From text to vector

Data representation and processing framework

Hyunjoong Kim

soy.lovit@gmail.com

github.com/lovit

Vector space representation

- One hot representation (Bag of Words model)
  - (row, column) 은 (문서, 단어) 해당하는 값은 단어의 중요도 혹은 빈도수를 의미합니다.

	기계	학습	은	텍스트	마이닝	는
Doc 1	3	2	5	0	0	0
Doc 2	0	0	0	3	5	5
…	…	…	…	…	…	…

Doc 1 = [(0, 3), (1, 2), (2, 5)]
Doc 2 = [(3, 3), (4, 5), (5, 5)]

	0	1	2	3	4	5
Doc 1	3	2	5	0	0	0
Doc 2	0	0	0	3	5	5
…	…	…	…	…	…	…

Vector space representation

- One hot representation (Bag of Words model)

	0	1	2	3	4	5
Doc 1	3	2	5	0	0	0
Doc 2	0	0	0	3	5	5
…	…	…	…	…	…	…

  - Column 개수 |V|는 문서 전체에서 등장한 단어 종류로, 매우 큽니다.
  - 한 문서에 등장하는 단어의 개수는 적기 때문에, 대부분의 값이 0입니다 (Sparse vector).
  - 문서에 등장한 단어를 쉽게 확인할 수 있어 해석이 쉽지만, 모든 단어는 다른 단어로 취급합니다. 단어간 유사성을 표현하기 어렵습니다.

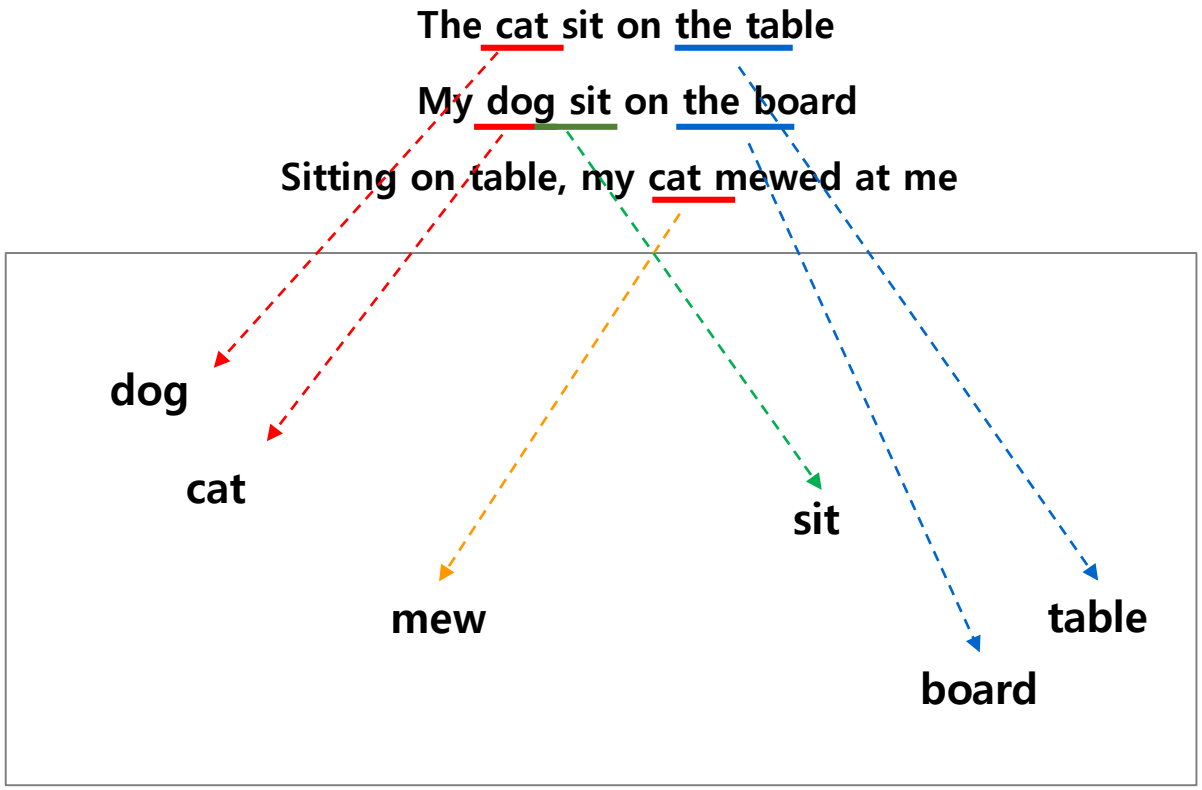
Vector space representation

- Distributed representation
  - 단어/문서를 d차원 공간의 벡터로 표현합니다.
    - 대표적으로 Word2Vec이 있으며, 각 차원이 특별한 의미를 지니지는 않습니다.
  - 벡터 공간은 단어의 "의미적 유사성"을 반영합니다.
    - 벡터가 비슷한 단어/문서는 의미가 비슷합니다.
    - 비슷함의 정의는 알고리즘마다 다릅니다.

'dog'=	[**0.31, -0.21, 2.01, 0.58**, …]
'cat'=	[**0.45, -0.17, 1.79, 0.61**, …]
'topic modeling'=	[**-2.01, 0.03, 0.22, 0.54**, …]
'dim. reduction'=	[**-1.88, 0.11, 0.19, 0.45**, …]

Vector space representation

- Distributed representation
  - 각 벡터는 "의미 공간"에서의 좌표값 역할을 합니다.



< 단어의 의미 공간 >

Why vector representation?

- 많은 머신 러닝 알고리즘은 벡터 공간에서 작동하도록 설계 되었습니다.

- 텍스트 데이터를 알고리즘이 인식할 수 있는 벡터 형태로 변환합니다.
  - “One hot / distributed” or “sparse / dense” representation 모두 벡터로 단어/문서를 기술하는 표현 방법입니다.

Why vector representation?

- 많은 머신 러닝 알고리즘은 벡터 공간에서 작동하도록 설계 되었습니다.

- 텍스트 데이터를 알고리즘이 인식할 수 있는 벡터 형태로 변환합니다.
  - “One hot / distributed” or “sparse / dense” representation 모두 벡터로 단어/문서를 기술하는 표현 방법입니다.

