

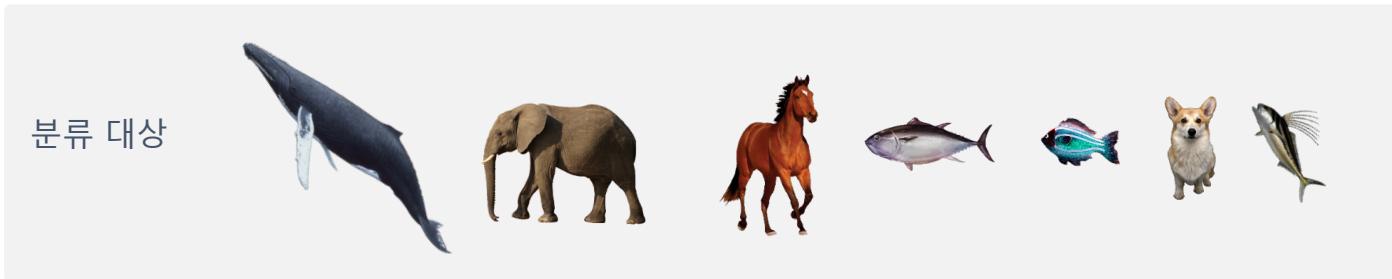
언어를 표현하는 방법, 임베딩

각종 언어(자연어)를 배우는 모델이 어떻게 언어를 표현하는지를 학습합니다.

특징 추출과 분류

자연어 처리 (Natural Language Processing, NLP)

- ‘분류’를 위해선 데이터를 수학적으로 표현
- 먼저, 분류 대상의 특징 (Feature)을 파악 (Feature extraction)

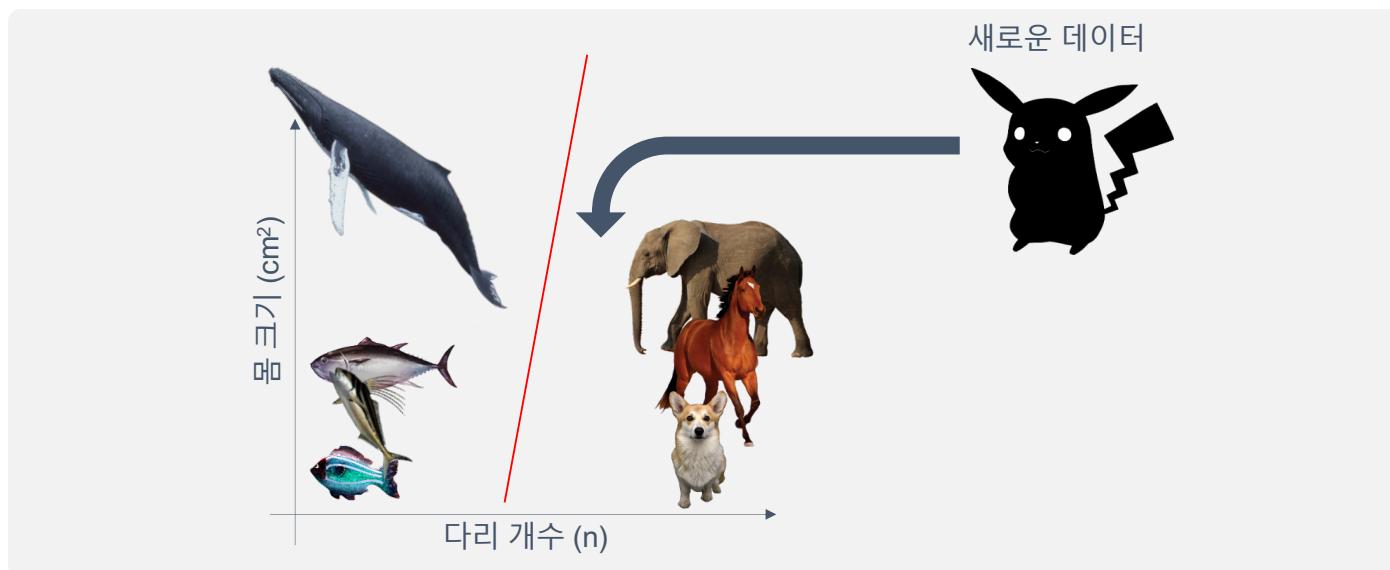


크기가 다양
다리의 개수가 다양

특징 추출과 분류

자연어 처리 (Natural Language Processing, NLP)

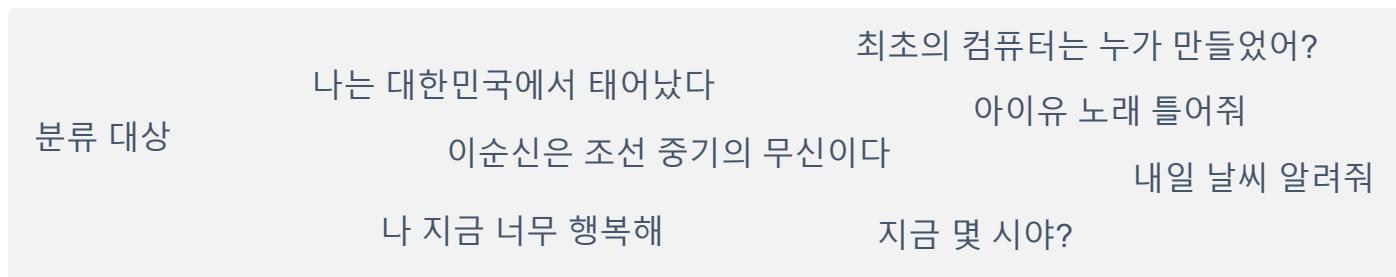
- 분류 대상의 특징 (Feature)를 기준으로, 분류 대상을 **그래프** 위에 표현 가능
- 분류 대상들의 경계를 수학적으로 나눌 수 있음 (Classification)
- 새로운 데이터 역시 특징을 기준으로 그래프에 표현하면, 어떤 그룹과 유사한지 파악 가능



자연어에서의 특징 추출과 분류

자연어 처리 (Natural Language Processing, NLP)

- 과거에는 사람이 직접 특징 (Feature)를 파악해서 분류
- 실제 복잡한 문제들에선 분류 대상의 특징을 사람이 파악하기 어려울 수 있음
- 이러한 특징을 컴퓨터가 스스로 찾고 (Feature extraction), 스스로 분류 (Classification) 하는 것이 ‘기계학습’의 핵심 기술



