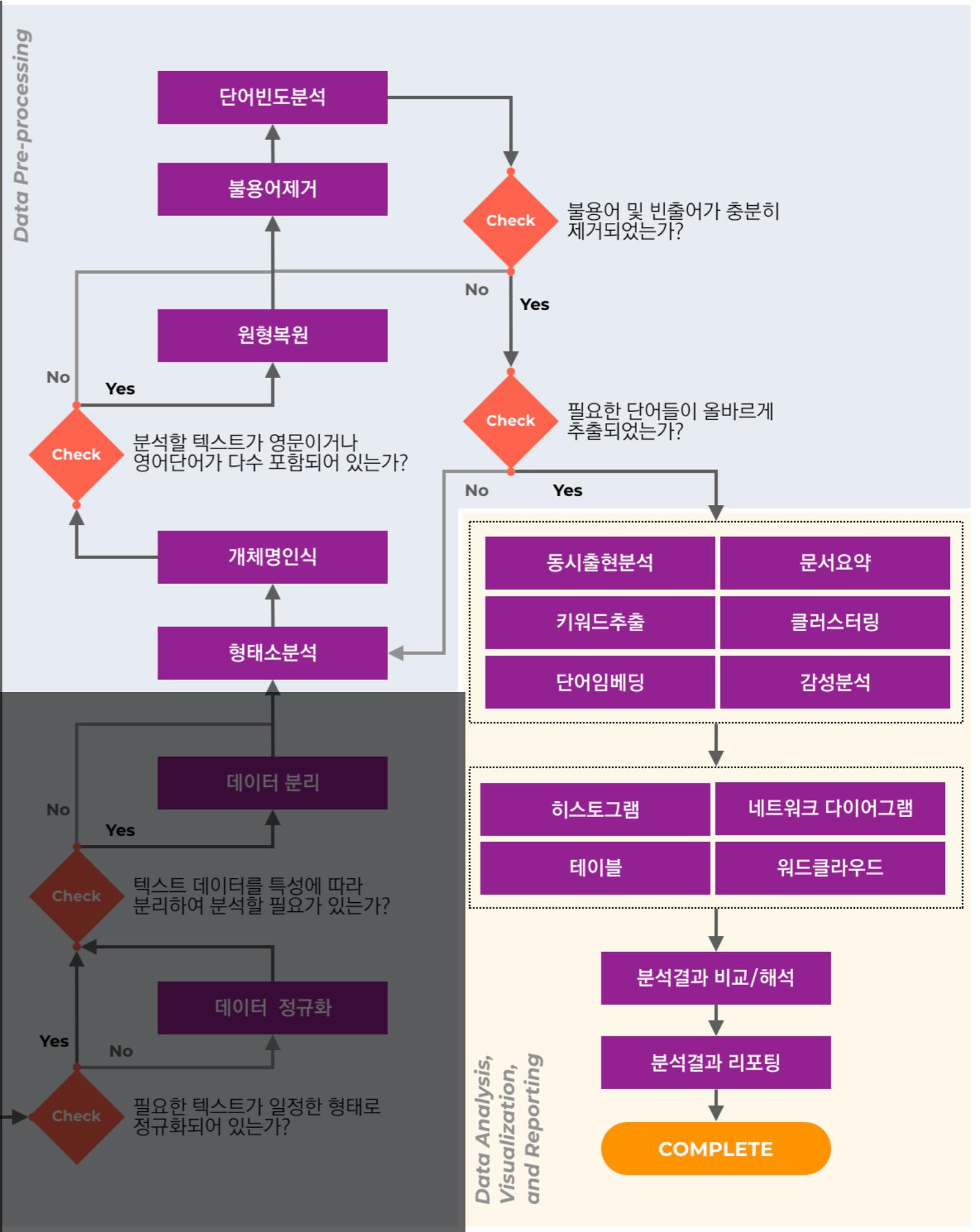
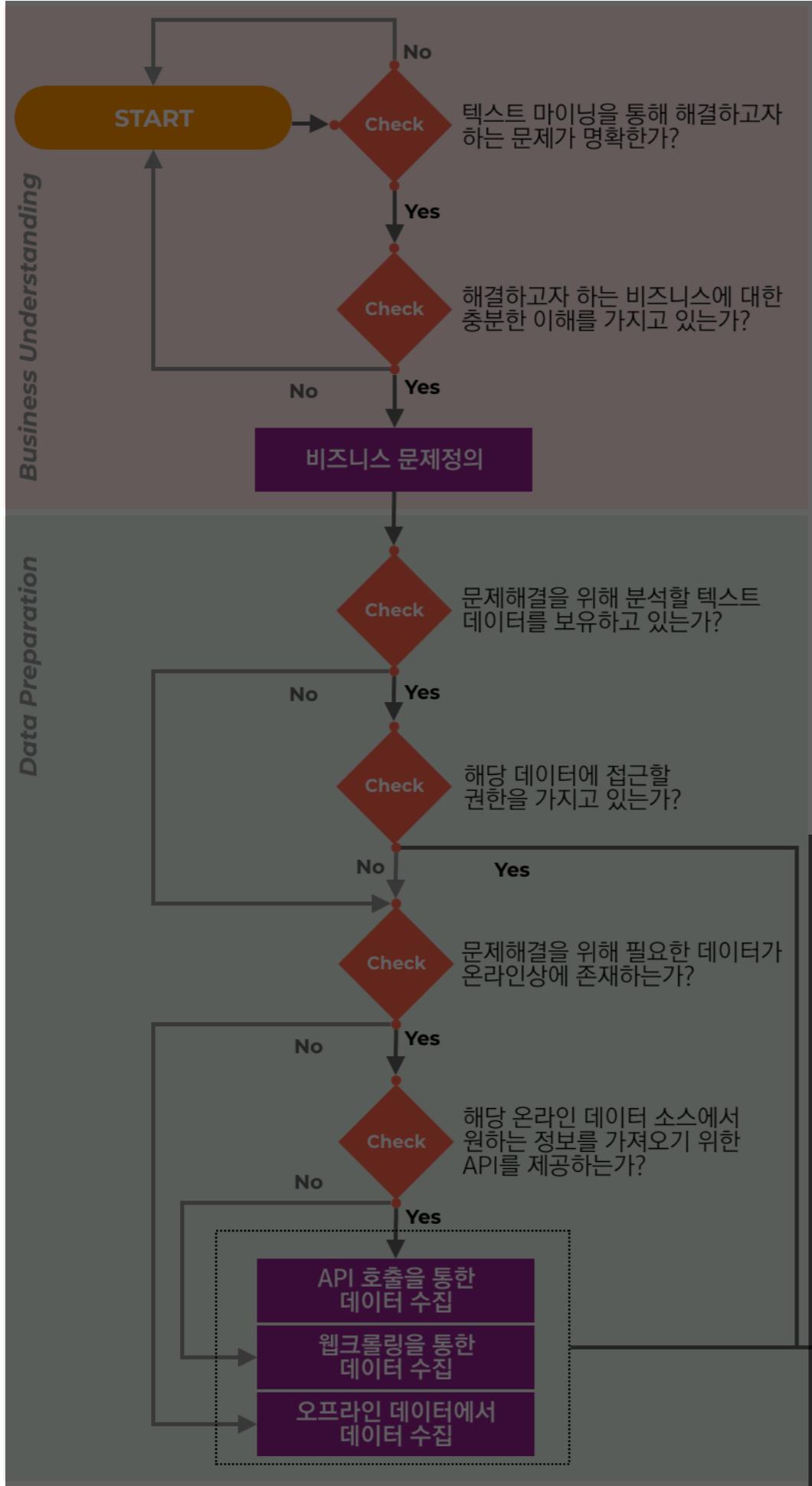