Document clustering

- 군집화 (Clustering)는 비슷한 데이터를 하나의 집합으로 그룹화합니다.
  - 리뷰가 비슷한 영화들의 군집화 결과

cluster # 5	cluster # 2	cluster # 1	cluster # 4
다크 나이트 라이즈	해무	인터스텔라	응답하라 1988
배트맨 대 슈퍼맨: 저스티스의 시작	베를린	미스터 고	인턴
메이즈 러너: 스코치 트라이얼	내가 살인범이다	다크 나이트	님아, 그 강을 건너지 마오
캡틴 아메리카: 시빌 워	신세계	영웅: 샐러멘더의 비밀	카트
빅 히어로	곡성(哭聲)	인셉션	인사이드 아웃
인디펜던스 데이: 리써전스	검은 사제들	트랜스포머 3	형
제이슨 본	악마를 보았다	배틀쉽	비긴 어게인
쥬라기 월드	용의자	스카이라인	두근두근 내 인생
엑스맨: 데이즈 오브 퓨처 패스트	감기	2012	라라랜드
워크래프트: 전쟁의 서막	감시자들	그래비티	반창꼬

Document clustering

- 벡터 공간에서의 거리 척도로 Euclidean, Cosine 등이 이용됩니다.
- Euclidean distance 는 벡터의 크기 (문서의 길이)에 영향을 받습니다.
  - doc 1: 3 단어 / doc 2: 4 단어 / doc 3: 7 단어
  - 문서마다 단어의 개수가 다르므로, 방향이 비슷해도 doc 1과 doc 3은 거리가 멀어집니다

	Term 1	Term 2	Term 3	Term 4	Term 5
Doc 1	1	1	1		
Doc 2			2	1	1
Doc 3	2	2	2		1

Document clustering

- (Euclidean) norm
  - $|v|_2 = \sqrt{v_1^2 + \cdots + v_p^2}$
  - Unit vector of $v = \frac{(v_1, v_2, \ldots v_p)}{\sqrt{v_1^2 + \cdots + v_p^2}}$
    - $\mathrm{n}ormalize\ (\ |(3,0,2)|_2) = \frac{(3,0,2)}{\sqrt{3^2+2^2}}$

Document clustering

- Cosine similarity
  - $\cos(u, v) = \frac{u \cdot v}{|v|_2 \cdot |u|_2} = \frac{\sum u_1 \cdot v_1 + \cdots + u_p \cdot v_p}{\sqrt{v_1^2 + \cdots + v_p^2} \cdot \sqrt{u_1^2 + \cdots + u_p^2}}$
  - $\cos(\,(3, 0, 2), (1, 2, 0)) = \frac{3\times1 \,+\, 0\times2 \,+\, 2\times0}{\sqrt{3^2+2^2}\times\sqrt{1^2+2^2}} = \frac{3}{\sqrt{13\times5}}$
- Cosine distance = 1 − cosine similarity
  - $1 - \frac{u \cdot v}{|v|_2 \cdot |u|_2}$

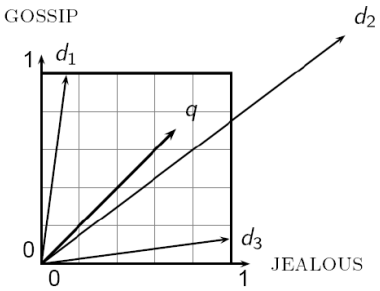
Document clustering

- Bag of words model 에는 Euclidean 보다 Cosine 이 적절합니다.
  - 두 문서에 공통으로 등장한 단어에 대해서만 유사성을 판단합니다.
  - Cosine은 문서 길이에 (벡터의 크기,norm) 영향을 받지 않습니다
  - Sparse representation 에서는 벡터의 방향이 가장 중요합니다.

	Term 1	Term 2	Term 3	Term 4	Term 5
Doc 1	1	1	1		
Doc 2			2	1	1
Doc 3	2	2	2		1

Euclidean(d1, d2) = Euclidean(d1, d3) 이지만,
d1과 d3이 공통된 단어가 많기 때문에 더 비슷

$$Sim(D_1, D_2) = \frac{\sum_i x_{1i} x_{2i}}{\sqrt{\sum_j x_j^2} \sqrt{\sum_k x_k^2}}$$



Document clustering

- Document distance/similarity 를 계산할 때에는 Cosine 이 적합합니다.
  - 문서 표현에 distributed representation 을 이용한다 하더라도 벡터의 방향이 가장 중요합니다.
  - Logistic regression, Neural network 등의 머신 러닝 알고리즘도 벡터 방향이 큰 영향을 미칩니다.

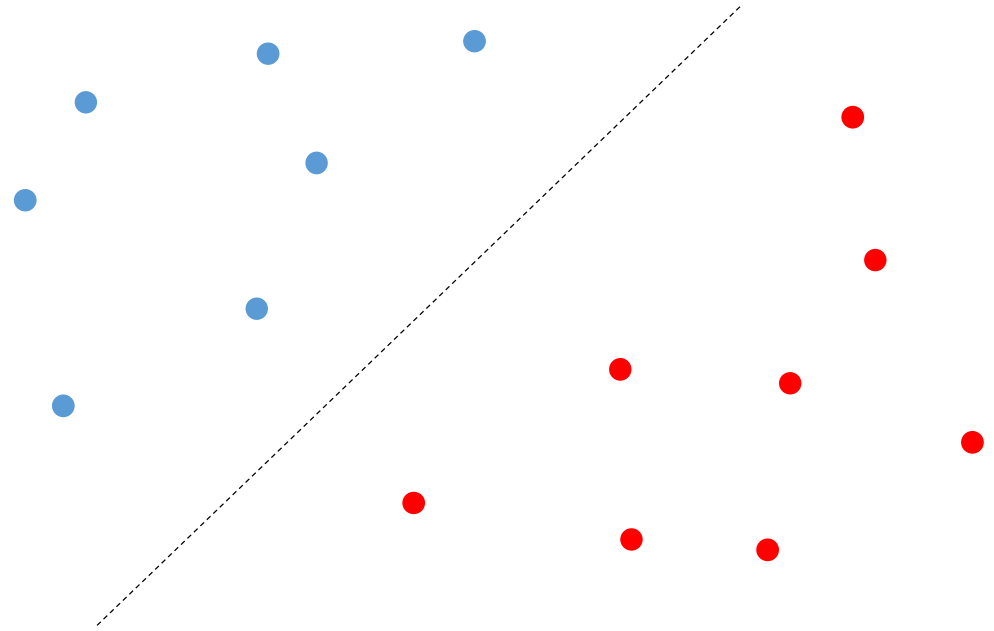
Document classification

- 판별 (Classification)은 데이터의 클래스를 구분합니다.
  - 영화평의 감성 판별 예시

Text	Label	Probability
'크리스토퍼 놀란에게 우리는 놀란다'	Positive	(neg= 0.007, pos= 0.993)
'평점 1점도 아깝다 이게 왜 1위인지 이해가 안됨'	Negative	(neg= 0.991, pos= 0.009)

Document classification

- 판별 (Classification)은 데이터의 클래스를 구분합니다.
  - 클래스 간의 경계를 학습합니다.

