# TEXT MINING for PRACTICE

Python을 활용한 비정형 데이터 분석 - WEEK 05

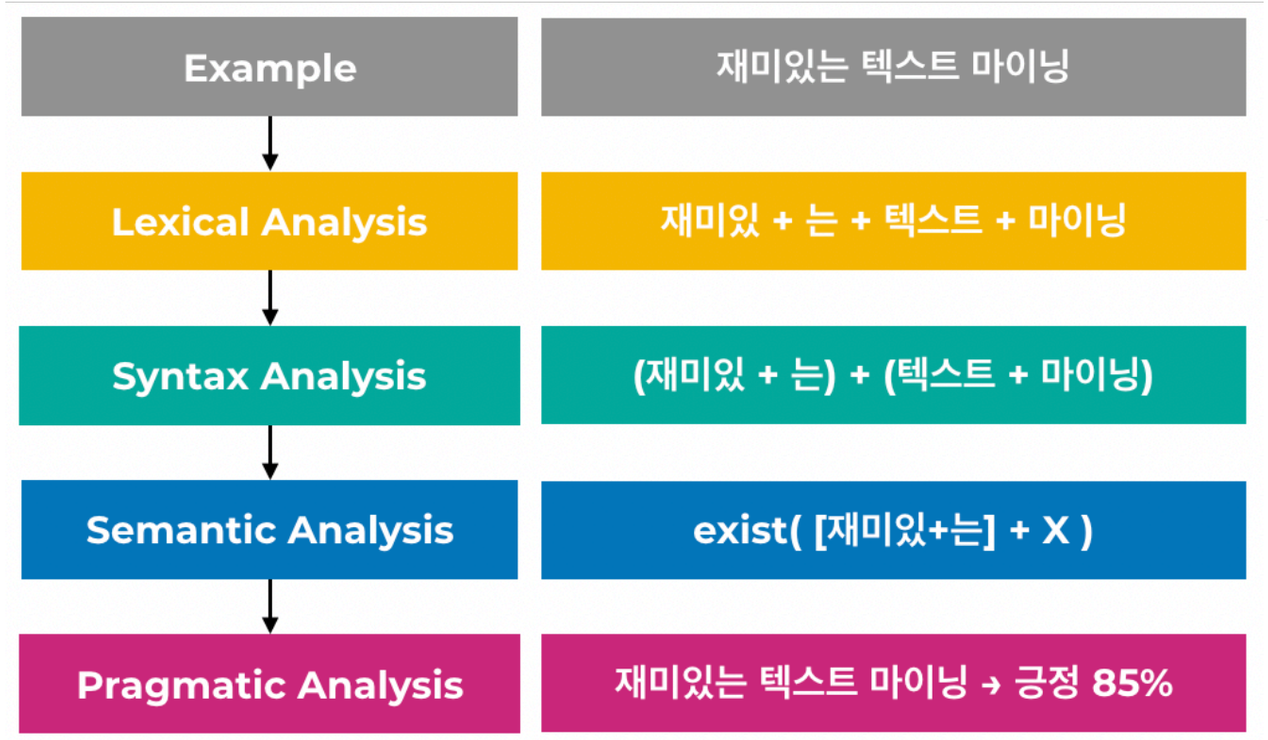
**형태소분석 & 개체명인식**

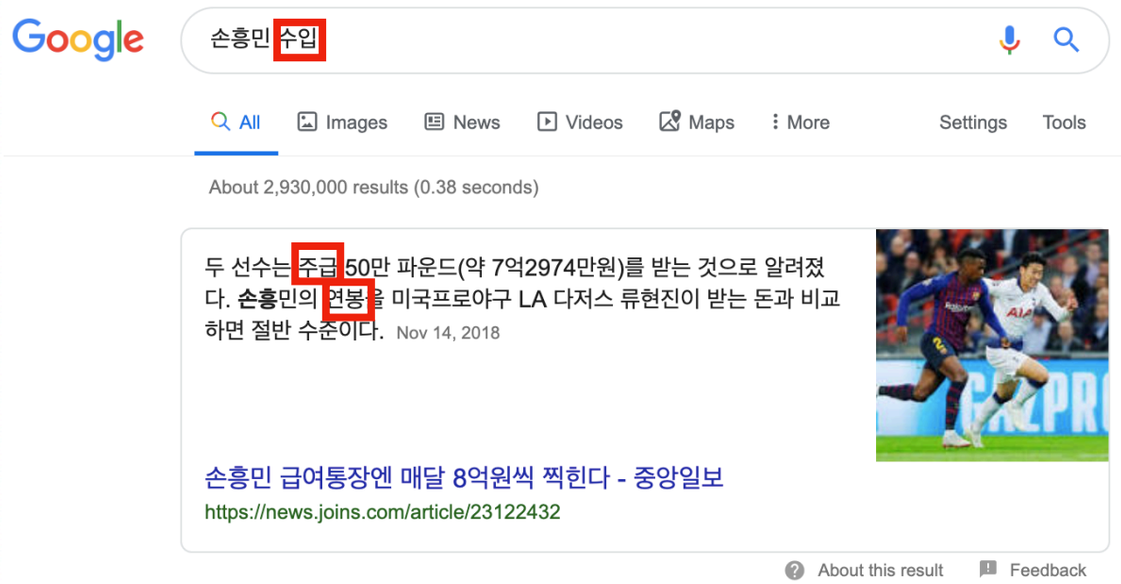
연세대학교 | 서중원

# 어휘분석 (Lexical Analysis)

## 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



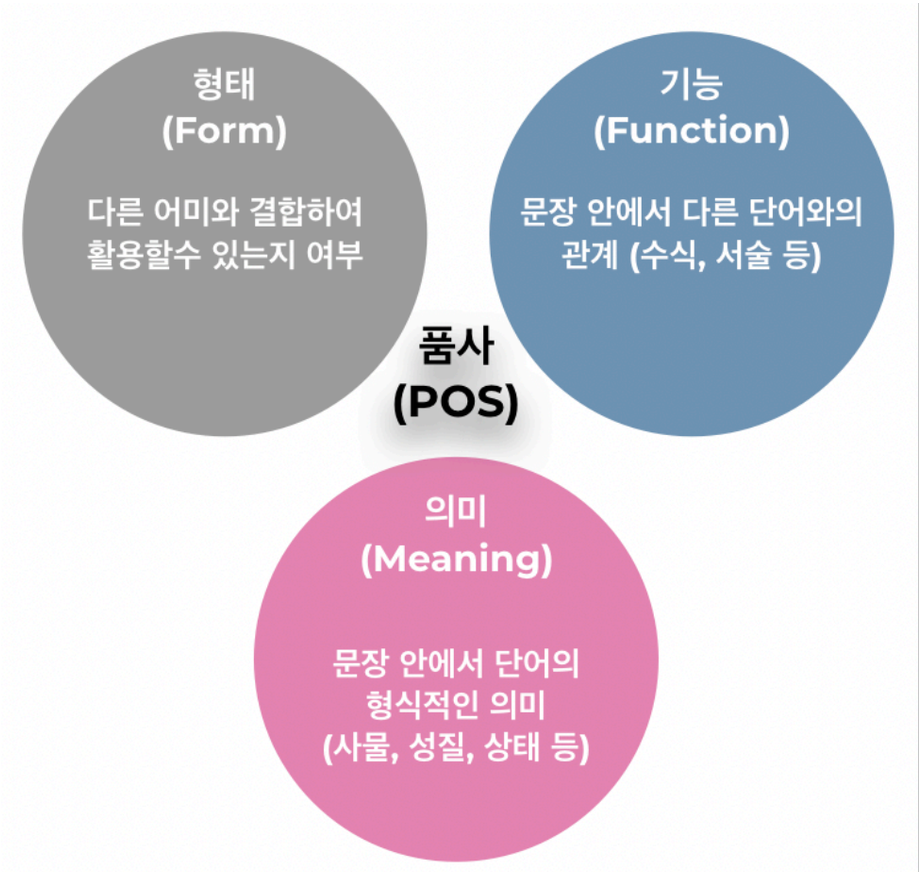

**[구글의 Semantic Search의 예]**

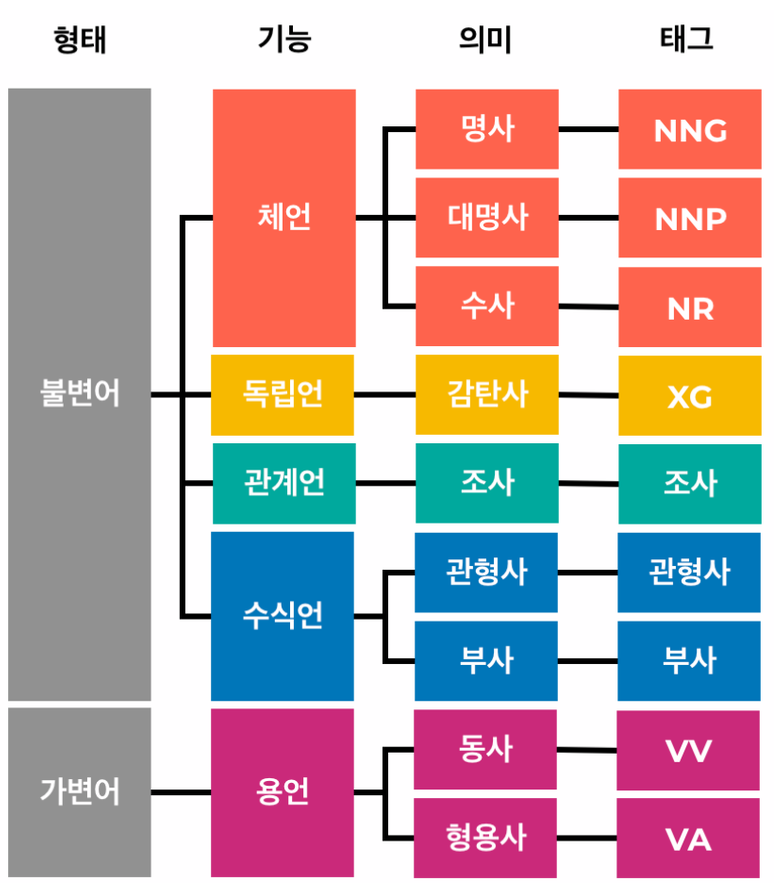
*Source : Javed Shaikh, Machine Learning, NLP: Text Classification using scikit-learn, python and NLTK, 2017.6.24., https://towardsdatascience.com/machine-learning-nlp-text-classification-using-scikit-learn-python-and-nltk-c52b92a7c73a/.

# 한국어 품사구분

## 한국어의 5언 9품사

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




*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, …)

## 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
- 모든언어의자연어처리과정중가장중요하고기초적인역할수행
- 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 | • 여러분 안녕하세요. 재미있는 텍스트 마이닝 수업입니다. |
| 형태소 분석 | • 여러분/NP + 안녕/NNG + 하세요/EF + ./SF<br>• 재미있/VA + 는/ETM + 텍스트/NNG + 마이닝/NNG + 수업/NNG + 입니다/EF + ./SF |