# **TEXT MINING for PRACTICE**

Python을 활용한 비정형 데이터 분석 - WEEK 05  
**형태소분석 & 개체명인식**

연세대학교 | 서중원

**얼마나 했나?**



# 자연어 (Natural Language)

## 프로그래밍 언어

- ▶ C, C++, Python, Java 등등 정해진 규칙을 따르는 인공언어

## 자연어

- ▶ 정해진 규칙만을 따르지 않고 일상적으로 쓰이는 언어
- ▶ 어떤 정돈된 완벽한 문법이나 형식이 없는 언어



안녕하세요

Hello

Привет

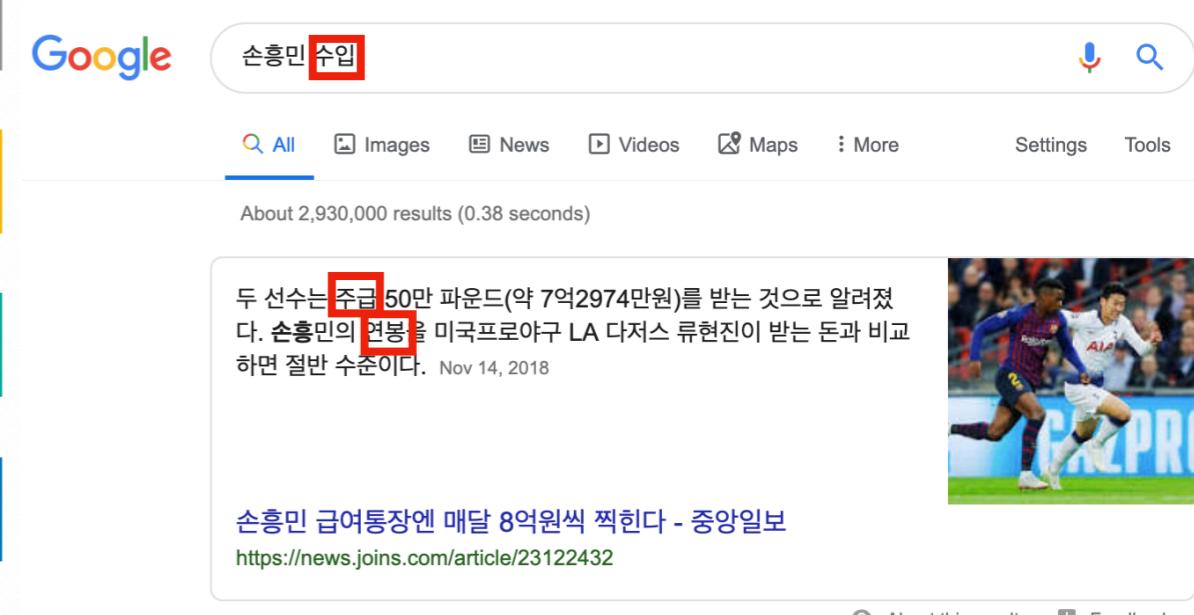
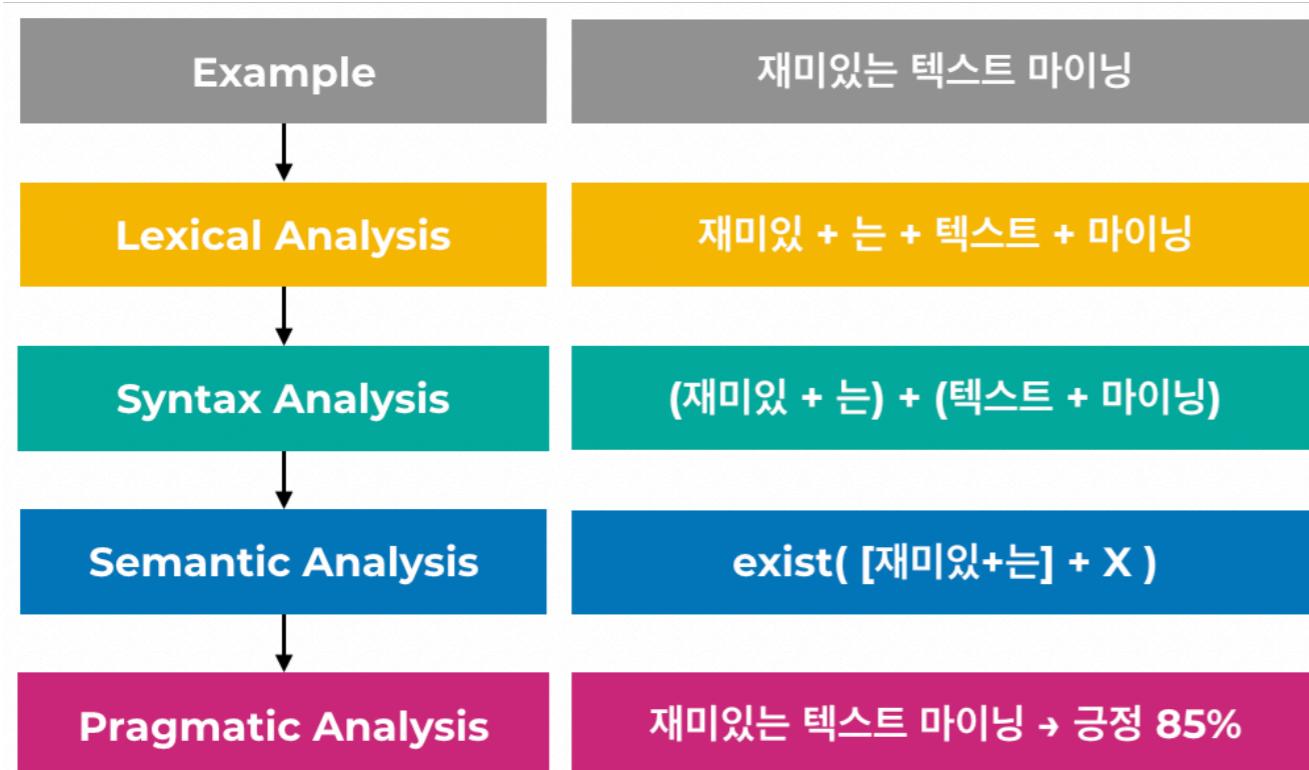
مرحبا

Olá

# 자연어의 분석 단계

## 자연어처리의 유형

- ▶ Lexical (=Morphology) : 단어의 유의미한 성분에 관한 연구
- ▶ Syntax : 단어간 구조적 관계에 관한 연구
- ▶ Semantics : 문장/단어의 의미론적 연구 (예: 단순 키워드 뿐만 아니라 의도와 문맥까지 파악)
- ▶ Pragmatics : 언어를 사용하여 특정한 목표를 달성하기 위한 연구



[구글의 Semantic Search의 예]

