

Unsupervised Word Extraction & Tokenization

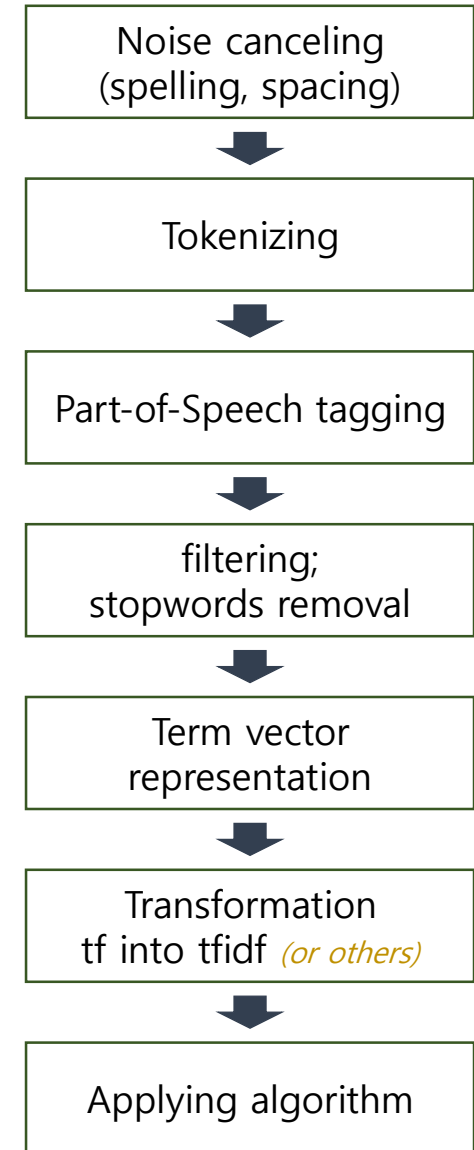
Hyunjoong Kim

soy.lovit@gmail.com

github.com/lovit

Framework

- 한국어 텍스트의 토큰나이징 / 품사 판별을 위하여 말뭉치 기반으로 학습된 모델을 이용할 수 있습니다.
- 미등록단어 문제가 발생할 수 있습니다.
- 분석의 주요 단어가 제대로 인식되지 않으면 키워드 추출이나 토픽 모델링의 품질이 저하됩니다.



Out of vocabulary

- 새롭게 만들어진 단어들은 잘 인식되지 않습니다.

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

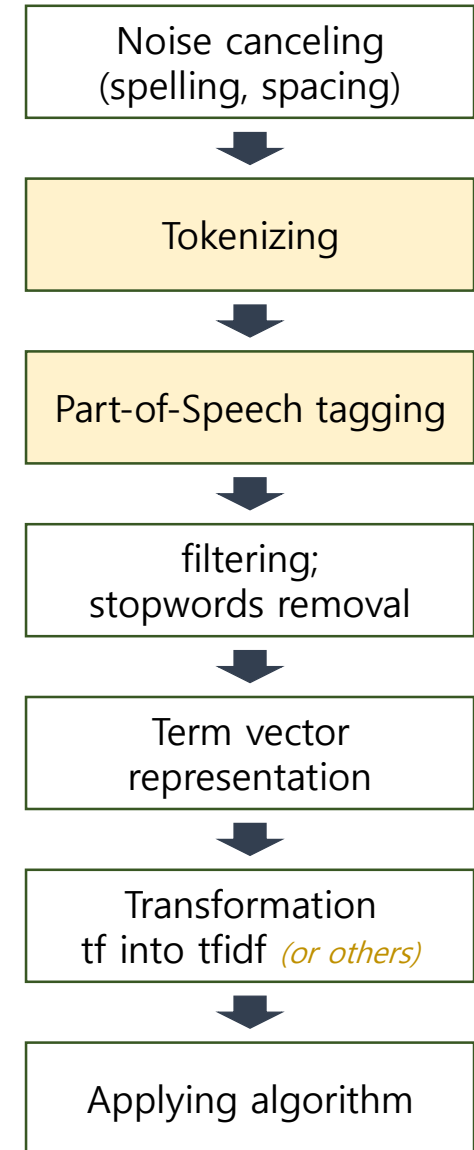
너무/MAG, 너무너무/MAG, 는/JX, 아이오/NNG, 아이/NNG, 의/JKG, 노래/NNG, 예/JKM, 요/JX