# 형태소 분석 (Part of Speech Tagging)

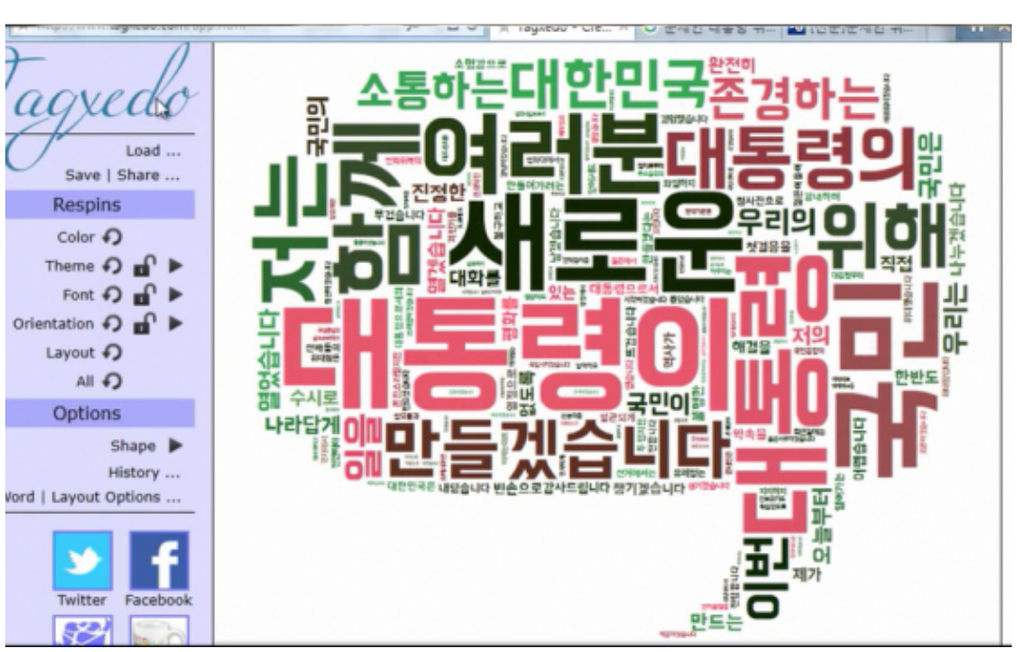
| 대분류 | 세종 품사 태그 | | KKMA 단일 태그 V 1.0 | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 태그 | 설명 | 묶음 1 | 묶음 2 | 태그 | 설명 | 확률태그 | 저장사전 |
| 체언 | NNG | 일반 명사 | N | NN | NNG | 보통 명사 | NNA | noun.dic |
| | NNP | 고유 명사 | | | NNP | 고유 명사 | | |
| | NNB | 의존 명사 | | | NNB | 일반 의존 명사 | NNB | simple.dic |
|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

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

* Source : 꼬꼬마 한국어 형태소 분석기, 한글 형태소 품사 (Part Of Speech, POS) 태그표, 2019., http://kkma.snu.ac.kr/documents/index.jsp?doc=postag/.

# 형태소 분석 (Part of Speech Tagging)









* Source : 이정훈,텍스트의 시각화: 단어 구름 (태그 클라우드), 2016.12.29., http://visualoft.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

## ❶ 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 토트넘이 성과 낼 수 있기도 하는 거."
kkma = Kkma()
pos_result = kkma.pos(text)

Result : [('그리하', 'VV'), ('여도', 'ECD'), ('쏘', 'VV'), ('니', 'ECD'), ('는', 'JX'), ('팀', 'NNG'), ('빨', 'NNB'), ('이', 'VCP'), ('네', 'EFN'), ('이', 'MDT'), ('소', 'NNG'), ('린', 'UN'), ('안', 'MAG'), ('듣', 'VV'), ('음', 'ETN'), ('.', 'SF'), ('쏠', 'VV'), ('니', 'ECD'), ('잇', 'VV'), ('었', 'EPT'), ('기에', 'ECD'), ('토트', 'NNG'), ('넘', '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

## ❷ 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 토트넘이 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'),
('리ㄴ', 'E'), ('알', 'P'), ('ㄴ', 'E'), ('들', 'P'), ('ㅁ', 'E'), ('.', 'S'), ('쏘', 'P'), ('니', 'E'), ('잇', 'P'), ('었
기', 'E'), ('에', 'J'), ('토트넘', '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

## ❸ 코모란 형태소 분석기: Komoran

- 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 Komoran
text="그래도 쏘니는 팀빨이네 이 소린 안 들음. 쏘니 잇었기에 토트넘이 성과 낼 수 있기도 하는 거."
komoran = Komoran()
pos_result = komoran.pos(text)

Result : [('그래도', 'MAJ'), ('쏘', 'VV'), ('니', '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'), ('니', 'EC'), ('잇', 'VV'), ('었', 'EP'), ('기', 'ETN'), ('에',
'JKB'), ('토트넘', '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