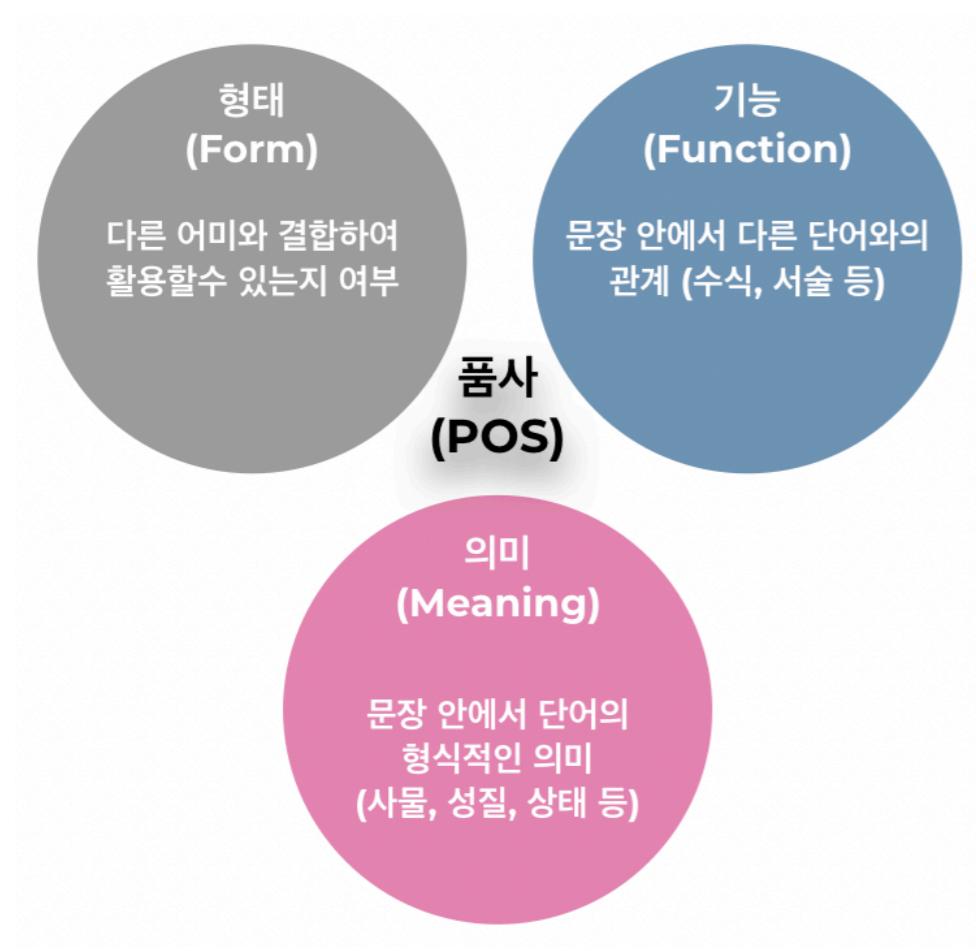
# 자연어 처리 단계 예

원문	나는 그 과자를 먹었다.	너는 그를 사랑하니?
형태소 분석	나(대명사) + 는(조사) + 그(대명사) + 과자(명사) + 먹-(동사)+-었(선어말어미)+-다(어말 어미)+ .(문장 부호)	너(대명사) + 는(조사) + 그(대명사) + 사랑(명사) + -하(동사)+니(어말 어미) + ? (문장부호)
구문 분석	(S (NP 나/Noun) 는/Josa (NP 그/Noun 사 과/Noun) 를/Josa (VP 먹었다/Verb)./ Punctuation)	(S (NP 너/Noun) 는/Josa (NP 그/Noun) 를/ Josa (NP 사랑/Noun) (VP 하니/Verb) ?/ Punctuation)
의미 분석	술부: 먹다 행위자: 나 대상: 그 과자	술부: 사랑하다 행위자: 너 대상: 그
담화 분석	단순 서술	의문형

# 한국어 품사구분

## 한국어의 5언 9품사

- ▶ 단어를 기능 (function), 의미 (meaning), 형태 (form)의 세 가지 기준에 의해 분류함



형태	기능	의미	태그
불변어	체언	명사	NNG
	체언	대명사	NNP
	수식언	수사	NR
	독립언	감탄사	XG
가변어	관계언	조사	조사
	수식언	관형사	관형사
	용언	부사	부사
	용언	동사	VV
		형용사	VA

\*Source : Daum 백과사전, 품사의 분류 기준, <http://igoindol.net/siteagent/100.daum.net/encyclopedia/view/24XXXXX49949/>.

\*\*Source : for textmining, 한국어 품사 분류와 분포(distribution), 2017.4.21., <https://ratsgo.github.io/korean%20linguistics/2017/04/21/wordclass/>.

# 형태소 분석 (Part of Speech Tagging)

## 교착어, 굴절어, 그리고. 고립어

- ▶ 교착어 (agglutinative) : 어근에 접사가 결합되어 각 단어의 기능을 나타내는 언어 (한국어, 일본어, 몽골어, …)
- ▶ 굴절어 (inflectional) : 단어 자체의 형태변화로 그 단어의 문법성을 나타내는 언어 (라틴어, 독일어, 러시아어, …)
- ▶ 고립어 (isolating) : 단어의 형태변화 없이 문법적 관계는 어순에 의해 정해지는 언어 (**영어**, 중국어, …)

나는 그를 사랑했다 = 그를 나는 사랑했다

**Geschrieben-Schreiben-Schriftsteller**  
**(written)-(to write)-(writer)**

**Tom loves June != June loves Tom**

## Korean verb '하다' Conjugated

### regular verb

Form	Conjugation
base	하 ha
base2	하 ha
base3	하 ha
declarative present informal low	해 hae
declarative present informal high	해요 hae-yo
declarative present formal low	한다 han-da
declarative present formal high	합니다 hab-ni-da
past base	했 haess
declarative past informal low	했어 haess-eo
declarative past informal high	했어요 haess-eo-yo
declarative past formal low	했다 haess-da
declarative past formal high	했습니다 haess-seub-ni-da
future base	할 hal
declarative future informal low	할 거야 hal geo-ya
declarative future informal high	할 거예요 hal geo-ye-yo
declarative future formal low	할 거다 hal geo-da
declarative future formal high	할 겁니다 hal geob-ni-da
declarative future conditional informal low	하겠어 ha-gess-eo
declarative future conditional informal high	하겠어요 ha-gess-eo-yo
declarative future conditional formal low	하겠다 ha-gess-da
declarative future conditional formal high	하겠습니까 ha-gess-seub-ni-da
inquisitive present informal low	해? hae?
inquisitive present informal high	해요? hae-yo?
inquisitive present formal low	하니? ha-ni?
inquisitive present formal high	합니까? hab-ni-gga?
inquisitive past informal low	했어? haess-eo?
inquisitive past informal high	했어요? haess-eo-yo?
inquisitive past formal low	했니? haess-ni?
inquisitive past formal high	했습니까? haess-seub-ni-gga?
imperative present informal low	해 hae
imperative present informal high	하세요 ha-se-yo
imperative present formal low	해라 hae-ra
imperative present formal high	하십시오 ha-sib-si-o
propositive present informal low	해 hae
propositive present informal high	해요 hae-yo
propositive present formal low	하자 ha-ja
propositive present formal high	합시다 hab-si-da
connective if	하면 ha-myeon
connective and	하고 ha-go
nominal ing	함 ham