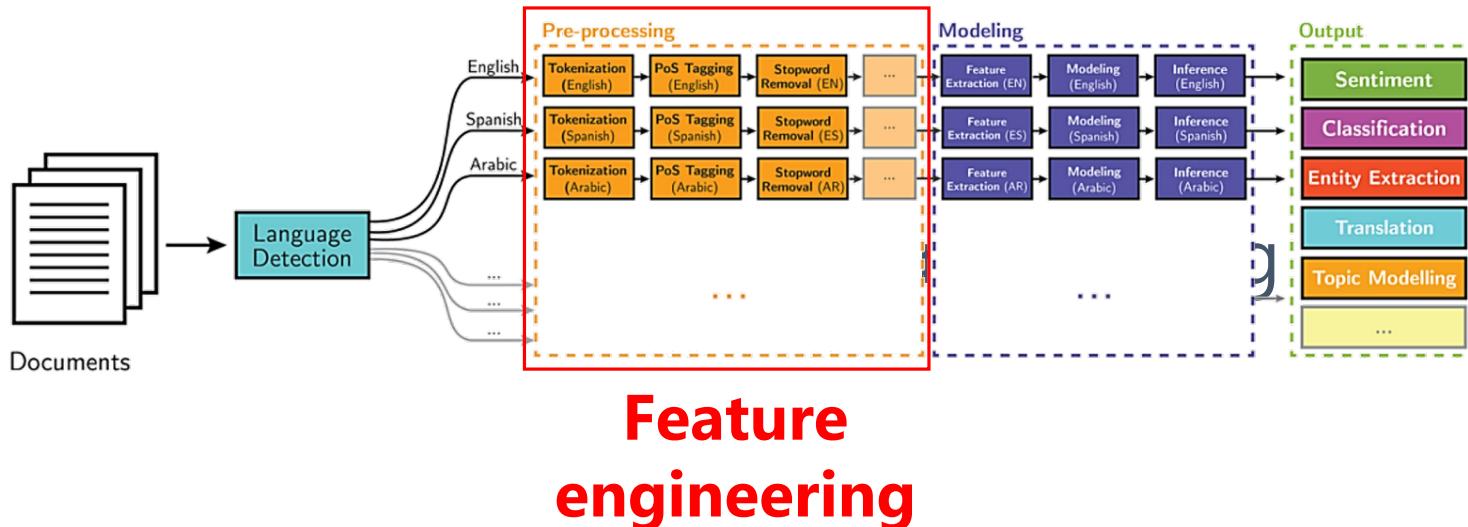
분류 대상의 특징



자연어에서의 특징 추출과 분류

자연어 처리 (Natural Language Processing, NLP)

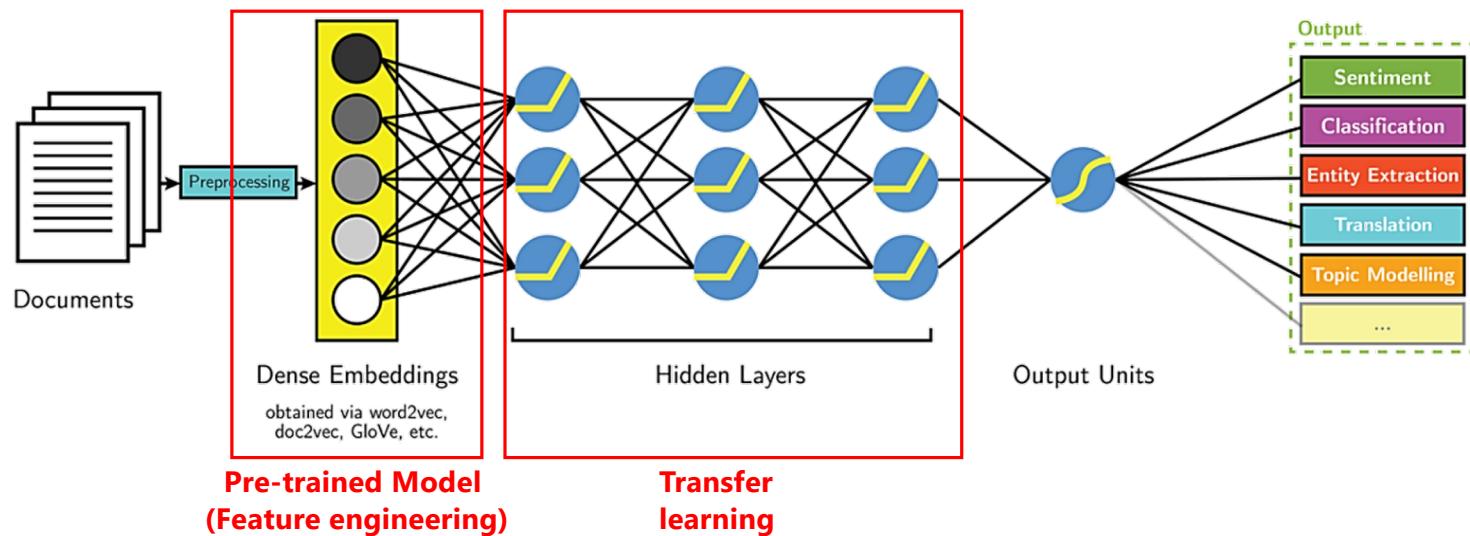
- 과거에는 사람이 직접 특징 (Feature)를 파악해서 분류
- 실제 복잡한 문제들에선 분류 대상의 특징을 사람이 파악하기 어려울 수 있음
- 이러한 특징을 컴퓨터가 스스로 찾고 (Feature extraction), 스스로 분류 (Classification) 하는 것이 ‘기계학습’의 핵심 기술



자연어에서의 특징 추출과 분류

자연어 처리 (Natural Language Processing, NLP)

- 과거에는 사람이 직접 특징 (Feature)를 파악해서 분류
- 실제 복잡한 문제들에선 분류 대상의 특징을 사람이 파악하기 어려울 수 있음
- 이러한 특징을 컴퓨터가 스스로 찾고 (Feature extraction), 스스로 분류 (Classification) 하는 것이 ‘기계학습’의 핵심 기술



자연어에서의 특징 추출과 분류

자연어 처리 (Natural Language Processing, NLP)

1. 사전 학습:

자기지도학습

언어 모델

2. 미세조정:

감독 학습

언어 모델

분류기

[Wikipedia] Super Bowl LIV was an American football game played to determine the champion of the National Football League (NFL) for the 2019 season. The American Football Conference (AFC) champion Kansas City Chiefs defeated the National Football Conference (NFC) champion San Francisco 49ers 31-20, marking their first Super Bowl victory since Super Bowl IV and ...