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

너무/Noun, 너무/Noun, 너무/Noun, 는/Josa, 아이오/Noun, 아이/Noun, 의/Josa, 노래/Noun,
예요/Josa

Framework

- 수작업으로 사용자 사전을 구축하여 기학습된 모델에 추가할 수 있습니다.
 - 단어 추출 기법을 통하여 사전 구축 비용을 줄입니다.
- 토픽 모델링 / 키워드 추출을 위해서는 명사만 이용하기도 합니다.
 - 명사는 다른 품사의 단어보다도 추출이 쉽습니다.



Unsupervised Word Extraction

한국어의 특성 관찰

- **관찰 1.** 의미를 지니는 단어는 어절의 왼쪽에 등장합니다.

- 명사, 동사, 형용사, 부사, 감탄사 : 의미를 지니는 단어
- 조사, 어미 : 문법 기능의 단어와 형태소

- (예시)

- 발표/명사 + 를/조사
- 하/동사어근 + 면서/어미

한국어의 특성 관찰

- **관찰 2.** 어절의 다양성은 문법 기능을 하는 단어에 의하여 일어납니다.
 - 문법 기능을 하는 단어를 어절에서 분리하면 분리된 단어의 종류는 작습니다.
 - 새롭게 만들어지는 단어는 주로 의미를 지니는 부분입니다.
- (예시)
 - [명사 + 조사]: **발표**+를, **발표**+에서, **발표**+도, ...
 - [동사/형용사 + 어미]: **하**+면서, **하**+고, **하**+니까

한국어의 특성 관찰

- **관찰 3.** 한국어 어절의 형태는 $L + [R]$ 입니다.
- 복합형태소는 하나의 R 로 생각하면 어절 구조가 단순해집니다.
- 형태소 수준의 분석까지 필요하지 않기도 합니다.
- 수업하는데 = 수업_{/명사} + 하_{/동사파생접미사} + 는데_{/어미}
- 수업하는데 = 수업_{/L} + 하는데_{/R}

한국어의 특성 관찰

- 단어 추출은 어절에서 의미를 지니는 부분인 L을 인식하는 것이며,
- 토큰나이징은 어절을 $L + [R]$ 로 나누는 것으로 생각할 수 있습니다.

단어 인식과 문서 벡터 표현

- 분석 대상이 문서라면 단어를 제대로 인식하지 않아도 됩니다.
- **키워드 추출 / 토픽 추출**은 **분석의 대상이 단어**이기 때문에 토큰라이저가 올바른 단어 인식을 해야 합니다.
- **문서 판별 / 문서 군집화**는 **문서가 벡터로 잘 표현되는지**가 중요합니다.
 - term frequency vector나 doc2vec 등을 이용하여 문서를 벡터로 표현하면
 - 이를 이용하여 판별이나 군집화를 수행합니다.

부분어절을 이용한 문서 표현

- 어절의 왼쪽에 위치한 subwords 만 이용하여도 좋은 term frequency vector 를 만들 수 있습니다.



Term vector representation (Nouns and verbs using Kkma)

[이=18, **태환=16**, **박=15**, 원장=14, 하=14, 호르몬=13, 김=13, **박태=13**, 주사=11, 남성=11, 월=11, 받=10, 말하=10, 누나=10, 것=10, 병원=9, 고=8, 작년=7, 측=6, 대하=6, 맞=6, **환의=6**, **환=6**, 있=6]

Pseudo term vector representation (2-syllables)

[**박태=28**, 원장=14, 주사=11, 남성=10, 말했=10, 누나=10, 병원=9, 작년=7, 대하=6, 검찰=5, 간호=5, 문제=5, 했다=4, 측=4, 몰랐=4, 알려=4, 밝혔=4, 질문=4, 도핑=4, 다른=3, 없다=3, 주치=3, 있다=3, 월에=3, 성분=3, 받았=3, 측이=3, 받는=3, 그런=3, 치료=3, 것으=3, 청문=3, 맞고=3, 모두=3, 되풀=3, 회원=3, 생각=3]

Pseudo term vector representation (3-syllables)