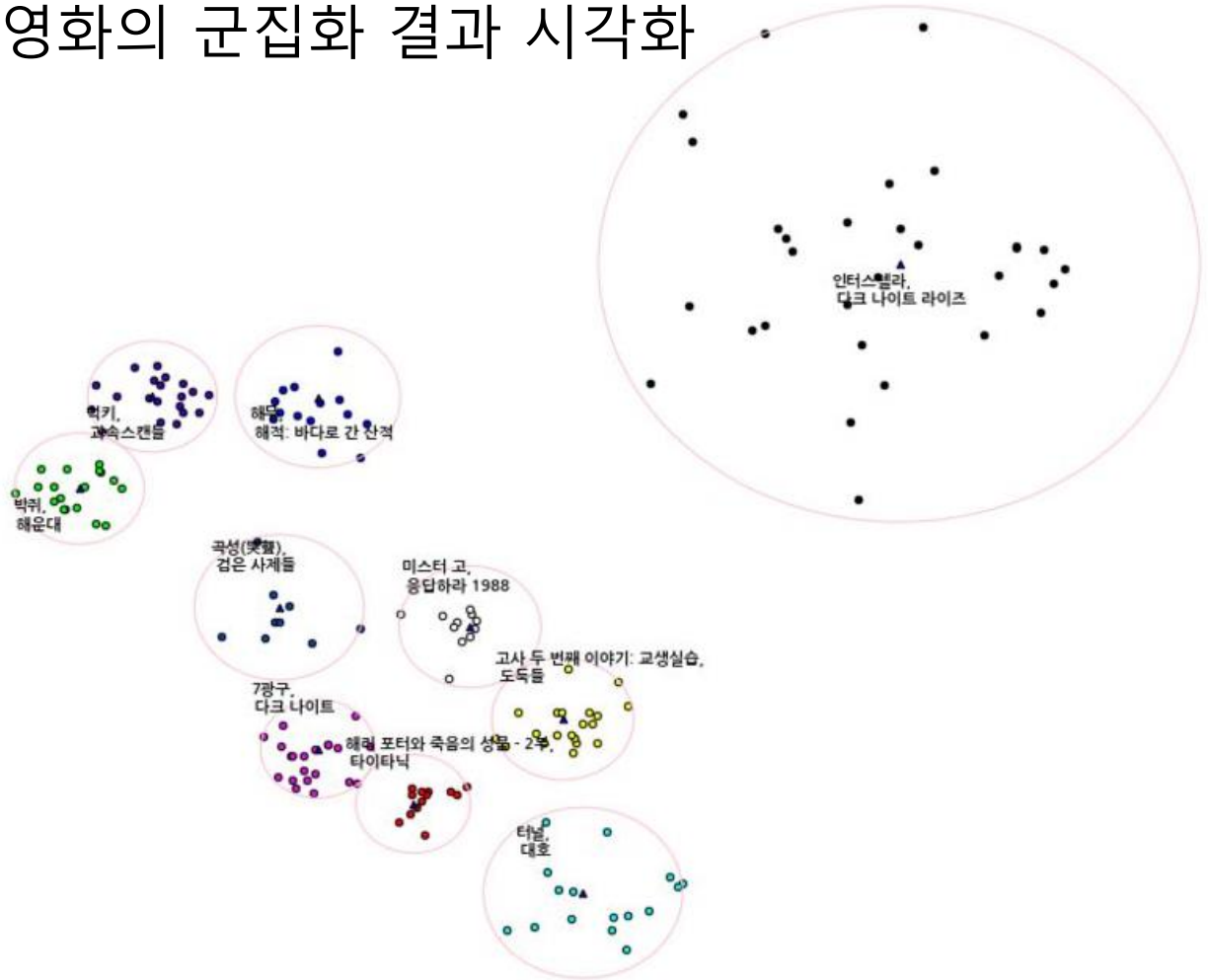


Document classification

- 판별 (Classification) 은 클래스 간의 경계면을 학습합니다.
  - Linear model 은 평면의 경계면을 학습합니다.
    - Logistic regression, Support Vector Machine, Decision Tree
  - Non-linear model 은 곡면의 경계면을 학습합니다.
    - Neural network, Support Vector Machine with Kernel, Deep learning 계열