[Fiction] It was the best of times, it was the worst of times, it was the age of wisdom, it was the age of foolishness, it was the epoch of belief, it was the epoch of incredulity, it was the season of Light, it was the season of Darkness, it was the spring of hope, it was the winter of despair, ...

입력 문장	라벨
A smile on your face.	참
Contains no wit, only labored gags.	거짓
Rich veins of funny stuff in this movie.	참
That's far too tragic to merit such superficial treatment.	거짓

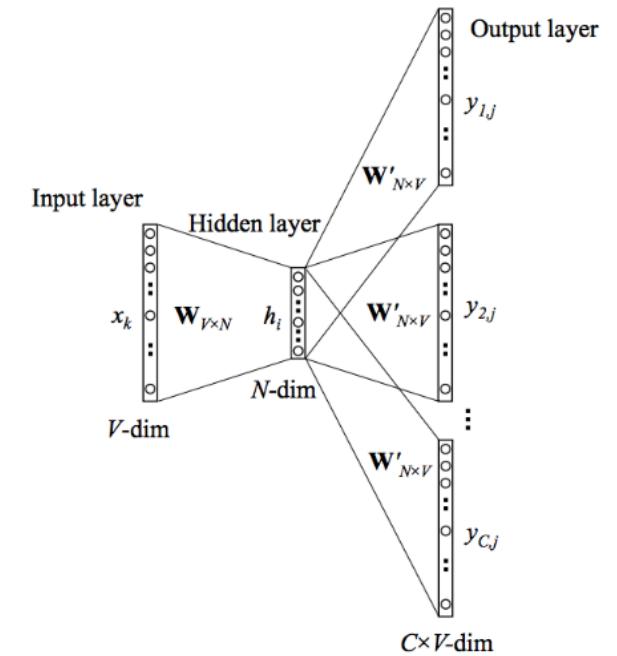
Word Representation

- 말뭉치(corpus)의 분포 정보(co-occurrence)를 활용
- 개별 단어의 의미나 형태를 고려하지 않아도 유사도 계산 가능
- 크게 두가지 방법으로 분류됨
 1. Prediction-based model: 특정 단어를 출력하는 모델이며, 모델의 가중치를 개별 단어의 벡터로 활용
 2. Count-based model: 단어 분포 행렬을 만들고, 행렬의 reconstruction error를 줄이는 방식으로 차원 축소를 진행. 행렬의 열 또는 행 벡터가 개별 단어의 벡터가 되는 방식

Prediction-based Model: Word2Vec

Source Text	Training Samples
The quick brown fox jumps over the lazy dog. ➔	(the, quick) (the, brown)
The quick brown fox jumps over the lazy dog. ➔	(quick, the) (quick, brown) (quick, fox)
The quick brown fox jumps over the lazy dog. ➔	(brown, the) (brown, quick) (brown, fox) (brown, jumps)
The quick brown fox jumps over the lazy dog. ➔	(fox, quick) (fox, brown) (fox, jumps) (fox, over)

$$P(o|c) = \frac{\exp(u_o^T v_c)}{\sum_{w=1}^W \exp(u_w^T v_c)}$$



Count-based Model: LSA(TF-IDF/PMI with SVD)

$$\begin{matrix}
 A & = & U & \times & \Sigma & \times & V^T \\
 \begin{matrix} A_k \\ \vdots \\ \end{matrix} & & \begin{matrix} u_k \\ \vdots \\ \end{matrix} & \times & \begin{matrix} \Sigma_k \\ \ddots \\ 0 \\ \vdots \\ 0 \\ \end{matrix} & \times & \begin{matrix} v_k^T \\ \vdots \\ \end{matrix}
 \end{matrix}$$

$m \times n$ $m \times m$ $m \times n$ $n \times n$

→ 특이벡터들끼리는 직교한다($U^T U = V^T V = I$)는 사실을 활용해 양변에 특이벡터 행렬을 내적해 원데이터 행렬 A 의 차원을 축소

Word embedding – Word2Vec

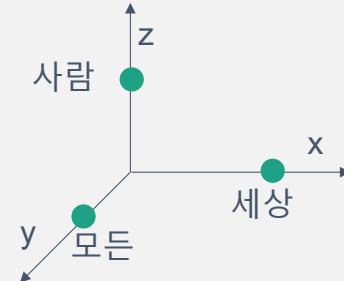
자연어 처리 (Natural Language Processing, NLP)

- 자연어를 어떻게 좌표평면 위에 표현할 수 있을까?
- 가장 단순한 표현 방법은 one-hot encoding 방식 → Sparse representation

세상 모든 사람

단어	Vector
세상	[1, 0, 0]
모든	[0, 1, 0]
사람	[0, 0, 1]

n개의 단어는 n차원의 벡터로 표현



단어 벡터가 sparse해서 단어가 가지는 ‘의미’를 벡터 공간에 표현 불가능

Word embedding – Word2Vec

자연어 처리 (Natural Language Processing, NLP)

- Word2vec (word to vector) 알고리즘: 자연어 (특히, 단어)의 의미를 벡터 공간에 임베딩
- 한 단어의 주변 단어들을 통해, 그 단어의 의미를 파악

جرو

كلب



컴퓨터가 볼 때, 자연어는 우리가 보는 아랍어처럼 ‘**기호**’로만 보일 뿐!

Word embedding – Word2Vec

자연어 처리 (Natural Language Processing, NLP)

- Word2vec (word to vector) 알고리즘: 자연어 (특히, 단어)의 의미를 벡터 공간에 임베딩
- 한 단어의 주변 단어들을 통해, 그 단어의 의미를 파악

جو
(개) 가 멍멍! 하고 짖었다.

كلب
(강아지) 가 멍멍! 하고 짖었다.