[**박태환=28**, 말했다=10, 남성은=10, 원장은=9, 간호사=5, 밝혔=4, 누나가=4, 것으로=3, 누나는=3, 되풀이=3, 검찰에=3, 주치의=3, 청문회=3, 문제없=3, 병원에=3, 수차례=2, 것처럼=2, 주사를=2, 있다고=2, 대답했=2, 호르몬=2, 그리고=2, 마찬가지로=2, 알려졌=2, 고소하=2, 운동하=2, 프로그=2, 통보를=2, 치료를=2, 병원을=2, 질문을=2, 회원들=2, 내용을=2, 양성반=2, 소속사=2, 몰랐던=2, 변호사=2, 이야기=2, 했다고=2, 없다고=2, 받았=2]

부분어절을 이용한 문서 표현

- 어절의 왼쪽부터 k 개의 음절만 취하여 만든 term frequency vector 도 해석이 가능합니다.
 - 많은 단어들이 2 ~ 3 음절이며,
 - 더 긴 단어라 하더라도 부분단어만으로 전체 단어가 예상되기 때문입니다.
 - 사람이 해석 가능한 벡터라면 머신 러닝 알고리즘 역시 유용하게 이용할 수 있습니다.

부분어절을 이용한 문서 표현

- 단어의 부분어절이 단어와 출현 빈도수가 같을 경우, 단어를 부분어절로 대체하여도 동일한 벡터를 얻습니다.



부분어절을 이용한 문서 표현

- 토큰나이징에서 중요한 점은 “하나의 개념이 하나의 feature 로 표현”하는 것입니다. 일관성이 있어야 합니다.

최순실과 ==> [('최', 'NNP'), ('순', 'NNG'), ('실과', 'NNG')]

최순실은 ==> [('최', 'NNP'), ('순', 'NNG'), ('실', 'NNG'), ('은', 'JX')]

최순실 ==> [('최', 'XPN'), ('순실', 'XR')]

부분어절을 이용한 문서 표현

- 하루의 뉴스 기사를 어절 왼쪽의 3 음절을 단어로 취한 뒤, cosine distance 를 이용하여 유사 문서를 검색하였습니다.
- 부분음절을 단어로 이용하여도 문서 군집화는 잘 학습됩니다.

Query doc

[('방탄소', 5),
('엠카운', 4),
('뉴스1', 2),
('자랑했', 2),
('출연했', 2),
('20일', 2),
('유수경', 1),
('노래다', 1),
('열창했', 1),
...]

Similar docs

doc id = 28947, cosine-dist = 0.334

몬스타 엠카운 20일 뉴스1 넘치는 다비치 레이디 맨스에 멤버들 무대를
무대에 방송됐 방송된 방출했 방탄소 백퍼센 빅브레 샤이니 선보였 선보이
신용재 아이오 에이핑크 열창했 오블리 유수경 의상을 재배포 출연했 카리스

doc id = 16492, cosine-dist = 0.354

방탄소 다비치 엠카운 레이디 몬스타 샤이니 스포츠 아이오 에이핑크 올랐다
10위 10일 1위에 20일 2관왕 5위를 감사를 갓세븐 걸그룹 공약으
공약했 귀여운 귀요미 그대인 김영록 누르고 두번째 드러넛 매력을 맨스에

doc id = 28966, cosine-dist = 0.373

아이오 엠카운 20일 뉴스1 출연했 과시했 귀여운 깜찍한 너무너 다비치
뒤흔들 레이디 마음을 매력을 맨스에 몬스타 몸짓이 무대를 물오른 미모와
방송된 방탄소 백퍼센 빅브레 상큼함 샤이니 선보였 신용재 에이핑크 오블리

doc id = 28931, cosine-dist = 0.375

엠카운 20일 뉴스1 세븐은 세븐이 8개월 꾸몄다 노련미 녹슬지 다비치
레이디 매너와 맨스에 몬스타 무대를 발표한 방송됐 방송된 방탄소 백퍼센
빅브레 샤이니 선보였 성숙한 신용재 실력을 아이오 에이핑크 오블리 유수경

단어 추출

- 새롭게 만들어지는 단어(정확히는 형태소)는 **명사**와 **어미**입니다
 - 명사는 **새로운 개념**을 표현하기 위해서, (eg: 아이오아이)
 - 어미는 다양한 말투를 위해서 만들어집니다 (eg: 하지**말라**궁)
 - 어미에 의하여 새로운 동사/형용사가 만들어집니다