## ❹ 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 토트넘이 성과 낼 수 있기도 하는 거."
mecab = Mecab()
pos_result = mecab.pos(text)

Result : [('그래도', 'MAJ'), ('쏘', 'VV'), ('니', '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'), ('니', 'EC'), ('잇', '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')]
```

# Python 한국어 형태소 분석기

## ❺ 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 토트넘이 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'), ('토트넘', '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

## ❻ 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 토트넘이 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'), ('토트넘', 'Noun'), ('이', 'Josa'), ('성과', 'Noun'), ('낼', 'Noun'), ('수', 'Noun'), ('있기도',
'Adjective'), ('하는', 'Verb'), ('거', 'Noun'), ('.', 'Punctuation')]
```

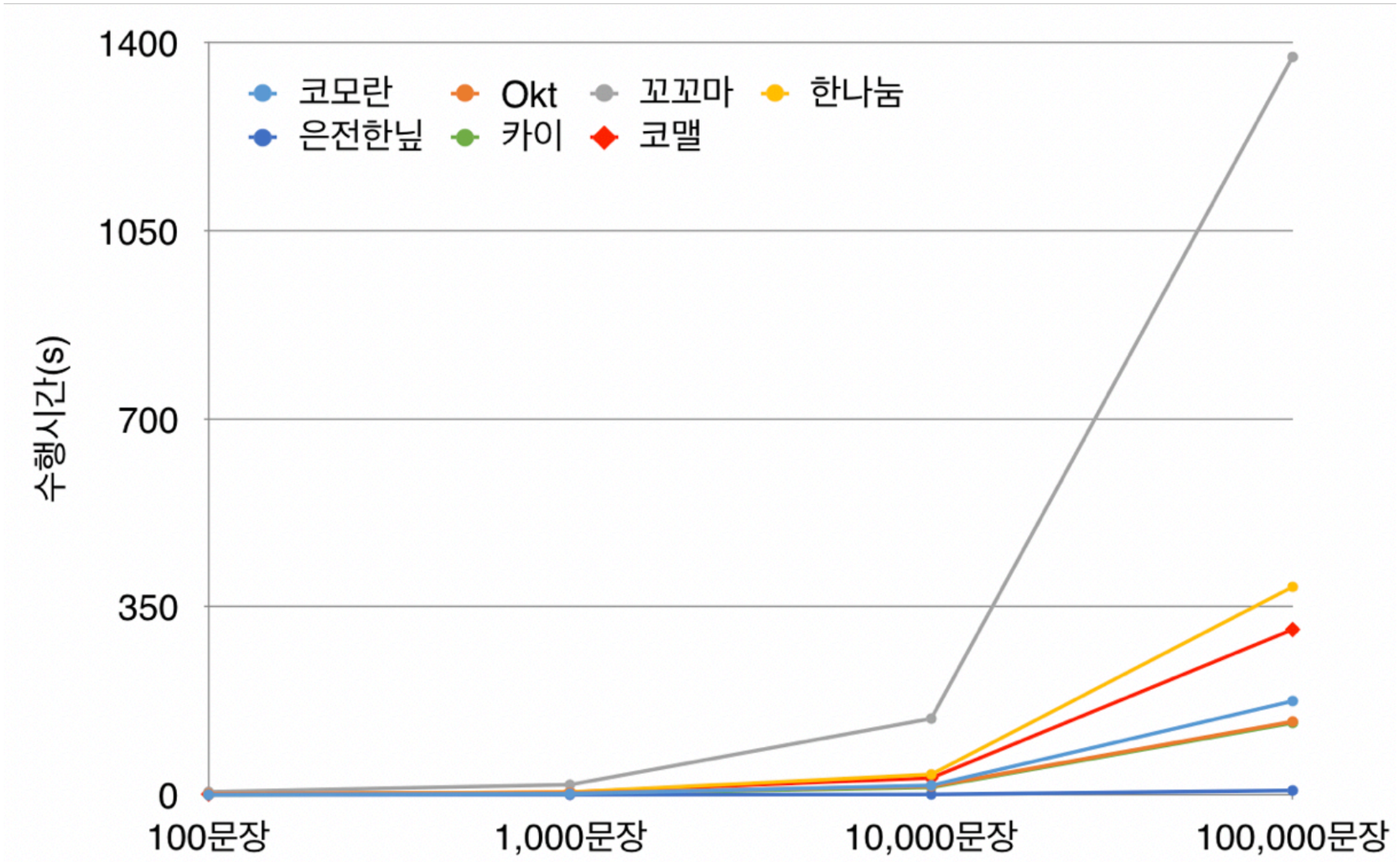
# 형태소 분석기 비교

## 형태소 분석 결과 비교

| 구분 | 형태소 분석 결과 |
|---|---|
| 원문 | 그래도 우리흥은 팀빨이네 이 소린 안 들음. 쏘니 있었기에 토트넘이 성과 낼 수 있기도 하는 거. |
| 코모란 | 그래도/MAJ 우리/NP 흥/NNG 은/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 니/EC 있/VX 었/EP 기/ETN 에/JKB 토트넘/NNP 이/JKS 성과/NNG 내/VV ㄹ/ETM 수/NNB 있/VV 기/ETN 도/JX 하/VV 는/ETM 거/NNB ./SF |
| Okt | 그래도/Adverb 우리/Noun 흥/Noun 은/Josa 팀빨/Noun 이네/Josa 이/Noun 소린/Noun 안/Noun 들음/Verb ./Punctuation 쏘니/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 흥/NNG 은/JX 팀/NNG 빨/NNB 이/VCP 네/EFN 이/MDT 소/NNG 린/UN 안/MAG 듣/VV 음/ETN ./SF 쏠/VV 니/ECD 있/VXV 었/EPT 기에/ECD 토트/NNG 넘/NNB 이/JKS 성과/NNG 내/VV ㄹ/ETD 수/NNB 있/VV 기/ETN 도/JX 하/VV 는/ETD 거/NNB ./SF |