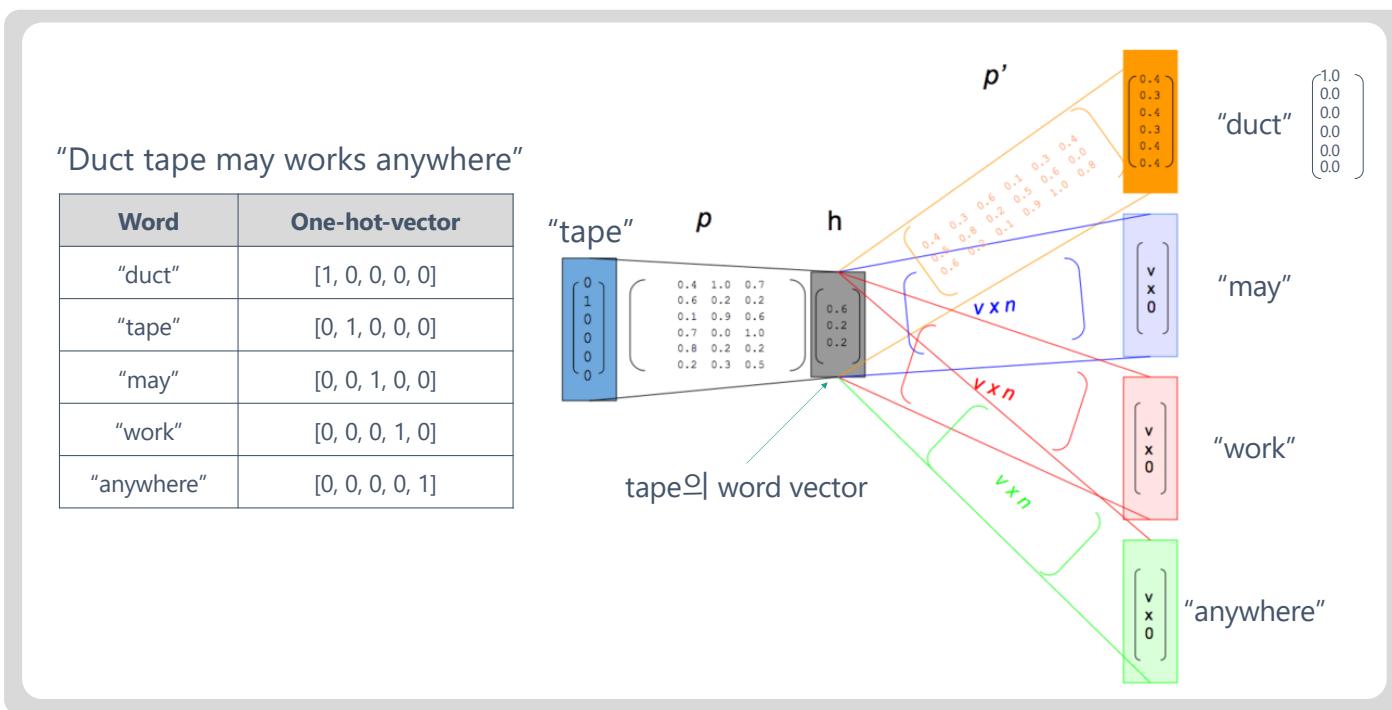


جو와 الكلب가 무슨 뜻인지는 모르겠지만, 주변 단어 형태가 비슷하니 **의미**도 비슷할 것이다!

Word embedding – Word2Vec

자연어 처리 (Natural Language Processing, NLP)

- Word2vec 알고리즘은 주변부의 단어를 예측하는 방식으로 학습 (Skip-gram 방식)
- 단어에 대한 dense vector를 얻을 수 있음



Word embedding – Word2Vec

자연어 처리 (Natural Language Processing, NLP)

- Word embedding의 방법론에 따른 특징

Sparse representation

- One-hot encoding
- n개의 단어에 대한 n차원의 벡터
- 단어가 커질 수록 무한대 차원의 벡터가 생성
- 주로 신경망의 입력단에 사용 (신경망이 임베딩 과정을 대체)
- 의미 유추 불가능
- 차원의 저주 (curse of dimensionality): 차원이 무한대로 커지면 정보 추출이 어려워짐
- One-hot vector의 차원 축소를 통해 특징을 분류하고자 하는 접근도 있음

Dense representation

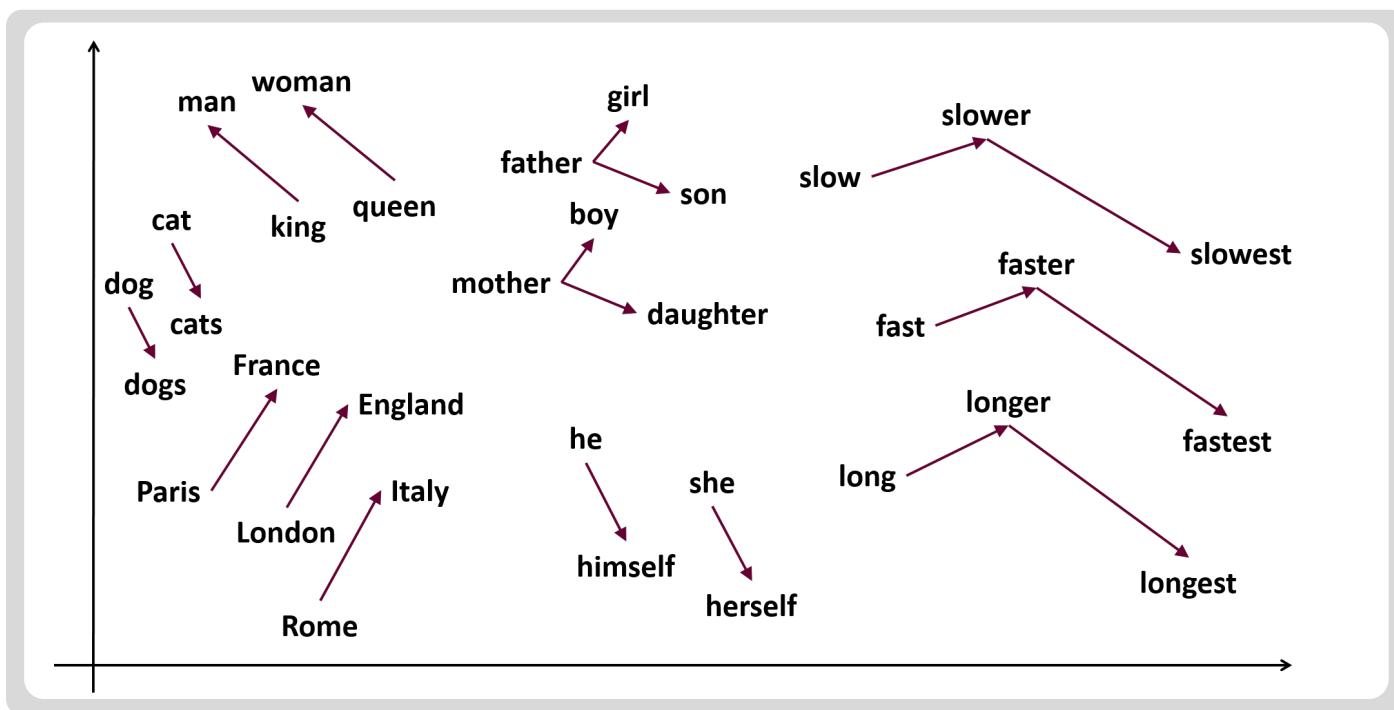
- Word embedding
- 한정된 차원으로 표현 가능
- 의미 관계 유추 가능
- 비지도 학습으로 단어의 의미 학습 가능