단어 추출

- 새롭게 만들어지는 단어(정확히는 형태소)는 **명사**와 **어미**입니다
- 새로운 형용사/동사의 어근은 “명사 + 이다/되다/하다”의 결합이 많습니다
(eg: 덕질/명사 + 하다/동사)

Unsupervised Word Extraction

Cohesion score

단어 추출

- 다양한 비지도학습 기반의 단어 추출 방법이 제안되었습니다
 - 그 중 character n-gram을 이용하여 단어를 추출하는 방법을 이용합니다

Cohesion (Character n-gram)

- 한국어의 의미를 지니는 단어 (명사/동사/형용사/부사)는 **어절 왼쪽**에 있습니다


짜장면/명사 + **을**/조사

먹/동사 + **었어**/어미

Cohesion (Character n-gram)

- 맥락이 충분히 주어지지 않으면 다음에 등장할 글자의 확률이 작습니다
 - 한글자 ('아')는 매우 모호한 문맥입니다

아이



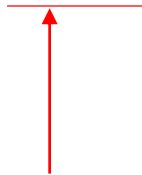
한글자는 특별한 문맥을
가지기가 어렵습니다

- > 아니 17.15 %
- > **아이 14.86 %**
- > 아시 8.06 %
- > 아닌 4.74 %
- > 아파 4.43 %
- > 아직 3.85 %
- ...

Cohesion (Character n-gram)

- 맥락이 충분히 주어지지 않으면 다음에 등장할 글자의 확률이 작습니다

아이오



어떤 경우는 두 글자라 하더라도
다양한 맥락에서 등장하기도 합니다

- > 아이폰 16.60 %
- > 아이들 13.37 %
- > 아이디 9.66 %
- > 아이돌 6.77 %
- > 아이뉴 6.77 %
- > **아이오 6.53 %**

...

Cohesion (Character n-gram)

- Subword 다음에 등장할 글자가 쉽게 예상된다면 (확률이 높다면) 아직 단어가 끝나지 않았다는 의미입니다

아이오아

문맥이 명확해 질수록
이전 단어 → 다음 글자 확률이
높아집니다

> **아이오아 87.95 %**

> 아이오닉 7.49 %

> 아이오와 3.26 %

> 아이오빈 0.65 %

> 아이오페 0.33 %

> 아이오케 0.33 %

Cohesion (Character n-gram)

- Subword 다음에 등장할 글자가 쉽게 예상된다면 (확률이 높다면) 아직 단어가 끝나지 않았다는 의미입니다

> 아이오아이 100.00 %

아이오아이

문맥이 확실하면 다음글자의
등장 확률이 높습니다

Cohesion (Character n-gram)

- 단어의 경계를 넘으면 다음 글자에 대한 확률이 다시 작아집니다

아이오아이는

단어 경계 뒤에는 다양한
조사/어미 들이 등장합니다

- > 아이오아이의 31.97 %
- > **아이오아이는 27.21 %**
- > 아이오아이와 13.61 %
- > 아이오아이가 12.24 %
- > 아이오아이에 9.52 %
- > 아이오아이까 1.36 %

...

Cohesion (Character n-gram)

- 단어의 점수(cohesion)를 아래처럼 정의해 봅니다

$$cohesion(c_{1:n}) = \sqrt[n-1]{\prod_{i=1}^{n-1} P(c_{1:i+1} | c_{1:i})}$$

$$P(c_{1:2} | c_1) = \frac{\#c_{1:2}}{\#c_1}$$

$$\begin{aligned} cohesion('아이오아이') = & \{ p(\text{아} \rightarrow \text{아이}) * \\ & p(\text{아이} \rightarrow \text{아이오}) * \\ & p(\text{아이오} \rightarrow \text{아이오아}) * \\ & p(\text{아이오아} \rightarrow \text{아이오아이}) \\ & \}^{1/(5-1)} \end{aligned}$$

학습은 오로지
string count

Cohesion (Character n-gram)

- 하루치 뉴스로부터 학습한 결과입니다

subword	frequency	$P(AB \mid A)$	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