| 한나눔 | 그래도/M 우리흥/N 은/J 팀빨이네/N 이/M 소/N 이/J 리ㄴ/E 알/P ㄴ/E 들/P ㅁ/E ./S 쏘/P 니/E 있/P 었기/E 에/J 토트넘/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 |
| 은전한닢 | 그래도/MAJ 우리/NP 흥/NNG 은/JX 팀/NNG 빨/VV 이/EP 네/EF 이/MM 소린/NNG+JX 안/MAG 들/VV 음/ETN ./SF 쏘/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 우리흥/NNP 은/JX 팀빨/NNG 이/VCP 네/XSN 이/MM 소리/VV ㄴ/ETM 안/MAG 들/VV 음/ETN ./SF 쏘/VV 니/MAG 있/VV 었/EP 기에/EC 토트넘/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 |

# 형태소 분석기 비교

## 수행시간 비교 (Time Analysis)




* Source : 전병진, (2019). 소셜미디어 텍스트 마이닝을 위한 통합 애플리케이션 개발 : KoALA. 연세대학교 정보대학원.

# 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 개쉽월클의 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'), ('인기도', 'NNG'), ('영향', 'NNG'), ('을', 'JKO'), ('미', 'NNG'), ('하', 'XSV'), ('지', 'ECD'), ('달', 'VXV'), ('니', 'ECD'), ('..', 'SW'), ('역시', 'MAG'), ('개쉽월클', '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), (개쓉월클, 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 | 쯔위 | 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 | **KB손해보험**은 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 ./SF … |

* Source : 유현욱(이데일리), [가정의 달! 든든한 금융]KB손보, The간편한치매간병보험 출시, 2019.05.18., http://www.edaily.co.kr/news/read?newsId=01407126622490232&mediaCodeNo=257&OutLnkChk=Y/.
** Source : 김현중, 미등록단어 문제 해결을 위한 비지도학습 기반 한국어자연어처리 방법론 및 응용, 2017.09., https://tv.naver.com/v/2553003/.

