
'chesterton-brown.txt',  
'chesterton-thursday.txt',  
'edgeworth-parents.txt',  
'melville-moby_dick.txt',  
'milton-paradise.txt',  
'shakespeare-caesar.txt',  
'shakespeare-hamlet.txt',  
'shakespeare-macbeth.txt',  
'whitman-leaves.txt']
```



● 영문 형태소 분석기 사용방법

▬ NLTK 연구용 코퍼스 다운로드

hamlet 코퍼스 로드

- gutenberg 코퍼스 중 셰익스피어의 햄릿 코퍼스 목록 로드

```
hamlet = nltk.corpus.gutenberg.raw('shakespeare-hamlet.txt')  
print(hamlet)
```



● 영문 형태소 분석기 사용방법

▬ NLTK 연구용 코퍼스 다운로드

[The Tragedie of Hamlet by William Shakespeare 1599]

Actus Primus. Scoena Prima.

Enter Barnardo and Francisco two Centinels.

Barnardo. Who's there?

Fran. Nay answer me: Stand & vnfold
your selfe

Bar. Long liue the King

Fran. Barnardo?

Bar. He

Fran. You come most carefully vpon your houre

Bar. 'Tis now strook twelue, get thee to bed Francisco

Fran. For this releefe much thankes: 'Tis bitter cold,
And I am sicke at heart

Barn. Haue you had quiet Guard?

Fran. Not



● 영문 형태소 분석기 사용방법

▬ 형태소 분석

Tagger 다운로드

```
import nltk
nltk.download('averaged_perceptron_tagger_eng')
```

품사 태깅

```
from nltk.tag import pos_tag
tagged_list = pos_tag(word_tokenize(hamlet))
print(tagged_list)
```

```
[('You', 'PRP'),
 ('come', 'VBP'),
 ('most', 'RBS'),
 ('carefully', 'RB'),
 ('vpon', 'VB'),
 ('your', 'PRP$'),
 ('houre', 'NN')]
```



● 영문 형태소 분석기 사용방법

▬ 형태소 분석

품사 태깅

```
from nltk.tag import pos_tag  
tagged_list = pos_tag(word_tokenize(hamlet))  
print(tagged_list)
```

```
[('You', 'PRP'),  
 ('come', 'VBP'),  
 ('most', 'RBS'),  
 ('carefully', 'RB'),  
 ('vpon', 'VB'),  
 ('your', 'PRP$'),  
 ('houre', 'NN')]
```

품사 태깅 제거

```
from nltk.tag import untag  
untag(tagged_list)
```

```
['You', 'come', 'most', 'carefully', 'vpon', 'your', 'houre']
```



- 영문 형태소 분석기 사용방법

- 영문 형태소 분석 실습

실습

2.1 영문 형태소 분석 실습_colab.ipynb



● 한글 형태소 분석기 설치(Python, KoNLPy)

Java JDK 설치



JDK 1.8 설치

- 이미 설치되어 있는 경우 생략 가능

JPytype1 설치



pip Install JPytype1

KoNLPy 설치



pip Install KoNLPy



한글 품사 태깅 클래스 종류 및 성능비교

한글 품사 태깅 클래스 종류

- KoNLPy의 한글 품사 태깅 클래스 종류

Kkma

Komoran

Hannanum

Okt(previous Twitter)

Mecab

Windows 환경에서는
Mecab 미지원



별도 설치 필요



● 한글 품사 태깅 클래스 종류 및 성능비교

▬ 품사 태깅 클래스 품사 태깅 성능 비교

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
시~다 / E	에 / JKM		방 / NNG	에 / Josa
	들어가 / W		에 / JKB	들어가신 / Verb
	시 / EPH		들어가 / W	다 / Eomi
	s다 / EFN		신다 / EP+EC	

<출처: KoNLPy 형태소 분석 및 품사 태깅, <https://konlpy.org/ko/latest/morph/>>



● 한글 품사 태깅 클래스 종류 및 성능비교

▬ 수행시간 성능 비교

품사 태깅 클래스	클래스 로딩 시간	10만 문자 품사 태깅 실행시간
Kkma	5.6988 secs	35.7163 secs
Komoran	5.4866 secs	25.6008 secs
Hannanum	0.6591 secs	8.8251 secs
Okt(previous Twitter)	1.4870 secs	2.4714 secs
Mecab	0.0007 secs	0.2838 secs

<출처: KoNLPy 형태소 분석 및 품사 태깅, <https://konlpy.org/ko/latest/morph/>>



● KoNLPy를 이용한 형태소 분석 방법

▬ 파이썬에서 KoNLPy 사용

KoNLPy 클래스 Import

```
from konlpy.tag import Okt, Kkma, Komoran, Hannanum
```

사용하고자 하는 클래스 객체 생성

```
tagger = Okt()  
tagger = Kkma()  
tagger = Komoran()  
tagger = Hannanum()
```



- KoNLPy를 이용한 형태소 분석 방법

- ▬ 파이썬에서 KoNLPy 사용

형태소 분석 수행