# 형태소 분석 (Part of Speech Tagging)

## 형태소 분석이란?

- ▶ 문장을 형태소 단위로 구분하고 품사를 구별하여 태깅하고 용언의 다양한 활용으로 인한 형태소 탈락현상을 복원하는 과정
- ▶ 분석기마다 형태소 구분 방식이 다르기 때문에 데이터에 맞는 분석기를 선택해야함
- ▶ 모든 언어의 자연어 처리과정 중 가장 중요하고 기초적인 역할 수행
- ▶ 형태소 분석의 활용
  - 언어학적 측면 : 특정 언어현상의 생성과정을 설명하는 데 용이하게 쓰일 수 있음
  - 전산학적 측면 : 정보검색이나 자연어 처리 자동 처리시스템의 구문 분석의 전 단계 등의 용도로 쓰일 수 있음

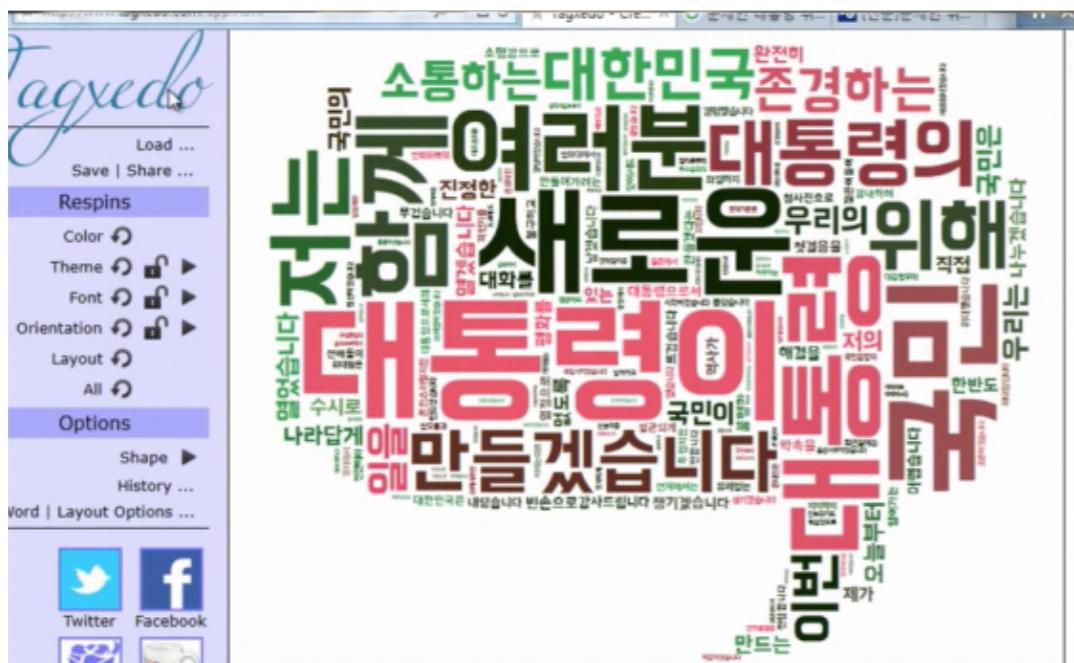
구분	내용
원문	<ul style="list-style-type: none"><li>• 여러분 안녕하세요. 재미있는 텍스트 마이닝 수업입니다.</li></ul>
형태소 분석	<ul style="list-style-type: none"><li>• 여러분/NP + 안녕/NNG + 하세요/EF + ./SF</li><li>• 재미있/VA + 는/ETM + 텍스트/NNG + 마이닝/NNG + 수업/NNG + 입니다/EF + ./SF</li></ul>

# 형태소 분석 (Part of Speech Tagging)

대분류	세종 품사 태그		KKMA 단일 태그 V 1.0					
	태그	설명	묶음 1	묶음 2	태그	설명	활용태그	저장사전
체언	NNG	일반 명사	N	NN	NNG	보통 명사	NNA	noun.dic
	NNP	고유 명사			NNP	고유 명사		
	NNB	의존 명사			NNB	일반 의존 명사		
	NNM	단위 의존 명사			NNM	단위 의존 명사		
	NR	수사		NR	NR	수사	NR	
용언	NP	대명사	V	NP	NP	대명사	NP	simple.dic
	VV	동사		VV	VV	동사	VV	
	VA	형용사		VA	VA	형용사	VA	
	VX	보조 용언		VX	VXV	보조 동사	VX	
	VCP	긍정 지정사		VC	VCP	긍정 지정사, 서술격 조사 '이다'	VCP	
	VCN	부정 지정사		VC	VCN	부정 지정사, 형용사 '아니다'	VCN	
	MM	관형사	M	MD	MDT	일반 관형사	MD	simple.dic
부사	MAG	일반 부사		MDN	MDN	수 관형사	MD	
	MAJ	접속 부사		MA	MAG	일반 부사	MAG	
감탄사	IC	감탄사	I	IC	IC	감탄사	IC	
	JKS	주격 조사	J	JK	JKS	주격 조사	JKS	raw.dic
조사	JKC	보격 조사		JKC	JKC	보격 조사	JKC	
	JKG	관형격 조사		JKG	JKG	관형격 조사	JKG	
	JKO	목적격 조사		JKO	JKO	목적격 조사	JKO	
	JKB	부사격 조사		JKM	JKM	부사격 조사	JKM	
	JKV	호격 조사		JKI	JKI	호격 조사	JKI	
	JKQ	인용격 조사		JKQ	JKQ	인용격 조사	JKQ	
	JX	보조사		JX	JX	보조사	JX	
	JC	접속 조사		JC	JC	접속 조사	JC	
선어말 어미	EP	선어말 어미	EP	EP	EPH	존칭 선어말 어미	EP	raw.dic
	EPT	시제 선어말 어미		EPT	EPT	시제 선어말 어미		
	EPP	공손 선어말 어미		EPP	EPP	공손 선어말 어미		

대분류	세종 품사 태그		KKMA 단일 태그 V 1.0					
	태그	설명	묶음 1	묶음 2	태그	설명	활용태그	저장사전
어말 어미	EF	종결 어미	E	EF	EFN	평서형 종결 어미	EF	simple.dic
	EFQ	의문형 종결 어미			EFO	명령형 종결 어미		
	EFA	청유형 종결 어미			EFI	감탄형 종결 어미		
	EFR	존칭형 종결 어미			ECE	대등 연결 어미		
	ECD	의존적 연결 어미			ECS	보조적 연결 어미		
	ETN	명사형 전성 어미			ETM	관형형 전성 어미	ETN	
접두사	XPN	체언 접두사	XP	XP	XPN	체언 접두사	XPN	simple.dic
	XPV	용언 접두사			XPV	용언 접두사	XPV	
접미사	XSN	명사 파생 접미사	XS	XS	XSN	명사 파생 접미사	XSN	
	XSV	동사 파생 접미사			XSV	동사 파생 접미사	XSV	
	XSA	형용사 파생 접미사			XSA	형용사 파생 접미사	XSA	
	XSM	부사 파생 접미사			XSM	부사 파생 접미사	XSM	
	XSO	자타 접미사			XSO	자타 접미사	XSO	
어근	XR	어근	XR	XR	XR	어근	XR	
	XR	어근			XR	어근	XR	
부호	SF	마침표물음표, 느낌표	S	S	SF	마침표물음표, 느낌표	SF	Symbol class
	SP	쉼표, 가운데점, 클론, 빛금			SP	쉼표, 가운데점, 클론, 빛금	SP	
	SS	따옴표, 괄호표, 줄표			SS	따옴표, 괄호표, 줄표	SS	
	SE	줄임표			SE	줄임표	SE	
	SO	불임표(줄침, 숨김, 빠짐)			SO	불임표(줄침, 숨김, 빠짐)	SO	
	SW	기타기호 (논리수학기호, 화폐기호)			SW	기타기호 (논리수학기호, 화폐기호)	SW	
분석 불능	NF	명사추정범주	U	U	UN	명사추정범주	NNA	N/A
	NV	용언추정범주			UV	용언추정범주	N/A	
	NA	분석불능범주			UE	분석불능범주	N/A	
한글 이외	SL	외국어	O	O	OL	외국어	NNA	N/A
	SH	한자			OH	한자	NNA	
	SN	숫자			ON	숫자	NR	

# 형태소 분석 (Part of Speech Tagging)



\* Source : 이정훈, 텍스트의 시각화: 단어 구름 (태그 클라우드), 2016.12.29., <http://visualloft.kr/tag-cloud/>.