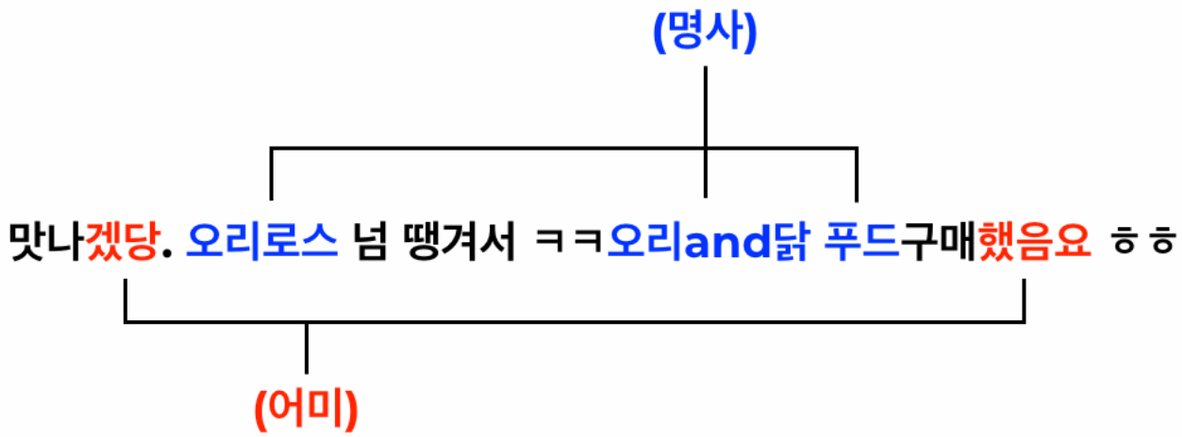
# 미등록 단어 추출

## 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)




* Source : 김현중, 미등록단어 문제 해결을 위한 비지도학습 기반 한국어자연어처리 방법론 및 응용, 2017.09., https://tv.naver.com/v/2553003/.

# 미등록 단어 추출

## 비지도학습과 Char N-gram을 활용한 미등록단어 추출방법

- Char N-gram : 일반적으로 사용하는 단어 단위의 N-gram과 달리 음절 단위로 묶는 방법
- 한국어에서 의미를 지니는 단어는 어절의 왼쪽에 존재함
  - 오리고기를 먹다 → (**오리고기/NNG** + 를/JKO) + (**먹/VV** + 다/EF)
- 단어에 대한 정보가 충분하지 않을 경우에 다음 글자가 등장할 확률을 통해 문맥의 모호성을 판별함

| Char N-gram | 아이 | 아이오 | 아이오아 | 아이오아이 | 아이오아이는 |
|---|---|---|---|---|---|
| 등장확률 (%) | 아니/17.15 | 아이폰/16.60 | 아이오아/87.95 | 아이오아이/100 | 아이오아이의/31.97 |
| | 아이/14.86 | 아이들/13.37 | 아이오닉/7.49 | | 아이오아이는/27.21 |
| | 아시/8.06 | 아이디/9.66 | 아이오와/3.26 | | 아이오아이와/13.61 |
| | 아닌/4.74 | 아이돌/6.77 | 아이오빈/0.65 | | 아이오아이가/12.24 |
| | 아파/4.43 | 아이뉴/6.77 | 아이오페/0.33 | | 아이오아이에/9.52 |
| | 아직/3.85 | 아이오/6.33 | 아이오케/0.33 | | 아이오아이까/1.36 |
| | … | … | … | … | … |

* 2016.10.22. 뉴스기사 기준 측정치

$$\text{cohesion}(\text{“아이오아이”}) = \{ P(\text{아} \rightarrow \text{아이}) * P(\text{아이} \rightarrow \text{아이오}) * P(\text{아이오} \rightarrow \text{아이오아}) * P(\text{아이오아}) \rightarrow (\text{아이오아이}) \}^{1/(5-1)}$$