Word embedding – Word2Vec

자연어 처리 (Natural Language Processing, NLP)

- 단어의 의미가 벡터로 표현됨으로써 벡터 연산이 가능

$$\vec{w}_{king} - \vec{w}_{man} + \vec{w}_{woman} \approx \vec{w}_{queen}$$



Word embedding – Word2Vec

자연어 처리 (Natural Language Processing, NLP)

- Word embedding 성능 검증 방법
- (<https://github.com/dongjun-Lee/kor2vec>, <https://github.com/SungjoonPark/KoreanWordVectors>)

WordSim353

- 13-16명의 사람이 annotate한 두 단어의 유사도
- 두 단어 벡터의 cosine similarity를 구한 후 정답과 Spearman's rank-order correlation값을 획득
- 경험상 0.7 이상의 값이 나오면 embedding이 잘 수행 됐음

사랑	섹스	6.77
호랑이	고양이	7.35
호랑이	호랑이	10
책	종이	7.46
컴퓨터	키보드	7.62
컴퓨터	인터넷	7.58

⋮

Semantic/Syntactic analogy

- Semantic analogy

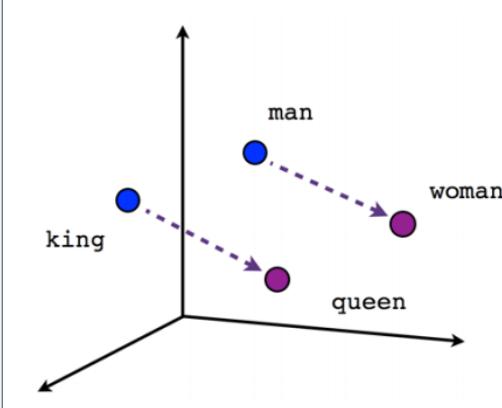
대한민국 – 서울 + 도쿄	일본
남동생 – 여동생 + 신부	신랑
왕 – 여왕 + 여동생	남동생
간디 – 인도 + 갈릴레이	이탈리아
베토벤 – 작곡가 + 단테	시인

- Syntactic analogy

밥 – 밥을 + 물을	물
여행이 – 여행은 + 흡연은	흡연이
보도했다 – 보도했습니다 + 알았습니다	알았다
마시다 – 마셨다 + 뽑습니다	뽑았습니다
오다 – 왔다 + 공부했다	공부하다

Word embedding – Word2Vec

자연어 처리 (Natural Language Processing, NLP)



- 단어가 가지는 의미 자체를 다차원 공간에 '벡터화'하는 것
- 중심 단어의 주변 단어들을 이용해 중심단어를 추론하는 방식으로 학습

장점

- 단어간의 유사도 측정에 용이
- 단어간의 관계 파악에 용이
- 벡터 연산을 통한 추론이 가능 (e.g. 한국 – 서울 + 도쿄 = ?)

단점

- 단어의 **subword information** 무시 (e.g. 서울 vs 서울시 vs 고양시)
- Out of vocabulary (OOV)에서 적용 불가능

Word embedding – FastText

자연어 처리 (Natural Language Processing, NLP)

- 한국어는 다양한 용언 형태를 가짐
- Word2Vec의 경우, 다양한 용언 표현들이 서로 독립된 vocab으로 관리

동사 원형: **모르다**

모르네	모르기까지	모르겠으나	몰라야	몰랐다면	몰랐겠으나
모르데	모르기를	모르겠으면	몰라요	몰랐다면	몰랐겠으면
모르지	모르기는	모르겠으면서	몰라라	몰랐을	몰랐겠으면서
모르더라	모르기도	모르겠거나	몰랐다	몰랐을까	몰랐겠거나
모르리라	모르기만	모르겠거든	몰랐네	몰랐을지	몰랐겠거든
모르는구나	모르는	모르겠는데	몰랐지	몰랐을지도	몰랐겠는데
모르잖아	모르던	모르겠지만	몰랐더라	몰랐어	몰랐겠지만
모르려나	모른	모르겠더라도	몰랐으리라	몰랐어도	몰랐겠더라도
모르니	모른다	모르겠다가도	몰랐구나	몰랐어야	몰랐겠다가도
모르고	모른다면	모르겠던	몰랐잖아	몰랐어요	몰랐겠던
모르나	모른다면	모르겠다면	몰랐으려나	몰랐더라면	몰랐겠다면
모르면	모른답시고	모르겠다만	몰랐으니	몰랐더라도	몰랐겠다만
모르면서	모르겠다	모를까	몰랐거나	몰랐겠다	몰랐겠어
모르거든	모르겠네	모를지	몰랐거든	몰랐겠네	몰랐겠어도
모르는데	모르겠지	모를지도	몰랐는데	몰랐겠지	몰랐겠어서
모르지만	모르겠더라	모를수록	몰랐지만	몰랐겠더라	몰랐겠어야
모르더라도	모르겠구나	몰라	몰랐더라도	몰랐겠구나	몰랐겠어요
모르다가도	모르겠니	몰라도	몰랐다가도	몰랐겠니	몰랐겠더라면
모르기조차	모르겠고	몰라서	몰랐던	몰랐겠고	몰랐겠더라도

Word embedding – FastText

자연어 처리 (Natural Language Processing, NLP)

Fasttext

- Facebook research에서 공개한 open source library (<https://research.fb.com/fasttext/>, fasttext.cc)
- C++11

Training

- 기존의 word2vec과 유사하나, 단어를 **n-gram**으로 나누어 학습을 수행
- n-gram**의 범위가 **2-5**일 때, 단어를 다음과 같이 분리하여 학습함
“**assumption**” = {as, ss, su,, ass, ssu, sum, ump, mpt,....., ption, **assumption**}
- 이 때, **n-gram**으로 나눠진 단어는 사전에 들어가지 않으며, 별도의 **n-gram vector**를 형성함