\*\* Source : NÉSTOR CORREA, Cómo implementar el Big Data en tu empresa, 2017., <http://bluelight.tistory.com/298/>.

\*\*\* Source : 몬데이터, [mondata] 남북정상회담 판문점 선언 Text 키워드 분석, 2018.4.28., <https://www.youtube.com/watch?v=ba4EMdzSK-A>.

# Python 한국어 형태소 분석기

## ① 꼬꼬마 형태소 분석기: Kkma

- ▶ 서울대학교 IDS (Intelligent Data Systems) 연구실에서 자연어 처리를 위한 모듈구축과제로 개발한 형태소 분석기
- ▶ Java 언어를 기반으로 하며, Python-Java 연동을 통해 Python에서 사용 가능함
- ▶ 동적 프로그래밍 (Dynamic Programming) 방식으로 가능한 모든 형태소 후보를 모두 찾아 가장 적합한 형태소를 판단함 → 매우느림

### #Python 형태소 분석 예시

```
from konlpy.tag import Kkma
```

```
text="그래도 쏘니는 팀빨이네 이 소린 안 들음. 쏘니 잇었기에 토크님이 성과 낼 수 있기도 하는 거."
```

```
kkma = Kkma()
```

```
pos_result = kkma.pos(text)
```

**Result :** [ ('그리하', 'VV'), ('여도', 'ECD'), ('쏘', 'VV'), ('ㄴ', 'ECD'), ('는', 'JX'), ('팀', 'NNG'), ('빨', 'NNB'), ('이', 'VCP'), ('네', 'EFN'), ('이', 'MDT'), ('소', 'NNG'), ('린', 'UN'), ('안', 'MAG'), ('듣', 'VV'), ('음', 'ETN'), ('.', 'SF'), ('쓸', 'VV'), ('ㄴ', 'ECD'), ('잇', 'VV'), ('었', 'EPT'), ('기애', 'ECD'), ('토크', 'NNG'), ('님', 'NNB'), ('이', 'JKS'), ('성과', 'NNG'), ('내', 'VV'), ('ㄹ', 'ETD'), ('수', 'NNB'), ('있', 'VV'), ('기', 'ETN'), ('도', 'JX'), ('하', 'VV'), ('는', 'ETD'), ('거', 'NNB'), ('.', 'SF') ]

# Python 한국어 형태소 분석기

## ② 한나눔 형태소 분석기: Hannanum

- ▶ KAIST Semantic Web Research Center (SWRC)에서 개발한 형태소 분석기
- ▶ 자동 띄어쓰기 모듈을 제공해 형태소 분석 결과를 활용하여 한글 문장에 대한 자동 띄어쓰기 수행 가능
- ▶ 사전 기반의 맞춤법 교정 모듈로 형태소 분석 결과를 활용하여 한글 단어에 대한 맞춤법 교정 수행 가능

#Python 형태소 분석 예시

```
from konlpy.tag import Hannanum
text="그래도 쏘니는 팀빨이네 이 소린 안 들음. 쏘니 잇었기에 토크넘이 성과 낼 수 있기도 하는 거."
hannanum = Hannanum()
pos_result = hannanum.pos(text)
```

**Result :** [('그래도', 'M'), ('쏘', 'P'), ('니는', 'E'), ('팀빨이네', 'N'), ('이', 'M'), ('소', 'N'), ('이', 'J'), ('리', 'E'), ('알', 'P'), ('ㄴ', 'E'), ('들', 'P'), ('ㅁ', 'E'), ('.', 'S'), ('쏘', 'P'), ('ㄴ', 'E'), ('잇', 'P'), ('었', 'E'), ('에', 'J'), ('토크넘', 'N'), ('이', 'J'), ('성', 'N'), ('과', 'J'), ('내', 'P'), ('ㄹ', 'E'), ('수', 'N'), ('있', 'P'), ('기', 'E'), ('도', 'J'), ('하', 'P'), ('는', 'E'), ('거', 'I'), ('.', 'S')]

# Python 한국어 형태소 분석기

## ③ 코모란 형태소 분석기: Komoran

- ▶ 서울대학교 IDS (Intelligent Data Systems) 연구실에서 자연어 처리를 위한 모듈구축과제로 제작한 형태소 분석기
- ▶ Shineware에서 개발된 한국어 형태소 분석기로서 Java Library 형태(jar)로 제공됨
- ▶ 타 형태소 분석기와 달리 여러 어절을 하나의 품사로 분석 가능함으로써 공백이 포함된 고유명사(영화 제목, 음식점 명 등)를 정확하게 분석

### #Python 형태소 분석 예시

```
from konlpy.tag import Komoran
text="그래도 쏘니는 팀빨이네 이 소린 안 들음. 쏘니 잇었기에 토크넘이 성과 낼 수 있기도 하는 거."
komoran = Komoran()
pos_result = komoran.pos(text)
```

**Result :** [('그래도', 'MAJ'), ('쏘', 'VV'), ('ㄴ', 'EC'), ('는', 'JX'), ('팀', 'NNG'), ('빨', 'VV'), ('ㄹ', 'ETM'), ('이', 'NNP'), ('네', 'XSN'), ('이', 'MM'), ('소리', 'NNG'), ('ㄴ', 'JX'), ('안', 'MAG'), ('들', 'VV'), ('음', 'ETN'), ('.', 'SF'), ('쏘', 'VV'), ('ㄴ', 'EC'), ('잇', 'VV'), ('었', 'EP'), ('기', 'ETN'), ('에', 'JKB'), ('토크넘', 'NNP'), ('이', 'JKS'), ('성과', 'NNG'), ('내', 'VV'), ('ㄹ', 'ETM'), ('수', 'NNB'), ('있', 'VV'), ('기', 'ETN'), ('도', 'JX'), ('하', 'VV'), ('는', 'ETM'), ('거', 'NNB'), ('.', 'SF')]

# Python 한국어 형태소 분석기

## ④ 은전한닢 형태소 분석기: Mecab

- ▶ 검색에서 쓸만한 오픈소스 한국어 형태소 분석기를 목적으로 개발된 한국어 형태소 분석기
- ▶ 오픈소스 검색엔진 Elasticsearch에 적용되어 활용되고 있음
- ▶ 사용자 사전 등록기능을 제공하여 다양한 도메인에서 생성되는 단어들을 인식할 수 있도록 도와줌