```
tagger.morphs(txt)
```

명사만 추출

```
tagger.nouns(txt)
```

품사 태깅

```
tagger.pos(txt)
```



● KoNLPy를 이용한 형태소 분석 방법

▬ 파이썬에서 KoNLPy 사용 예시

```
from konlpy.tag import Okt, Kkma, Komoran, Hannanum
```

```
# tagger = Kkma()  
# tagger = Komoran()  
# tagger = Hannanum()  
tagger = Okt()
```

```
txt = '나는 하늘을 나는 비행기를 보았습니다.'
```



● KoNLPy를 이용한 형태소 분석 방법

▬ 파이썬에서 KoNLPy 사용 예시

```
tagger.morphs(txt)
```

```
['나', '는', '하늘', '을', '나', '는', '비행기', '를', '보았습니다', '.']
```

```
tagger.nouns(txt)
```

```
['나', '하늘', '나', '비행기']
```



● KoNLPy를 이용한 형태소 분석 방법

▬ 파이썬에서 KoNLPy 사용 예시

```
tagger.pos(txt)
```

```
[('나', 'Noun'),  
 ('는', 'Josa'),  
 ('하늘', 'Noun'),  
 ('을', 'Josa'),  
 ('나', 'Noun'),  
 ('는', 'Josa'),  
 ('비행기', 'Noun'),  
 ('를', 'Josa'),  
 ('보았습니다', 'Verb'),  
 ('.', 'Punctuation')]
```

● KoNLPy를 이용한 형태소 분석 방법

- ▬ 윈도우즈 환경에서 Python Mecab 설치 및 사용
 - 윈도우즈 환경에서 설치 방법

리눅스나 IOS에서는 설치가 용이함



KoNLPy를 통한 Mecab형태소 분석기 설치 및 사용



윈도우즈 환경에서는 설치가 용이하지 않음

- Konlpy를 이용한 형태소 분석 방법

- 윈도우즈 환경에서 Python Mecab 설치 및 사용

- 윈도우즈 환경에서 설치 방법

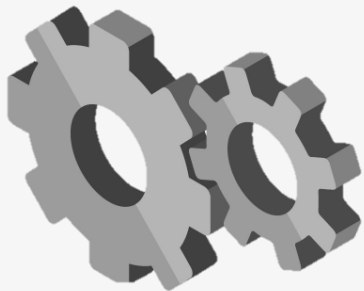
윈도우즈 환경에서는 별도의 eunjeon 패키지를 통해 Mecab 설치 및 사용 용이

Pyeunjeon이란?



은전한닢 프로젝트와 Mecab 기반의
한국어 형태소 분석기의 독립형 Python 인터페이스

- Konlpy를 이용한 형태소 분석 방법
 - ▬ 윈도우즈 환경에서 Python Mecab 설치 및 사용
 - 윈도우즈 환경에서 설치 방법

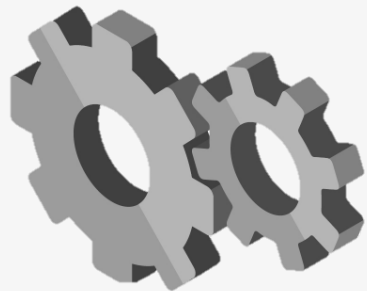


윈도우즈의 환경에서
eunjeon 패키지 설치



```
pip install eunjeon
```

- Konlpy를 이용한 형태소 분석 방법
 - ▬ 리눅스 환경에서 Python Mecab 설치 및 사용
 - 리눅스 환경에서 설치 방법



리눅스 colab 환경에서
mecab 패키지 설치



```
!git clone https://github.com/SOMJANG/Mecab-ko-for-Google-Colab.git
%cd Mecab-ko-for-Google-Colab
!bash install_mecab-ko_on_colab_light_220429.sh
```



● 한글 형태소 분석기 사용방법

■ 한글 형태소 분석 실습

실습

2.2 한글 형태소 분석 실습_colab.ipynb

2.3 한글 형태소 분석 실습2_colab.ipynb

ETRI OpenAPI : <https://epretx.etri.re.kr/>

● JSON 데이터 타입

▬ JSON 데이터 타입의 종류

값	타입
1	숫자(number)
"str"	문자열(string)
true	불리언(boolean)
{inKey : value}	객체(object)
["일", "이"]	배열(array)
null	널(NULL)

● JSON 데이터 표현

▬ JSON 데이터 표현

이름	나이	직업	도시	취미	가족.배우자	가족.자녀	활성_사용자
홍길동	30	개발자	서울	코딩, 독서, 등산	김영희	2	TRUE

JSON

```
{
  "이름": "홍길동",
  "나이": 30,
  "직업": "개발자",
  "도시": "서울",
  "취미": ["코딩", "독서", "등산"],
  "가족": {
    "배우자": "김영희",
    "자녀": 2
  },
  "활성_사용자": true
}
```