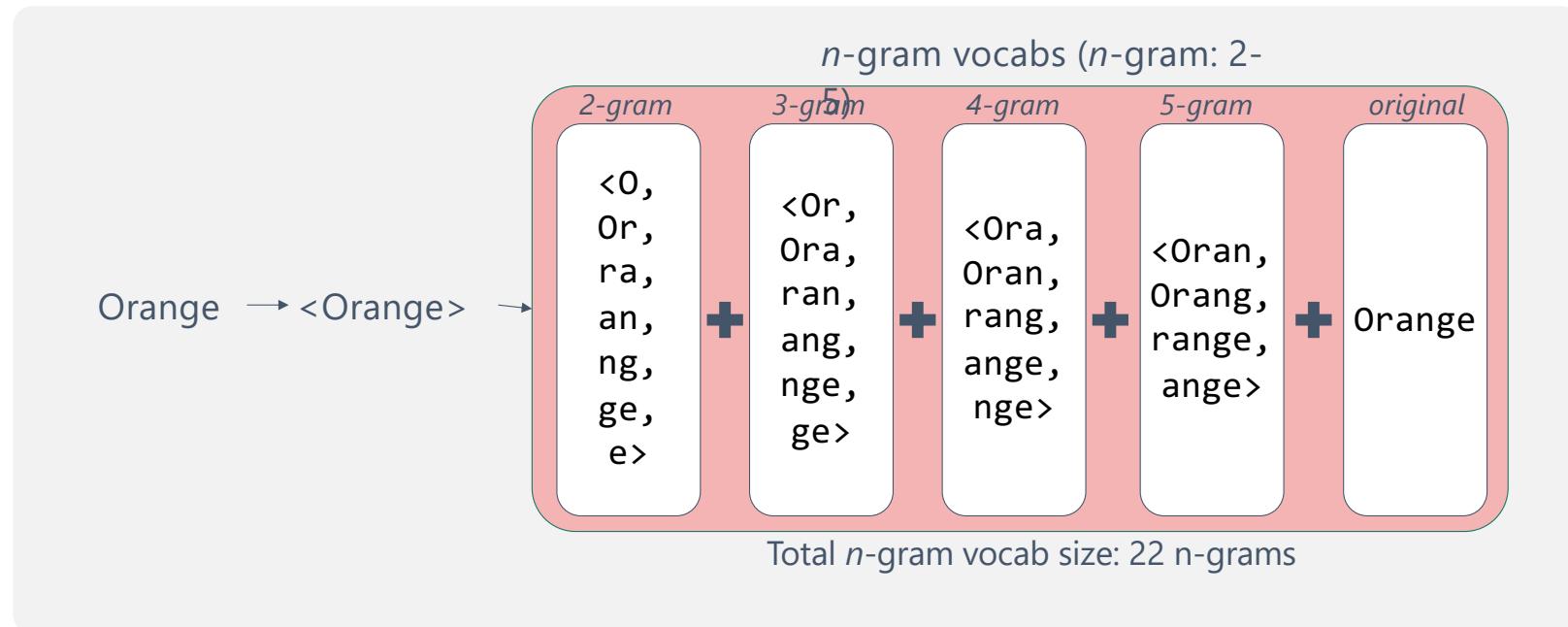
Testing

- 입력 단어가 vocabulary에 있을 경우, word2vec과 마찬가지로 해당 단어의 word vector를 return함
- 만약 OOV일 경우, 입력 단어의 n-gram vector들의 합산을 return함

Word embedding – FastText

자연어 처리 (Natural Language Processing, NLP)

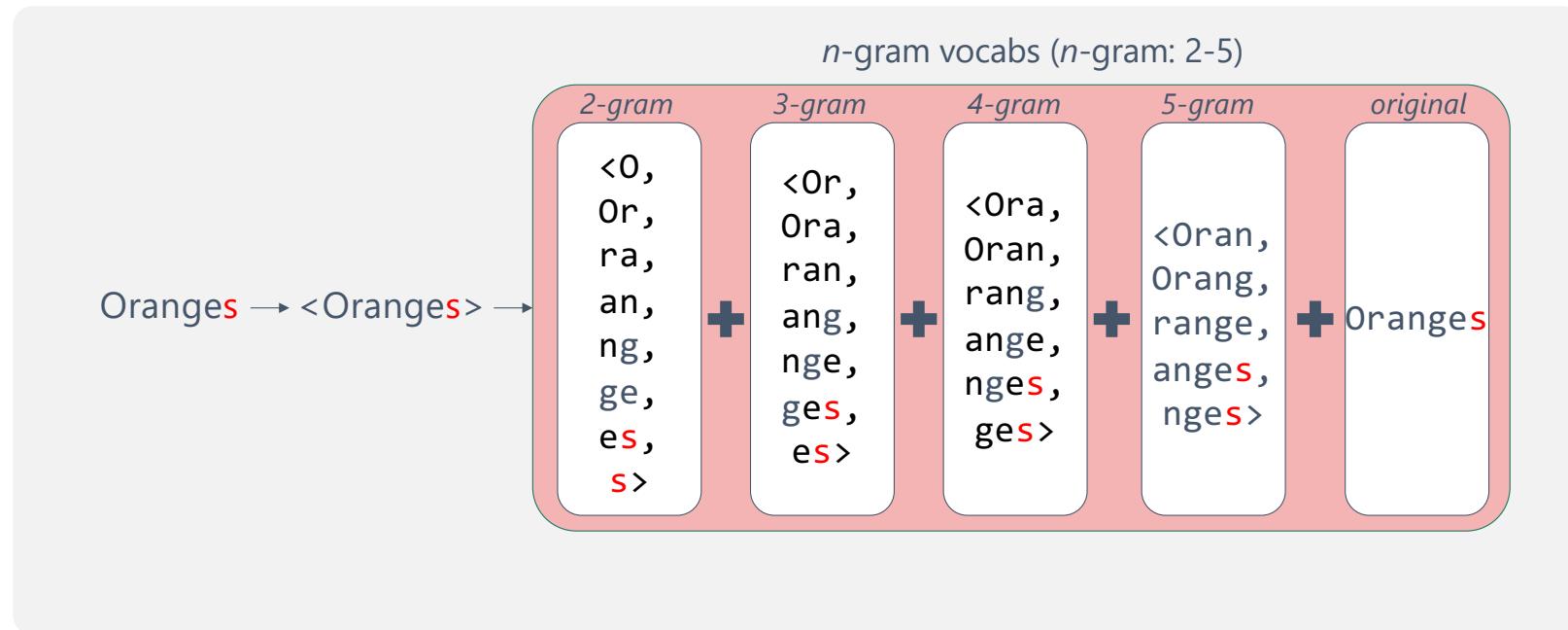
- FastText는 단어를 n -gram으로 분리를 한 후, 모든 n -gram vector를 합산한 후 평균을 통해 단어 벡터를 획득