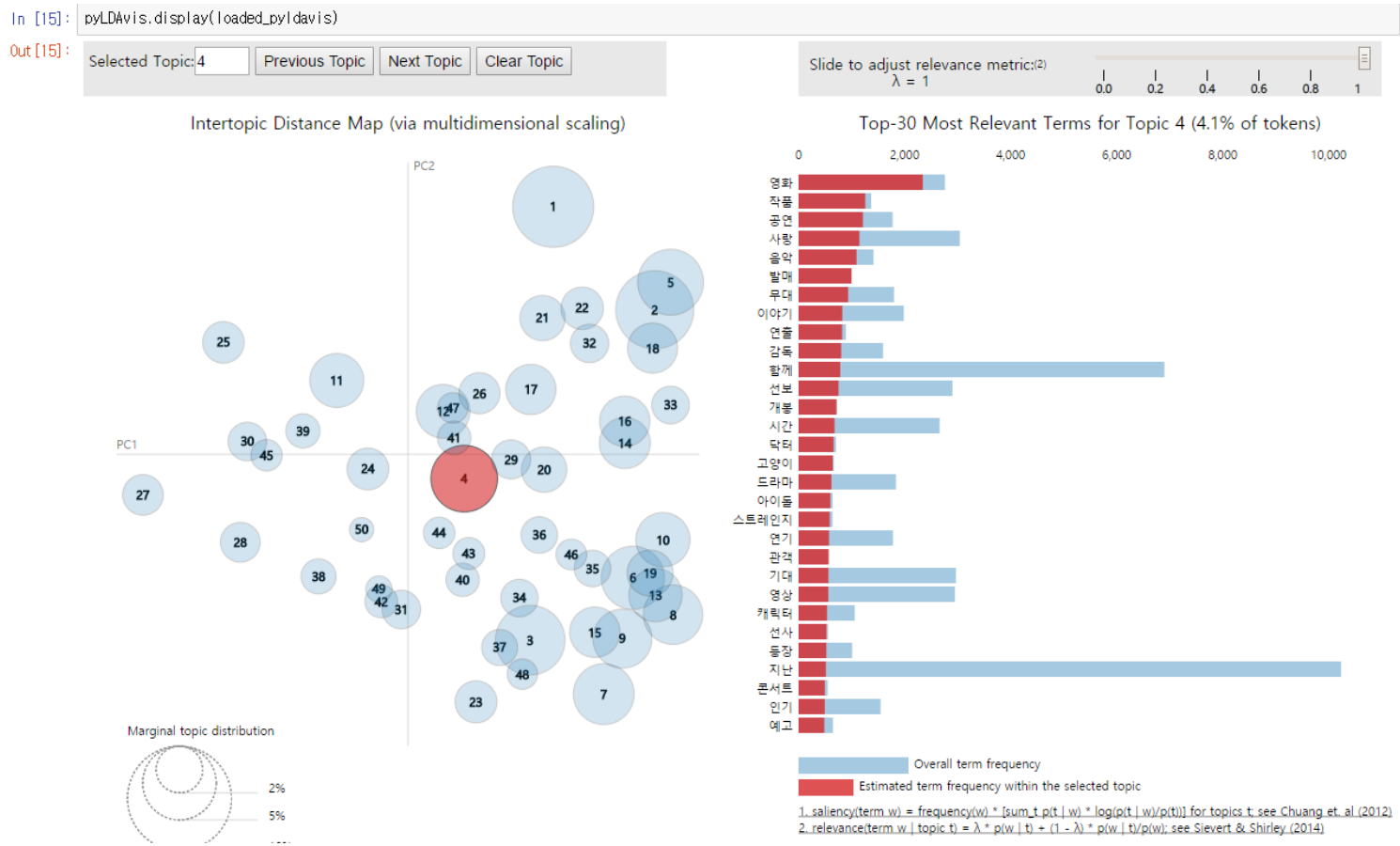
Word/Document Visualization

- 고차원의 벡터를 2차원으로 압축함으로써, 벡터 공간을 설명합니다.
  - 리뷰 기반 영화의 군집화 결과 시각화



Word/Document Visualization

- 고차원의 벡터를 2차원으로 압축함으로써, 벡터 공간을 설명합니다.
  - pyLDAVis 를 이용한 토픽 모델링의 시각화



Keyword extraction

- 키워드 추출은 해당 문서/집합을 요약합니다.
  - 각 영화의 키워드를 영화평 요약

영화, 라라랜드	영화, 신세계	영화, 엑스맨 퍼스트 클래스
관람객 (112.340)	황정민 (98.533)	엑스맨 (106.492)
음악 (37.625)	연기 (89.466)	관람객 (68.034)
사랑 (21.136)	이정재 (46.256)	시리즈 (43.284)
뮤지컬 (20.736)	무간도 (36.069)	역시 (20.009)
꿈을 (19.528)	배우들 (33.970)	액션 (17.090)
여운이 (19.403)	한국 (26.173)	ㅋㅋ (15.580)
아름 (18.650)	신세계 (24.254)	싱어 (15.251)
영상 (18.110)	대박 (23.215)	퍼스트 (14.674)
노래 (16.902)	최민식 (19.561)	진심 (14.637)
좋은 (15.466)	느와르 (19.515)	명작 (12.979)
현실 (15.077)	ㅋㅋ (19.317)	완전 (11.447)
인생 (14.264)	완전 (16.529)	제일 (11.105)
좋고 (13.997)	잔인 (13.826)	이건 (9.705)
감동 (13.482)	역시 (13.320)	울버린 (9.203)
계속 (11.508)	조폭 (12.819)	이번 (9.149)