### #Python 형태소 분석 예시

```
from konlpy.tag import Mecab
```

```
text="그래도 쏘니는 팀빨이네 이 소린 안 들음. 쏘니 잇었기에 토크넘이 성과 낼 수 있기도 하는 거."
```

```
mecab = Mecab()
```

```
pos_result = mecab.pos(text)
```

```
Result : [('그래도', 'MAJ'), ('쏘', 'VV'), ('ㄴ', 'EC'), ('는', 'JX'), ('팀', 'NNG'), ('빨', 'VV'), ('이', 'EP'), ('네', 'EF'), ('이', 'MM'), ('소린', 'NNG+JX'), ('안', 'MAG'), ('들', 'VV'), ('음', 'ETN'), ('.', 'SF'), ('쏘', 'VV'), ('ㄴ', 'EC'), ('잇', 'VX'), ('었', 'EP'), ('기', 'ETN'), ('에', 'JKB'), ('토크넘', 'NNP'), ('이', 'JKS'), ('성과', 'NNG'), ('낼', 'VV+ETM'), ('수', 'NNB'), ('있', 'VV'), ('기', 'ETN'), ('도', 'JX'), ('하', 'VV'), ('는', 'ETM'), ('거', 'NNB'), ('.', 'SF')]
```

# Python 한국어 형태소 분석기

## ⑤ 카이 형태소 분석기: Khaiii (Kakao Hangul Analyzer III)

- ▶ 카카오에서 DHA2 (Daumkakao Hangul Analyzer 2)를 계승하여 개발하고 2018년 공개된 두 번째 버전의 형태소분석기
- ▶ 속도를 매우 중요시하여 신경망 알고리즘들 중에서 Convolutional Neural Network (CNN)을 사용하여 개발됨
- ▶ 사용자 사전 등록기능을 제공하여 다양한 도메인에서 생성되는 단어들을 인식할 수 있도록 도와줌

#Python 형태소 분석 예시

```
from khaiii import KhaiiiApi
text="그래도 쏘니는 팀빨이네 이 소린 안 들음. 쏘니 잇었기에 토크넘이 성과 낼 수 있기도 하는 거."
api = KhaiiiApi()
pos_result = api.analyze(text)
```

**Result :** [ ('그러', 'VV'), ('어도', 'EC'), ('쏘니', 'VV'), ('는', 'ETM'), ('팀빨', 'NNG'), ('이', 'VCP'),  
('네', 'XSN'), ('이', 'MM'), ('소리', 'VV'), ('ㄴ', 'ETM'), ('안', 'MAG'), ('들', 'VV'), ('음', 'ETN'), ('.', 'SF'),  
('쏘', 'NNG'), ('니', 'MAG'), ('잇', 'VV'), ('었', 'EP'), ('기에', 'EC'), ('토크넘', 'NNG'), ('이', 'JKS'),  
('성', 'NNG'), ('과', 'JC'), ('내', 'VV'), ('ㄹ', 'ETM'), ('수', 'NNB'), ('있', 'VV'), ('기', 'ETN'), ('도', 'JX'),  
('하', 'VX'), ('는', 'ETM'), ('거', 'NNB'), ('.', 'SF') ]

# Python 한국어 형태소 분석기

## ⑥ 트위터 형태소 분석기: Twitter (Okt)

- ▶ 트위터에서 개발한 한국어 형태소 분석기
- ▶ SNS에서 발생하는 언어에서 자주 발생하는 인물명, 신조어 등을 잘 인식하는 편이며, 속도가 빠르지만 형태소 분석 품질은 상대적으로 낮음

#Python 형태소 분석 예시

```
from konlpy.tag import Okt
text="그래도 쏘니는 팀빨이네 이 소린 안 들음. 쏘니 잇었기에 토크넘이 성과 낼 수 있기도 하는 거."
okt = Okt()
pos_result = okt.pos(text)
```

**Result :** [('그래도', 'Adverb'), ('쏘니는', 'Verb'), ('팀빨', 'Noun'), ('이네', 'Josa'), ('이', 'Noun'), ('소린', 'Noun'), ('안', 'Noun'), ('들음', 'Verb'), ('.', 'Punctuation'), ('쏘니', 'Verb'), ('잇었기에', 'Verb'), ('토크넘', 'Noun'), ('이', 'Josa'), ('성과', 'Noun'), ('낼', 'Noun'), ('수', 'Noun'), ('있기도', 'Adjective'), ('하는', 'Verb'), ('거', 'Noun'), ('.', 'Punctuation')]