Word embedding – FastText

자연어 처리 (Natural Language Processing, NLP)

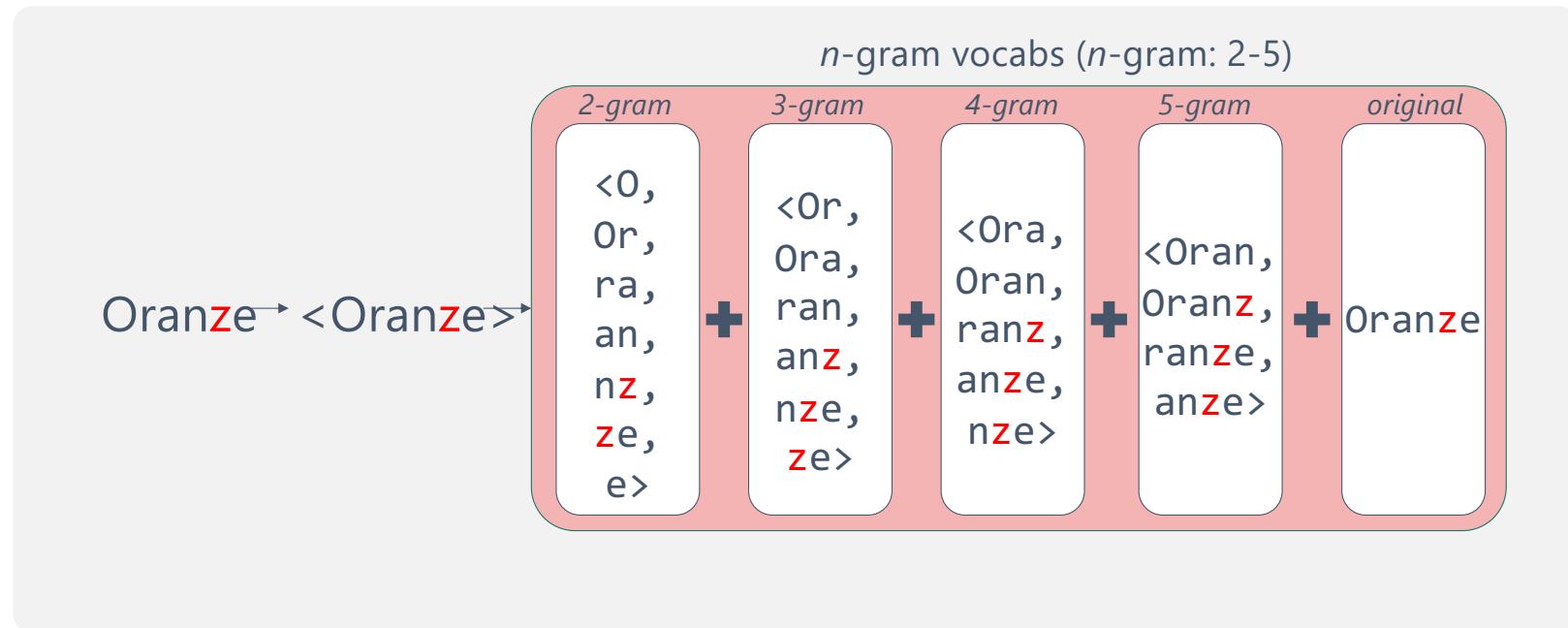
- FastText는 단어를 n -gram으로 분리를 한 후, 모든 n -gram vector를 합산한 후 평균을 통해 단어 벡터를 획득



Word embedding – FastText

자연어 처리 (Natural Language Processing, NLP)

- 오탈자 입력에 대해서도 본래 단어와 유사한 n -gram이 많아, 유사한 단어 벡터를 획득 가능



Other pre-trained word embeddings

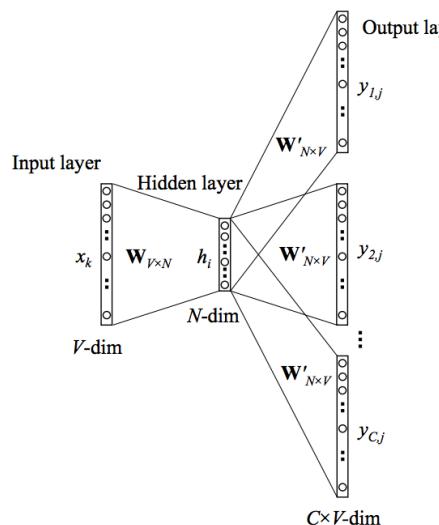
- GloVe
- <https://nlp.stanford.edu/projects/glove/>
- fastText
- <https://fasttext.cc/>

fastText in detail

Skip-gram을 이용하여 embedding matrix를 업데이트하며, 계산시간 감소를 위해 negative sampling을 사용하고, sub-word information을 활용하는 모델

Skip-gram

<http://i.imgur.com/TupGxMl.png>



Negative Sampling

<https://arxiv.org/pdf/1607.04606.pdf>

$$\sum_{t=1}^T \sum_{c \in \mathcal{C}_t} \log \left(\frac{e^{s(w_t, w_c)}}{\sum_{j=1}^W e^{s(w_t, j)}} \right)$$



$$\sum_{t=1}^T \left[\sum_{c \in \mathcal{C}_t} \log \left(1 + e^{-s(w_t, w_c)} \right) + \sum_{n \in \mathcal{N}_{t,c}} \log \left(1 + e^{s(w_t, n)} \right) \right]$$

where the function $s(\text{word}, \text{context})$ is a score function to \mathbb{R}