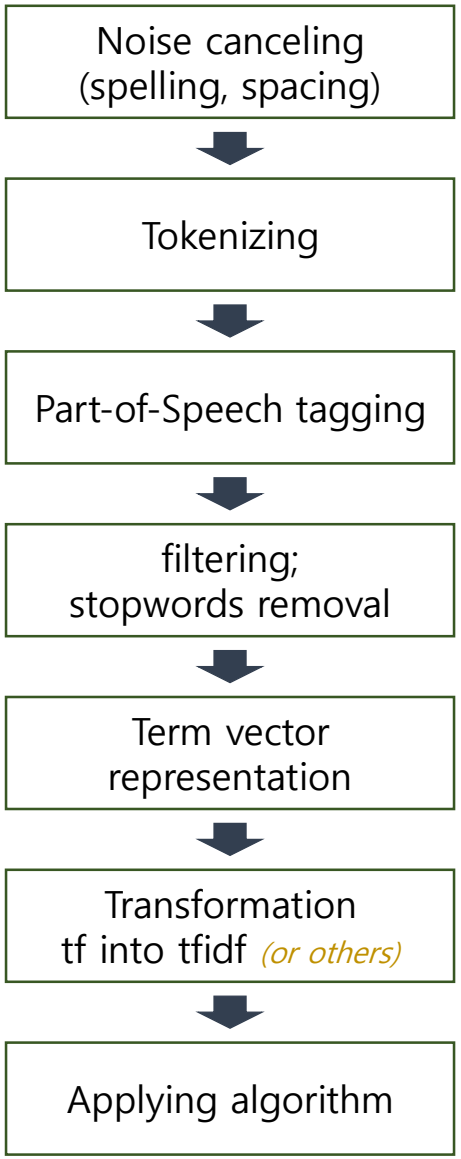
Keyword extraction

- 키워드 추출은 해당 문서/집합을 요약합니다.
  - 문서 군집화 결과의 레이블링

no.	meaning	Keywords
1	렌트카 광고	제주렌트카, 부산출발제주도, 제주신, 이끌림, 제주올레, 왕복항공, 불포함, 제주도렌트카, 064, 롯데호텔, 자유여행, 객실, 제주여행, 특가, 해비치, 제주시, 제주항, 티몬, 2박3일, 올레, 유류, 항공권, 조식, 제주도여행, 제주공항, 2인
2	중고차 매매	최고급형중고, 최고급, 프리미어, 프라임, 2011년식, YF소나타TOP, 2010년식, 풀옵션, 2011년, YF소나타PR, 1인, Y20, 2010년, 완전무사고, 판매완료, 군포, 검정색, YF쏘나타, 2011, 하이패스, 2010, 무사고, 등급, 파노라마, 허위매물
3	클래식 음악	금관악기, 아이엠, Tru, 트럼펫, 트럼, 나팔, 금관, 텔레만, Eb, 호른, 오보에, Tr, Concerto, 하이든, 협주곡, Ha, 악기, 연주하는, 오케, 오케스트라, 독주, 악장, 작곡가, 곡
4	아이비 "유혹의 소나타 "	Song, 공부할, 부른, 노래, 가사, 부르는, 가수, 보컬, 목소리, 발라드, 명곡, 신나, 들으면, 듣기, 유혹의, 앨범,아이비, 제목
5	광염 소나타 및 일제강점기 소설들	백성수, 발가락, 현진, 이광수, 김유, 자연주의, 친일, 평양, 운수, 유미, 저지르, 야성, 탐미, 김동인, 복녀, 광염, 닮았다, 사실주의, 광기, 저지, 1920, 단편소설, 범죄, 감자, 동인, 한국문학

Text data processing

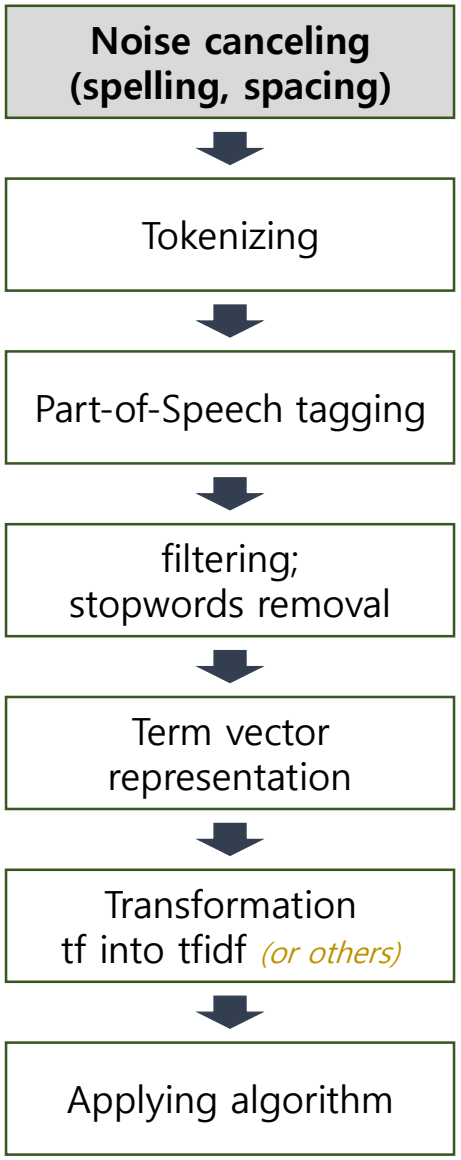
Framework


Noise canceling
(spelling, spacing)
Tokenizing
Part-of-Speech tagging
filtering;
stopwords removal
Term vector
representation
Transformation
tf into tfidf (or others)
Applying algorithm

Spelling

- 한 단어가 다르게 적힌다면,
  - 같은 개념이 다른 단어로 표현됩니다.
  - Bag of words model 의 벡터 공간이 커집니다
  - 미등록단어 (Out of vocabulary) 문제가 발생합니다

- 사전에 존재하는 올바른 단어로 수정합니다
  - Edit distance
  - FastText 같은 embedding 은 이를 우회합니다.



Spelling

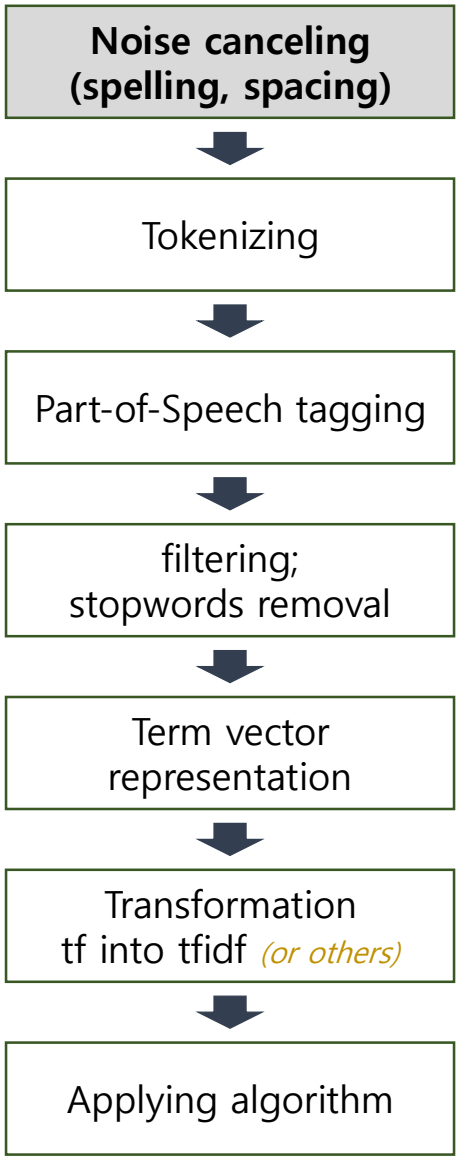
- 오자는 수기로 입력된 데이터에서 자주 발생합니다

데이터
제조외
제조, 도매, 부동산
건설업
조립금속제조, 기타화학제조
서빗스 도소매
편의점, 담배
소매업.서비스업. 부동산업
식음료
제조 및 도소, 부동산업
제조업