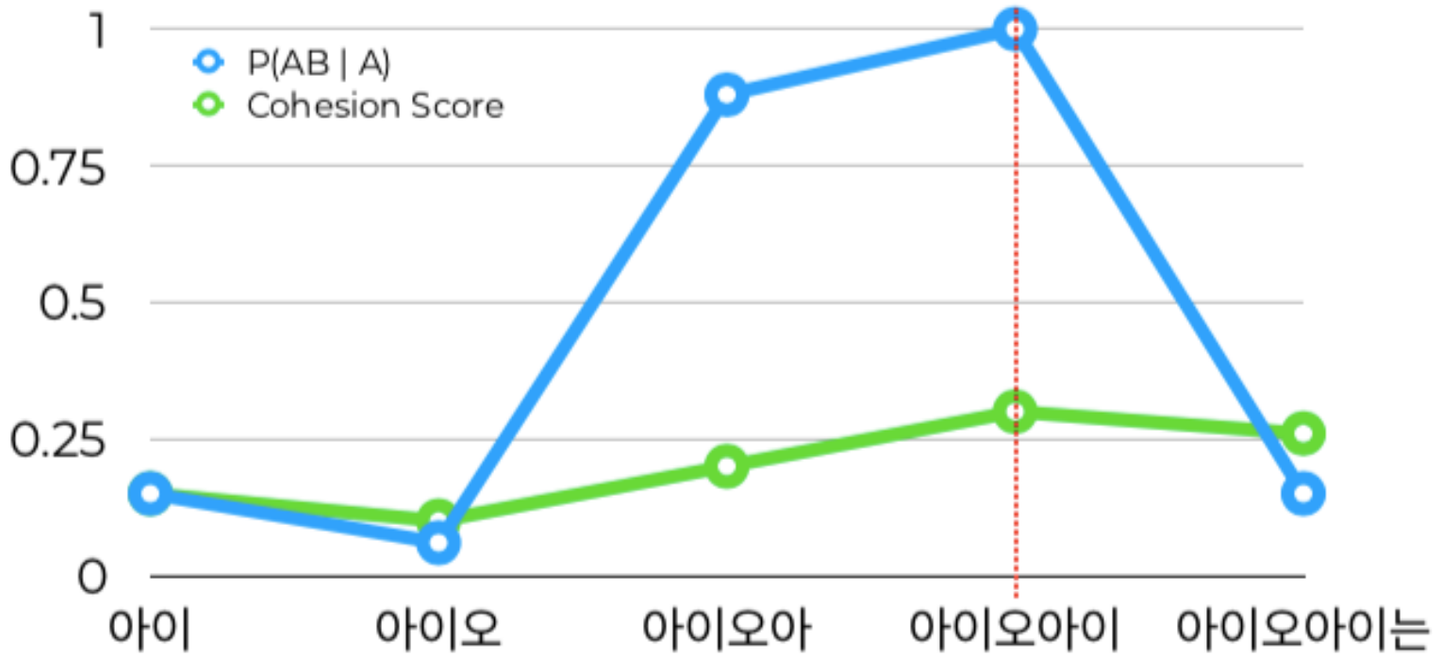
* Source : 김현중, 미등록단어 문제 해결을 위한 비지도학습 기반 한국어자연어처리 방법론 및 응용, 2017.09., https://tv.naver.com/v/2553003/.

# 미등록 단어 추출

## 하루치 뉴스에서 계산한 Cohesion Score

- “아이오아이”로 검색된 하루치 뉴스로 부터 학습된 결과

| Char N- | Frequency | P(AB \| A) | Cohesion Score |
|---|---|---|---|
| 아이 | 4,910 | 0.15 | 0.15 |
| 아이오 | 307 | 0.06 | 0.10 |
| 아이오아 | 270 | 0.88 | 0.20 |
| 아이오아이 | 270 | 1.00 | 0.30 |
| 아이오아이는 | 40 | 0.15 | 0.26 |




* Source : 김현중, 미등록단어 문제 해결을 위한 비지도학습 기반 한국어자연어처리 방법론 및 응용, 2017.09., https://tv.naver.com/v/2553003/.

E.O.D