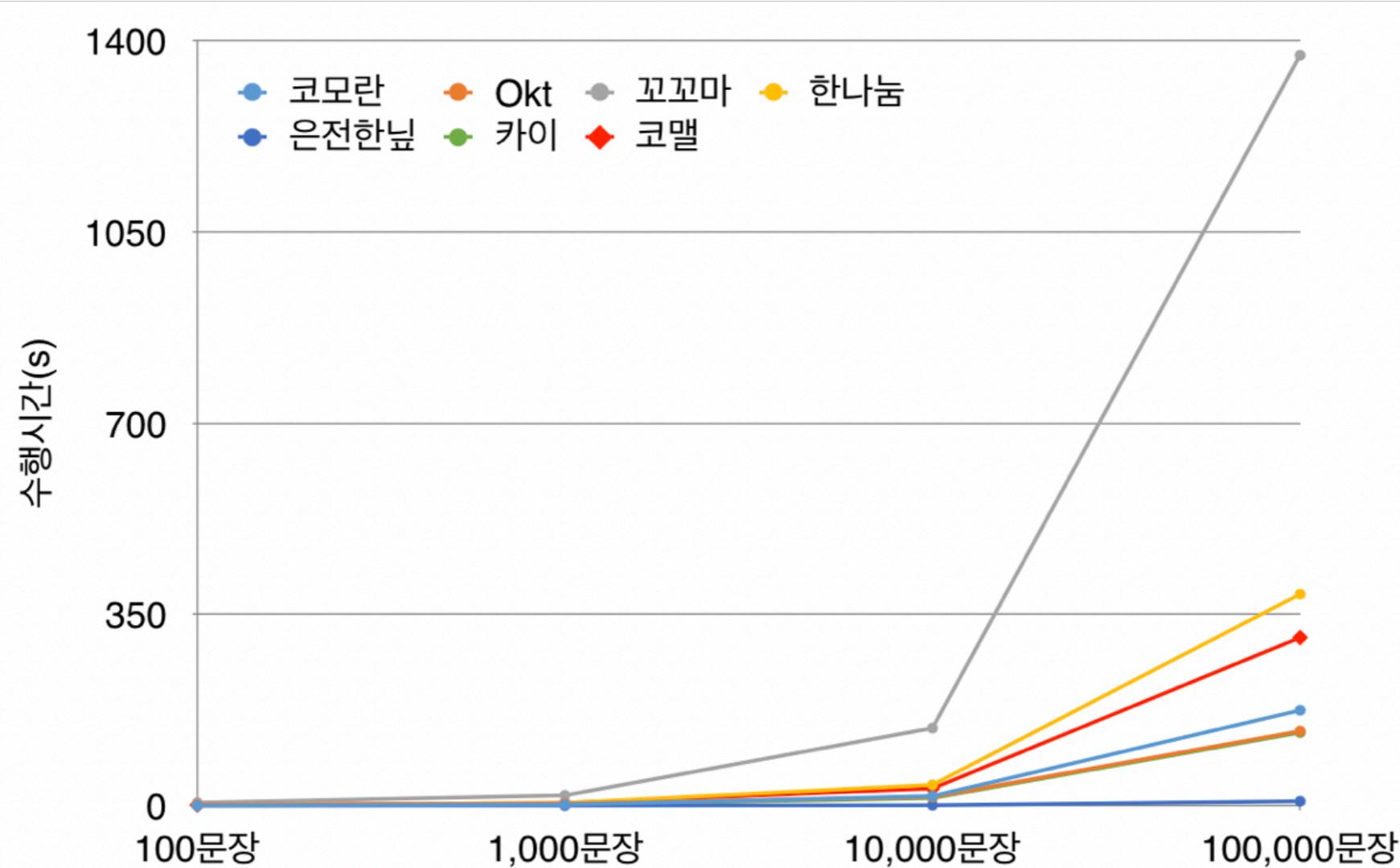
# 형태소 분석기 비교

## 형태소 분석 결과 비교

구분	형태소 분석 결과
원문	그래도 <b>우리</b> / <b>홍</b> / <b>NNG</b> <b>은</b> / <b>JX</b> <b>팀</b> / <b>NNG</b> <b>빨</b> / <b>VV</b> <b>이</b> / <b>ETM</b> <b>있</b> / <b>NNP</b> <b>었</b> / <b>XSN</b> <b>기</b> / <b>MM</b> 소리/ <b>NNG</b> 안/ <b>MAG</b> 들/ <b>VV</b> 음/ <b>ETN</b> ./ <b>SF</b> <b>쓰</b> / <b>VV</b> <b>니</b> / <b>EC</b> <b>있</b> / <b>VX</b> <b>었</b> / <b>EP</b> <b>기</b> / <b>ETN</b> <b>에</b> / <b>JKB</b> <b>토트</b> / <b>NNP</b> <b>이</b> / <b>JKS</b> 성과/ <b>NNG</b> 내/ <b>VV</b> <b>이</b> / <b>ETM</b> 수/ <b>NNB</b> <b>있</b> / <b>VV</b> <b>기</b> / <b>ETN</b> <b>도</b> / <b>JX</b> <b>하</b> / <b>VV</b> <b>는</b> / <b>ETM</b> <b>거</b> / <b>NNB</b> ./ <b>SF</b>
코모란	그래도/MAJ 우리/NP 홍/NNG 은/JX 팀/NNG 빨/VV 이/ETM 이/MM 소리/NNG 안/MAG 들/VV 음/ETN ./SF 쓰/VV 니/EC 있/VX 었/EP 기/ETN 에/JKB 토트/NNP 이/JKS 성과/NNG 내/VV 이/ETM 수/NNB 있/VV 기/ETN 도/JX 하/VV 는/ETM 거/NNB ./SF
Okt	그래도/Adverb 우리/Noun 홍/Noun 은/Josa 팀빨/Noun 이네/Josa 이/Noun 소린/Noun 안/Noun 들음/Verb ./Punctuation 쓰니/Verb 있었기에/Adjective 토트넘/Noun 이/Josa 성과/Noun 낼/Noun 수/Noun 있기도/Adjective 하는/Verb 거/Noun ./Punctuation
꼬꼬마	그리하/VV 여도/ECD 우리/NP 홍/NNG 은/JX 팀/NNG 빨/NNB 이/VCP 네/EFN 이/MDT 소/NNG 린/UN 안/MAG 들/VV 음/ETN ./SF 쏠/VV 니/ECD 있/VXV 었/EPT 기에/ECD 토트/NNG 넘/NNB 이/JKS 성과/NNG 내/VV 이/ETD 수/NNB 있/VV 기/ETN 도/JX 하/VV 는/ETD 거/NNB ./SF
한나눔	그래도/M 우리홍/N 은/J 팀빨이네/N 이/M 소/N 이/J 리느/E 알/P 안/E 들/P 으/E ./S 쓰/P 니/E 있/P 었기/E 에/J 토트넘/N 이/J 성/N 과/J 내/P 이/E 수/N 있/P 기/E 도/J 하/P 는/E 거/I ./S
은전한닢	그래도/MAJ 우리/NP 홍/NNG 은/JX 팀/NNG 빨/VV 이/EP 네/EF 이/MM 소린/NNG+JX 안/MAG 들/VV 음/ETN ./SF 쓰/VV 니/EC 있/VX 었/EP 기/ETN 에/JKB 토트넘/NNP 이/JKS 성과/NNG 낼/VV+ETM 수/NNB 있/VV 기/ETN 도/JX 하/VV 는/ETM 거/NNB ./SF
카이	그러/VV 어도/EC 우리홍/NNP 은/JX 팀빨/NNG 이/VCP 네/XSN 이/MM 소리/VV 안/MAG 들/VV 음/ETN ./SF 쓰/VV 니/MAG 있/VV 었/EP 기에/EC 토트넘/NNG 이/JKS 성/NNG 과/JC 내/VV 이/ETM 수/NNB 있/VV 기/ETN 도/JX 하/VX 는/ETM 거/NNB ./SF

# 형태소 분석기 비교

## 수행시간 비교 (Time Analysis)



# 개체명 인식

(Named Entity Recognition)

문장에서 하나의 개체로써 인식되어야하는 단어를 구별하는 과정

- ▶ 데이터에서 개체명을 구별하고 태깅함(지명, 사명, 인물명, 약자, 기관명 등)
- ▶ 사전 기반의 개체명 인식에서 개체명은 매일 새롭게 생겨나고 변형되므로, 개체명 사전을 유지하는 것이 매우 중요함
- ▶ 분석의 목적에 따라서 머신러닝 기반의 개체명 인식을 사용할 수 있으나 새로 생겨나거나 변형되는 단어에 취약함

#Python 형태소 분석 예시

```
from konlpy.tag import Kkma
```

```
text="호날두 한명이 주는 효과가 세리에 전체 인기도 영향을 미치다니.. 역시 개드립월클의 힘"
```

```
kkma = Kkma()
```

```
pos_result = kkma.pos(text)
```

**Result :** [('호', 'NNG'), ('날', 'NNG'), ('두', 'MDN'), ('한명', 'NNG'), ('이', 'JKS'), ('줄', 'VV'), ('는', 'ETD'), ('효과', 'NNG'), ('가', 'JKS'), ('세리', 'NNG'), ('에', 'JKM'), ('전체', 'NNG'), ('인기', 'NNG'), ('영향', 'NNG'), ('을', 'JKO'), ('미', 'NNG'), ('하', 'XSV'), ('지', 'ECD'), ('달', 'VXV'), ('니', 'ECD'), ('..', 'SW'), ('역시', 'MAG'), ('개드립월클', 'UN'), ('의', 'JKG'), ('힘', 'NNG')]

# 개체명 인식

(Named Entity Recognition)

문장에서 하나의 개체로써 인식되어야하는 단어를 구별하는 과정

- ▶ 데이터에서 개체명을 구별하고 태깅함(지명, 사명, 인물명, 약자, 기관명 등)
- ▶ 사전 기반의 개체명 인식에서 개체명은 매일 새롭게 생겨나고 변형되므로, 개체명 사전을 유지하는 것이 매우 중요함
- ▶ 분석의 목적에 따라서 머신러닝 기반의 개체명 인식을 사용할 수 있으나 새로 생겨나거나 변형되는 단어에 취약함

#Eucalyptus 형태소 분석 예시

```
from Eucalyptus.NerTagger import NerTagger  
input_file, output_file = "output_pos.txt", "output_ner.txt"  
ner_tagger = euc.NerTagger(input_file, output_file)  
ner_tagger.tagging()
```