사전		
가구내 고용활동	보험	운수
가스	부동산	원료재생
개인	사업시설관리	음식점
건설	사업지원	임대
과학	사회복지	임업
광업	서비스	자가소비생산활동
교육	소매	전기
국제	수도사업	전문
금융	수리	정보
기술	숙박	제조
기타	스포츠	증기
농업	어업	출판
단체	여가	폐기물처리
도매	영상	하수처리
방송통신	예술	협회
보건	외국기관	환경복원

spacing

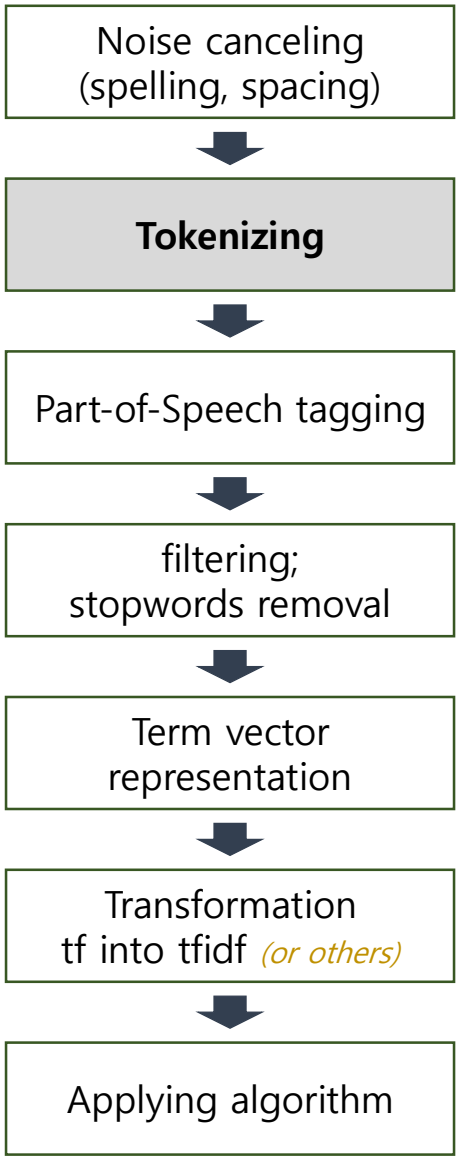
- 한국어의 어절은 띄어쓰기로 구분됩니다
  - 띄어쓰기 오류는 자연어처리기의 정확도와 계산 시간에 영향을 줍니다
  - 띄어쓰기 오류에 대응할 수 있는 토크나이저 혹은
  - 띄어쓰기 오류 교정이 필요합니다



Tokenizing

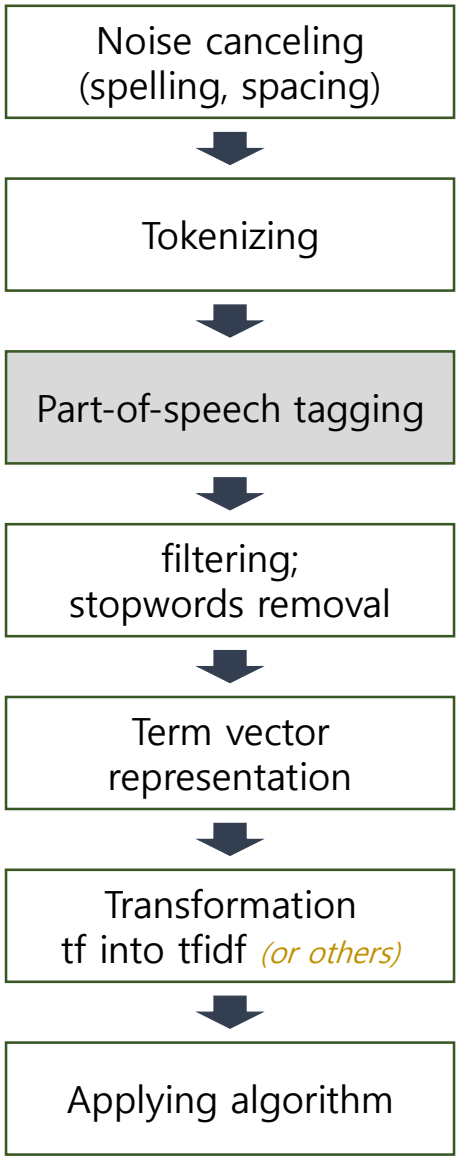
- 토크나이징은 어절에서 단어를 나누는 것입니다
  - [토크나이징, 은, 어절, 에서, 단어,를, 나누는, 것, 입니다]

- 정확히는 "문장"을 "토큰"으로 나누는 것입니다
  - 토큰은 n-gram, 어절, 단어, phrase 등으로 목적에 따라 다르게 정의합니다
  - 이 자료에서는 좁은 의미의 '토크나이저', 문장을 단어로 나누는 것으로 이야기합니다.



Part-of-Speech tagging

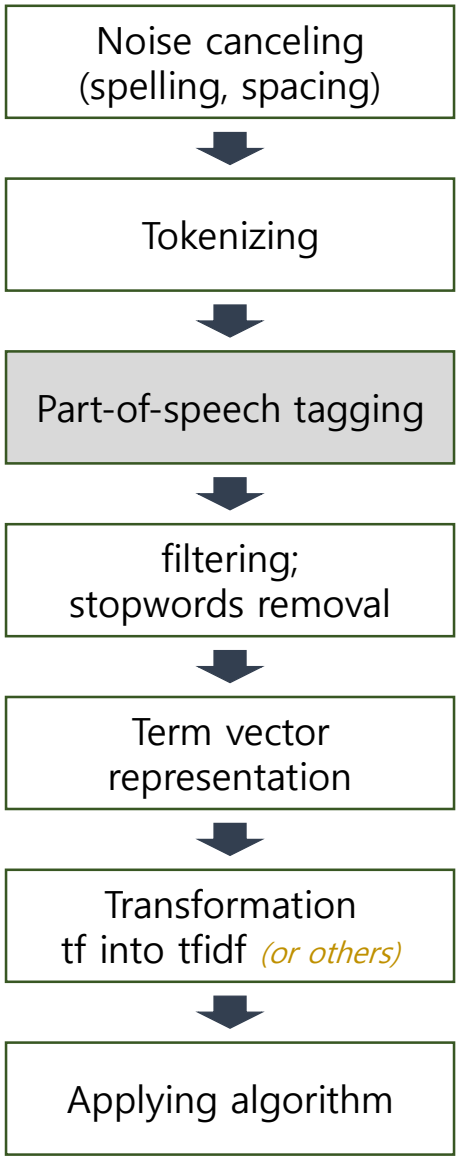
- 품사 판별은 주어진 단어의 품사를 구분합니다
  - [토크나이징, 은, 어절, 에서, 단어,를, 나누는, 것, 입니다] →
    [(토크나이징, 명사),
      (은, 조사),
      (어절, 명사),
      (에서, 조사),
      (단어, 명사),
      (를, 조사),
      (나누는, 동사),
      (것, 명사),
      (입니다, 형용사)]