Sub-word Informations

<https://arxiv.org/pdf/1607.04606.pdf>

Ex. 3-gram with word *where*
<wh, whe, her, ere, er> + <where>

Usually, all 3 to 6-grams are used

$$s(w, c) = \sum_{g \in \mathcal{G}_w} z_g^T v_c$$

where $\mathcal{G}_w = (\text{set of } n\text{-grams appearing in } w)$

Out of Vocabulary Issue

- 유저의 질의가 학습 데이터만으로는 모두 대응될 수 없기 때문에, 추가적인 방법을 통해 유저 질의에 대한 커버리지를 향상시켜야 서비스에 적합한 수준을 달성할 수 있음

NE	Add to Data	Global Embedding
#체크카드 <ul style="list-style-type: none"> 체크카드 체카 첵카 라이언카드 직불카드 현금카드 	+ <p>체크카드 만드는 법 알려줘</p> <p>체카 만드는 법 알려줘</p> <p>첵카 만드는 법 알려줘</p> <p>라이언카드 만드는 법 알려줘</p> <p>직불카드 만드는 법 알려줘</p> <p>현금카드 만드는 법 알려줘</p>	<ul style="list-style-type: none"> 체카 <ul style="list-style-type: none"> 체크카드 현금카드 직불카드 신용카드

Out of Vocabulary Issue

생성된 코퍼스 데이터만 사용하여 학습한 local embedding model은 코퍼스 외부에 존재하는 단어에는 대응할 수 없는 문제가 있음

“체크카드 만드는 법 알려줘” 는 답변이

나오는데,

“체카 만드는 법 알려줘” 는 답이 안나와요

Advanced FastText 모델

자연어 처리 (Natural Language Processing, NLP)

- 오탈자, OOV, 등장 회수가 적은 학습 단어에 대해서 강세

(https://link.springer.com/chapter/10.1007/978-3-030-12385-7_3, <https://github.com/MrBananaHuman/JamoFastText>)

Input word	FastText Model	Most similar words in range of top 3		
페널티 (Penalty) OOV word	Baseline	리날디 (Rinaldi)	페레티 (Ferretti)	마세티 (Machete)
	Jamo-advanced	페널티골 (Penalty goal)	페널티 (Penalty)	드록바 (Drogba)
나포탈렌 (Naphthalene) OOV word	Baseline	야렌 (Yaren)	콜루바라 (Kolubara)	몽클로아아라바카 (Moncloa-Aravaca)
	Jamo-advanced	나포탈렌 (Naphthalene)	테레프탈산 (Terephthalic acid)	아디프산 (Adipic acid)
스테이크 (Steak) OOV word	Baseline	스태너프 (Stanhope)	스탠너드 (Stannard)	화이트스네이크 (White Snake)
	Jamo-advanced	롱테이크 (Long take)	비프스테이크 (Beefsteak)	스테이크 (Steak)

FastText와 Word2Vec의 비교

자연어 처리 (Natural Language Processing, NLP)

- 오탈자, OOV, 등장 회수가 적은 학습 단어에 대해서 강세

(https://link.springer.com/chapter/10.1007/978-3-030-12385-7_3, <https://github.com/MrBananaHuman/JamoFastText>)

Input word	DF	Model	Most similar (1)	Most similar (2)	Most similar (3)	Most similar (4)	Most similar (5)
파일	10146	Word2vec	프로토콜	애플리케이션	url	디렉터리	포맷
		Fasttext	파일	확장자	음악파일	포맷	디렉터리
원인	11589	Word2vec	부작용	현상	요인	증상	질병
		Fasttext	주원인	초래	요인	직접	주요인
태생지	8	Word2vec	부타	구티에레즈	보아뱀	올림피코	집사장
		Fasttext	생지	탄생지	출생지	발생지	무생지
미스코리아	11	Word2vec	치평요람	神檀實記	컬투	방학기	김현승
		Fasttext	믹스코리아	라이코스코리아	악스코리아	보이스코리아	포브스코리아

- Document frequency (DF) 가 낮을 경우, **word2vec**은 유의미한 단어를 찾을 수 없음
- **Fasttext**의 경우, **subword information**이 유사한 단어를 찾았음

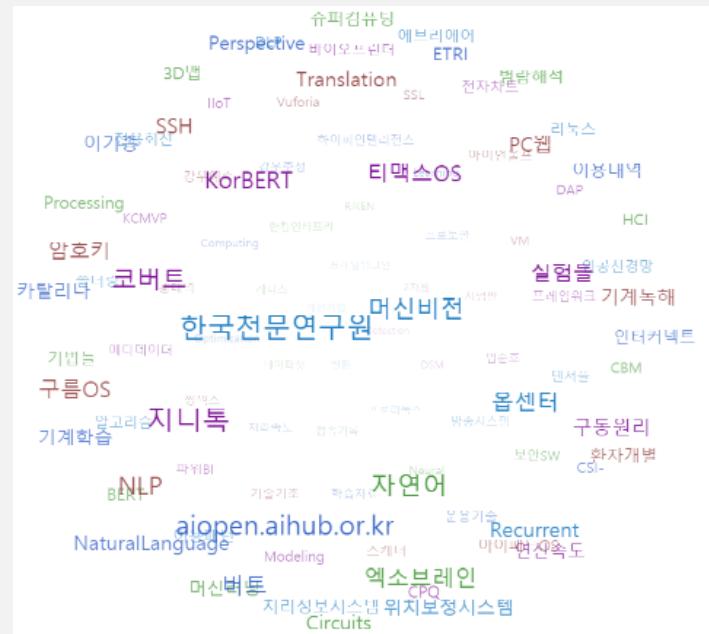
Word embedding의 활용

자연어 처리 (Natural Language Processing, NLP)

- 다른 자연어처리 모델의 입력으로 사용
(e.g. 학습 데이터의 양이 매우 적은 감성분석을 수행할 경우, 학습 데이터만으로는 특성 추출 불가능)
 - 토픽 키워드 (<https://www.adams.ai/apiPage?deeptopicrank>)



BERT 관련 토픽 키워드 추출



Word embedding 방식의 한계점

자연어 처리 (Natural Language Processing, NLP)

- Word2Vec이나 FastText와 같은 word embedding 방식은 동형어, 다의어 등에 대해선 embedding 성능이 좋지 못하다는 단점이 있음
- 주변 단어를 통해 학습이 이루어지기 때문에, ‘문맥’을 고려할 수 없음

Account

- I **account** him to be my friend ~라고 생각하다
- He is angry on **account** of being excluded from the invitation 이유, 근거
- I have an **account** with the First National Bank ~때문에
- The police wrote an **account** of the accident 보고서
- Your check has been properly credited, and your **account** is now full? 계좌