Result (output\_ner.txt) : [(호날두, NNP, Person), (한명, NNG), (이, JKS), (줄,VV), (는, ETD),  
(효과, NNG), (가, JKS), (세리에, NNP, Sports), (에, JKM), (전체, NNG), (인기도, NNG), (영향, NNG), (을,JKO), ... , (역시, MAG), (개드립월클, NNG, Neologism), (의, NNG), (힘, NNG)]

# 개체명 사전 (NER Corpus)

## [ 단순 개체명 사전 ]

구분	의학	인물	고유명사	블록체인
1	불량 식품	사나	서울플랜트엔지니어링	블록체인
2	진행 암	쭈위	서울플리머	블럭체인
3	전진 피판	정연	서울피브이시상사	비트코인
4	유해 효과	나연	서울피브이씨	이더리움
5	무력증	황민현	서울피비씨	알트코인
6	유산소 운동	강다니엘	서울피앤씨	추격매수
7	산소 호흡	옹성우	서울하이테크	풀매수
8	공기 삼킴증	전병진	서울학연구	총알
9	분무제	진상형	고광엔지니어링	운전수
10	에어로졸	서지석	서울합금	고점
11	분무 주입법	배현진	서울합판	저점
12	대기 요법	현빈	서울합판목재상사	장투
13	정동 장애	진세연	서울합판상사	단타
14	정감성	남지현	서울해체산업	떡상
15	정동성	주상욱	서울행정신문사	떡락
16	들신경	김태희	서울행정학회	횡보
17	협력 병원	허맹호	서울화성	손절
18	친화력	유아인	서울화인테크	익절
19	친화 크로마토그래피	이승기	서울화학	반등
20	무섬유소원 혈증	한예슬	고광훈	패닉셀

## [ 부가정보를 포함하는 개체명 사전 ]

구분	지역명	영문 지역명	구분
1	서울	Seoul	Metropolitan
2	종로	Jongno	district
3	중	Jung	district
4	용산	Yongsan	district
5	성동	Seongdong	district
6	광진	Gwangjin	district
7	동대문	Dongdaemun	district
8	중랑	Jungnang	district
9	성북	Seongbuk	district
10	강북	Gangbuk	district
11	도봉	Dobong	district
12	노원	Nowon	district
13	은평	Eunpyeong	district
14	서대문	Seodaemun	district
15	마포	Mapo	district
16	양천	Yangcheon	district
17	강서	Gangseo	district
18	구로	Guro	district
19	금천	Gumcheon	district
20	영등포	Yeongdeungpo	district

# 미등록 단어 추출

형태소 분석과 개체명 인식은 새로운 단어를 인식하기가 어려움

[가정의 달! 든든한 금융]KB손보, The간편한치매간병보험 출시

경증부터 중증까지 폭 넓은 보장

등록 2019-05-18 오전 10:06:05  
수정 2019-05-18 오전 10:06:05

가 가

구분	내용
원문	KB손해보험은 치매에 대해 경증부터 중증까지 폭넓게 보장하는 'The간편한치매간병 보험'을 판매 중이다. 이 상품은 경증치매, 중등도치매, 중증치매, 알츠하이머병, 파킨슨 병까지 치매와 관련된 질병들을 포괄적으로 보장하는 게 가장 큰 특징이다. ...
형태소 분석	kb/SL 손해/NNG 보험/NNG 은/JX 치매/NNG 에/JKB 대하/VV 아/EC 경증/ NNG 부터/JX 중증/NNG 까지/JX 폭넓/VA 게/EC 보장/NNG 하/XSV 는/ETM '/ SO the/SL 간편/XR 하/ XSA h/ETM 치매/NNG 간병/NNG 보험/NNG '/SO 을/JKO 판매/NNG 중/NNB 이/VCP 다/ EF ./SF 이/MM 상품/NNG 은/JX 경증/ NNG 치매/NNG ,/SP 중등/NNG 도/JX 치매/ NNG ,/SP 중증/NNG 치매/ NNG ,/SP 알츠하이머병/NNG ,/SP 파킨슨병/NNG 까지/JX 치매/NNG 와/JC 관련/NNG 되/XSV h/ETM 질병/NNG 들/XSN 을/JKO 포괄/NNG 적/XSN 으로/JKB 보장/NNG 하/XSV 는/ETM 것/NNB 이/JKS 가장/MAG 크/VA h/ETM 특징/NNG 이/VCP 다/EF ...

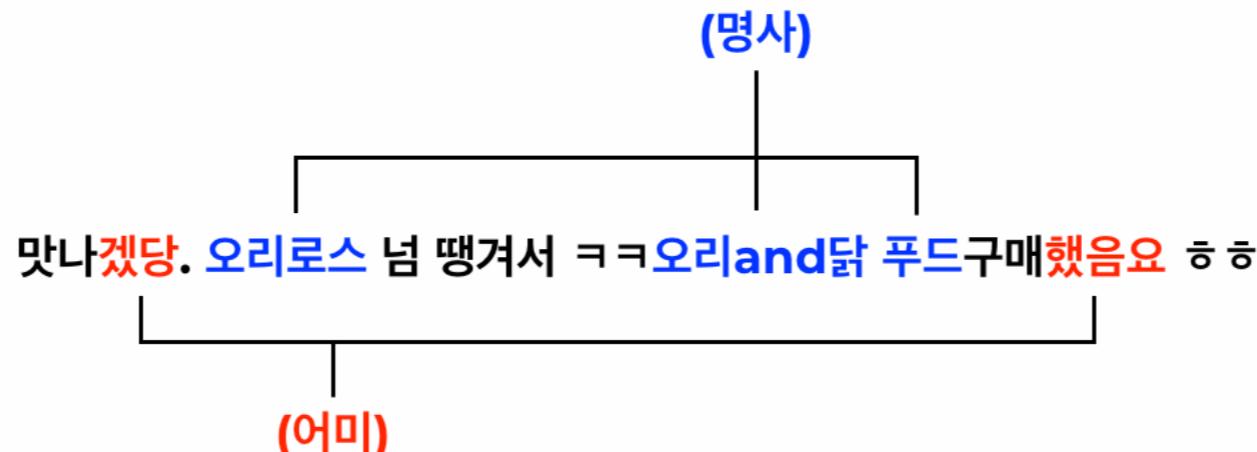
# 미등록 단어 추출

## 새로운 단어를 인식하기가 어려운 이유

- ▶ 형태소 사전 기반으로 형태소 분석을 수행하는 경우, 미등록단어를 알려진 형태소 단위로 분해함
- ▶ 특히 한국어는 한자어의 조합으로 구성된 단어들이 많기 때문에 작은 의미단위로 분해될 가능성이 큼
- ▶ 좋은 품질의 형태소 분석을 위해서는 새로운 단어들을 사전에 추가하는 과정이 반드시 필요함

## 신규 단어등록의 자동화 또는 반자동화

- ▶ 사용자 사전을 만들되, 효율적으로 구성하는 방법을 찾아야 텍스트 전처리 시간을 줄이고 전처리 결과의 질을 높일 수 있음
- ▶ 신규 단어의 대부분은 명사 또는 어미로 이루어지는 경우가 많음
  - 명사 : 새로운 개념을 표현하기 위해 생성됨
  - 어미 : 새로운 말투를 표현하기 위해 생성됨 (동사/형용사에 영향)



**E.O.D**