Morphological analysis

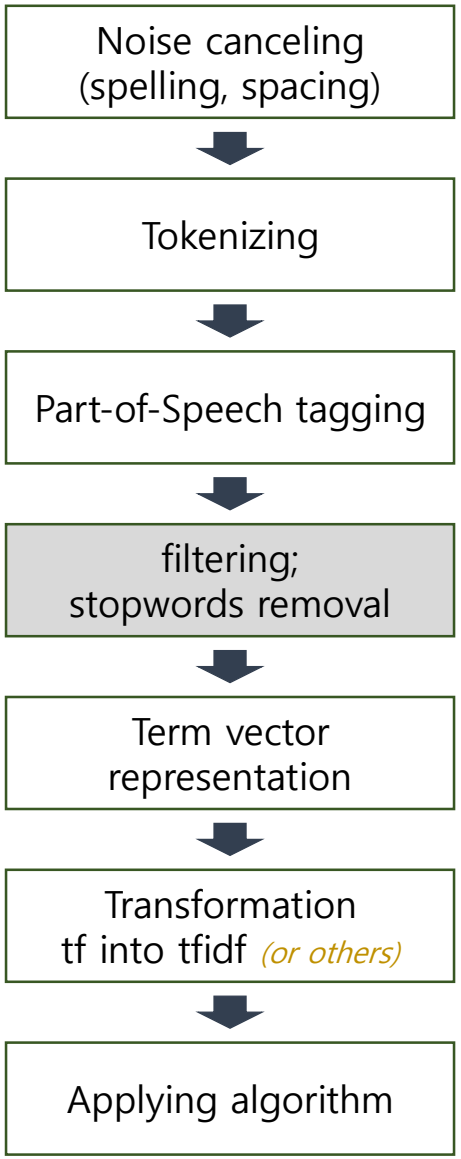
- 형태소 분석은 단어의 형태소를 인식합니다.
  - 형태소는 단어를 구성하는 최소단위 입니다.
  - 품사 판별: "입니다" → 형용사
  - 형태소 분석: "입니다"

    → 이/형용사어근 + ㅂ니다/어미
- 형태소 분석을 바탕으로 단어의 품사를 추정할 수 있습니다.



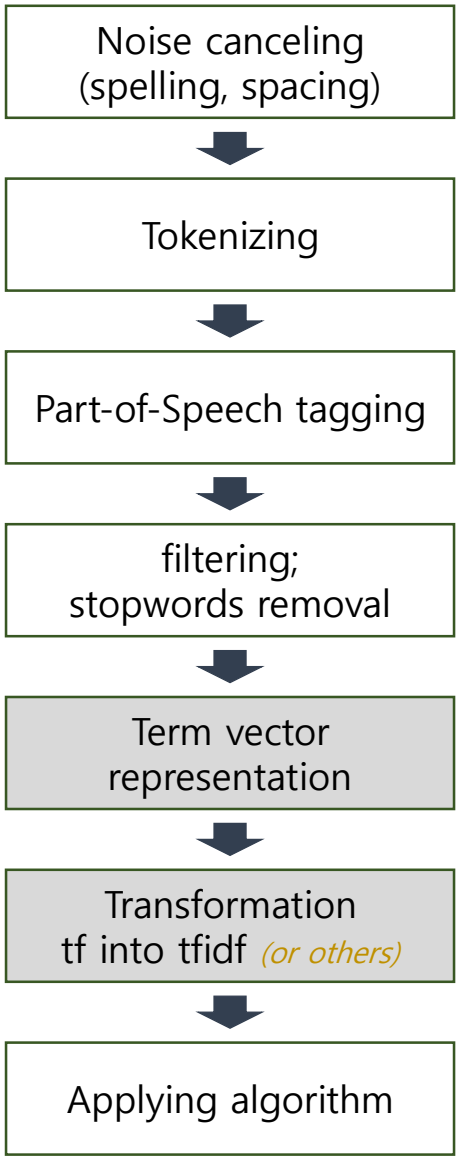
Stopwords removal

- Bag of words model 의 불필요한 단어를 제거합니다
  - 거의 등장하지 않는 단어 (min count 기준 커팅)
  - -은, -는과 같은 조사, (영어에서는 a, the, am, are, ...)
  - 키워드 추출을 위한 명사 선택



Term vector representation

- Term weighting
  - $(i, j) = weight$
  - Term frequency vector 는 문서 i 에서의 단어 j 의 중요도를 단어의 빈도수로 표현
  - (i, j) 의 중요도는 정의하기 나름이며, 반드시 TF 혹은 TF-IDF 를 이용해야 하는 것도 아닙니다



Term vector representation

- TF-IDF는 Information Retrieval 을 위하여 제안된 term weighting 입니다
  - 흔하게 등장하는 단어의 영향력을 줄입니다.

$$\mathrm{TF-IDF}(w,d) = \mathrm{TF}(w) \times \log\left(\frac{N}{\mathrm{DF}(w)}\right)$$

TF(w): 단어 w가 문서 d에서 등장한 빈도 수

DF(w): 단어 w가 등장한 문서의 개수

N: 문서집합에서 문서의 총 개수.

- DF(w) = N 이면, 그 단어는 정보력이 없기 때문에 TF-IDF(w,d) = 0

Term vector representation

- Document frequency (df) 가 큰 단어는 정보력이 적습니다
  - 어디에나 등장하는 것은 정보력이 없음을 의미합니다
  - 무의미하거나 문법적인 역할을 합니다 (조사)
  - 흔하게 등장하기 때문에 문서 간 거리를 계산할 때에도 무시할 수 있습니다

Word	Document frequency
1위	50
트와이스	500
노래	1000
-은, -는	10,000

Term vector representation

흔한 단어의 영향력은 낮추고

	이번	트와이스	의	는	TT	1위	노래	음악중심	빅뱅
Doc 1	5	12	8	15	8	3	3	2	0
Doc 2	1	0	7	8	0	4	1	0	8
Doc 3	2	1	5	7	1	1	0	2	4

	이번	트와이스	의	는	TT	1위	노래	음악중심	빅뱅
Doc 1	0.3	2.5	0.2	0.1	3.2	3.6	0.3	0.8	0
Doc 2	0.06	0	0.175	0.57	0	4.8	1	0	3.3
Doc 3	0.12	0.08	0.125	0.48	0.4	1.2	0	0.8	1.65

문서 집합 전체에서 흔하게 등장하지 않고,
특정 문서에서 자주 나오는 단어의 영향력을 높임

article

박태환이 금지 약물 양성반응 통보를 받은 이후에 '도핑 파문'이 일어난 T 병원 김모 원장과 나눈 대화 내용을 녹음해 검찰에 제출한 것으로 알려졌다. 일부 매체는 이에 대해 "박태환이 김 원장에게 '아무 문제가 없는 주사약이라고 해놓고 이게 무슨 일이냐'라고 강하게 따진 것으로 전해졌 …

토크나이징/ 품사판별 후

[박태환/N] [이/J] [금지/N] [약물/N] [양성반응/N] [통보/N] [를/J] …

Stopwords removal

Term	박태환/N	~~이/J~~	금지/N	약물/N	양성반응/N	통보/N	~~를/J~~	…
frequency	28	35	12	15	13	5	32	…

의미를 지닌 단어 선택: 명사, 동사, 형용사
문법 기능을 하는 단어 배제: 조사, 어미

Vector representation

Term	1		55	21	3	27		…
frequency	28		12	15	13	5		…

Text data processing

KoNLPy

- KoNLPy 를 이용하여 Python 에서 품사 판별 / 형태소분석을 합니다

  - KoNLPy 는 Python 이 아닌 언어로 구현된 다양한 라이브러리를 Python에서 이용할 수 있도록 도와줍니다.
  - 통일된 인터페이스는 여러 라이브러리를 비교하며 편하게 이용할 수 있도록 도와줍니다.

Quick starting

```
from konlpy.tag import Kkma


kkma = Kkma()
print(kkma.nouns(u'질문이나 건의사항은 깃헙 이슈 트래커에 남겨주세요.'))
```

```
[질문, 건의, 건의사항, 사항, 깃헙, 이슈, 트래커]
```

```
print(kkma.nouns(u'질문이나 건의사항은 깃헙 이슈 트래커에 남겨주세요.'))
```

```
[(오류, NNG), (보고, NNG), (는, JX), (실행, NNG), (환경, NNG), (,, SP),
  (에러, NNG), (메세지, NNG), (와, JKM), (함께, MAG), (설명, NNG), (을, JKO),
  (최대한, NNG), (상세히, MAG), (!, SF), (^^, EMO)]
```

KoNLPy

- 형태소 분석기마다 태그의 수준과 표기법이 다릅니다.
  - KoNLPy에서 제공하는 라이브러리들의 품사 태그 비교표 입니다.
  - (예문) '이건 테스트 문장입니다'

Hannanum	Kkma	Twitter
[('이', 'N'), ('이', 'J'), ('건', 'E'), ('테스트', 'N'), ('문장', 'N'), ('이', 'J'), ('ㅂ니다', 'E')]	[('이건', 'NNP'), ('테스트', 'NNG'), ('문장', 'NNG'), ('이', 'VCP'), ('ㅂ니다', 'EFN')]	[('이건', 'Noun'), ('테스트', 'Noun'), ('문장', 'Noun'), ('입니', 'Adjective'), ('다', 'Eomi')]

Out of vocabulary

- 새롭게 만들어진 단어들은 잘 인식되지 않습니다.

```
from konlpy.tag import Kkma, Twitter
kkma = Kkma()
kkma.pos('너무너무너무는 아이오아이의 노래에요')
```

```
너무/MAG, 너무너무/MAG, 는/JX, 아이오/NNG, 아이/NNG, 의/JKG, 노래/NNG, 에/JKM, 요/JX
```

```
twitter = Twitter()
twitter.pos('너무너무너무는 아이오아이의 노래에요')
```

```
너무/Noun, 너무/Noun, 너무/Noun, 는/Josa, 아이오/Noun, 아이/Noun, 의/Josa, 노래/Noun,
에요/Josa
```

형태소분석 vs 품사판별

- 형태소 분석은 단어의 구조까지 파악하며, 품사 판별은 품사를 인식합니다.
  - (예시) 재수강하겠어
    - 형태소분석: [재/관형사 + 수강/명사 + 하/동사 + 겠/선어말어미 + 어/어말어미]
    - 품사판별 : [재수강/명사 + 하겠어/동사]

한국어 품사				
불변어	체언	명사	대명사	수사
	수식언	관형사	부사	
	관계언	조사		
	독립언	감탄사		
가변어	용언	동사	형용사	

형태소분석 vs 품사판별

- 품사 판별을 위하여 형태소 분석이 이용될 수 있습니다.
  - 형태소 분석은 단어의 구성 요소들을 분해하여 인식하는 과정입니다.

SENT: 재공연을 했어요

POS: (**재공연**, 명사), (**을**, 조사), (**했어요**, 동사)
MORPHEMES: (**재**, 관형사), (**공연**, 명사), (**을**, 조사), (**하**, 동사), (**었**, 선어말어미), (**어요**, 종결어미)

- KoNLPy에 구현되어 있는 라이브러리들은 형태소분석기가 많습니다.
  - 꼬꼬마 형태소 분석기 / 한나눔 형태소 분석기 / 코모란 형태소 분석기 / MeCab-ko 형태소 분석기 / 트위터 한국어 처리기

형태소분석 vs 품사판별

- 한글은 사실상 표의문자이기 때문에 대부분 음절마다 의미를 지닙니다.
  - 특히 신조어 명사의 경우 부분음절들이 단어인 경우가 많습니다.
  - 형태소 분석기 입장에서는 모르는 단어를 모른 체 두는 것보다, 아는 단어들로 분해하려는 경향이 있습니다.

```
너무/MAG, 너무너무/MAG, 는/JX, 아이오/NNG, 아이/NNG, 의/JKG, 노래/NNG, 에/JKM, 요/JX
```

```
비/Noun, 선/Verb, 실/PreEomi, 세/PreEomi, 가/Eomi, 드러났/Verb, 다/Eomi
```

- 분석할 데이터에 적절한 단어 사전을 구축해야 합니다.