Tokenizer

- 단어를 잘 인식할 수 있다면 토크나이징도 쉽게 할 수 있습니다
 - 토크나이징은 여러 개의 단어로 이뤄진 문장/어절에서 단어를 구분하는 것
- 데이터의 띄어쓰기 품질에 따라 다른 토크나이징 전략을 사용할 수 있습니다

L-Tokenizer

- 띄어쓰기가 잘 되어 있다면, 어절의 **왼쪽에서**부터 단어의 **점수가 가장 큰 subword**를 기준으로 어절을 나눕니다

```
def ltokenize(w):  
    n = len(w)  
    if n <= 2: return (w, '')  
    tokens = []  
    for e in range(2, n+1):  
        tokens.append(w[:e], w[e:], cohesion(w[:e]))  
    tokens = sorted(tokens, key=lambda x:-x[2])  
    return tokens[0][:2]
```

```
sent = '뉴스의 기사를 이용했던 예시입니다'  
for word in sent.split():  
    print( ltokenize(word) )
```

('뉴스', '의')

('기사', '를')

('이용', '했던')

('예시', '입니다')

L-Tokenizer

꼬꼬마
형태소 분석기

(명사, 동사)

[서부/NNG=3.0, 일/NNG=3.0, **제주도/NNP=3.0**, **풍랑/NNG=3.0**, 기상청/NNG=3.0,
먼바다/NNG=3.0, 주의/NNG=2.0, 뉴스/NNG=2.0, 시/NNG=2.0, 불고/NNG=2.0,
남해/NNG=2.0, 전/NNG=2.0, **연합/NNG=2.0**, 일/VV=2.0, 제주/NNG=2.0,
기하/VV=2.0, 높/VA=1.0, 지방/NNG=1.0, 기자/NNG=1.0, 예/NNG=1.0, 앞바다/NNG=1.0,
낮/NNG=1.0, 서쪽/NNG=1.0...]

Cohesion
+ L-tokenizer

[**제주=5.0**, 기해=2.0, **풍랑주의보=2.0**, 일겠다=2.0, 먼바다에=2.0, 시를=2.0,
남해=2.0, 서부=2.0, 불고=2.0, 해제=1.0, 높이=1.0, 이날=1.0, 기자=1.0, 전망이다
=1.0, 저작권자=1.0, 오전=1.0, 서쪽=1.0, 항해=1.0, 당부했다=1.0, 이보다=1.0, 무
단=1.0, 물결=1.0, 파도가=1.0, 앞서=1.0, 전지혜=1.0, 강하게=1.0, 서풍=1.0,
연합뉴스=1.0, ...]

L-Tokenizer

꼬꼬마
형태소 분석기
(명사, 동사)

[**지리/NNG=11.0**, **나이/NNG=11.0**, **보/NNG=7.0**, **코/NNG=7.0**, 일/NNG=6.0,
카메룬/NNG=6.0, 시위/NNG=5.0, 대통령/NNG=5.0, 우리/NP=4.0, 알/VV=4.0,
하/VV=4.0, 말하/VV=4.0, 대하/VV=4.0, 바/NNG=4.0, 연합/NNG=3.0, 수천/NNG=3.0,
아프리카/NNG=3.0, 국/NNG=3.0, 명의/NNG=3.0, 벌이/VV=3.0, 군중/NNG=3.0,
시/NNG=3.0, 작전/NNG=3.0, 이날/NNG=3.0...]

Cohesion
+ L-tokenizer

[**나이지리아=10.0**, **보코하람=8.0**, **차드=8.0**, **카메=6.0**, 있는=6.0, 시위=5.0,
대통령=5.0, 있다=4.0, 우리=4.0, 말했다=4.0, 대한=3.0, 바가=3.0, 수천=3.0, 데비
=3.0, 수도=3.0, 명의=3.0, 이날=3.0, 그러나=2.0, 정작=2.0, 지난=2.0, 없다=2.0,
은자=2.0, 작전=2.0, 시큰=2.0, 보도했다=2.0, 일현지시간=2.0, 공동=2.0, 이슬람
=2.0, 통신=2.0, 모여=2.0, 군중이=2.0, 지지=2.0, 출발=2.0, 무관=2.0, 파병=2.0,
대해=2.0, 전쟁=2.0, 국경=2.0, 위해=2.0, 군사=2.0, 아프리카=2.0 ...]

Cohesion (Character n-gram)

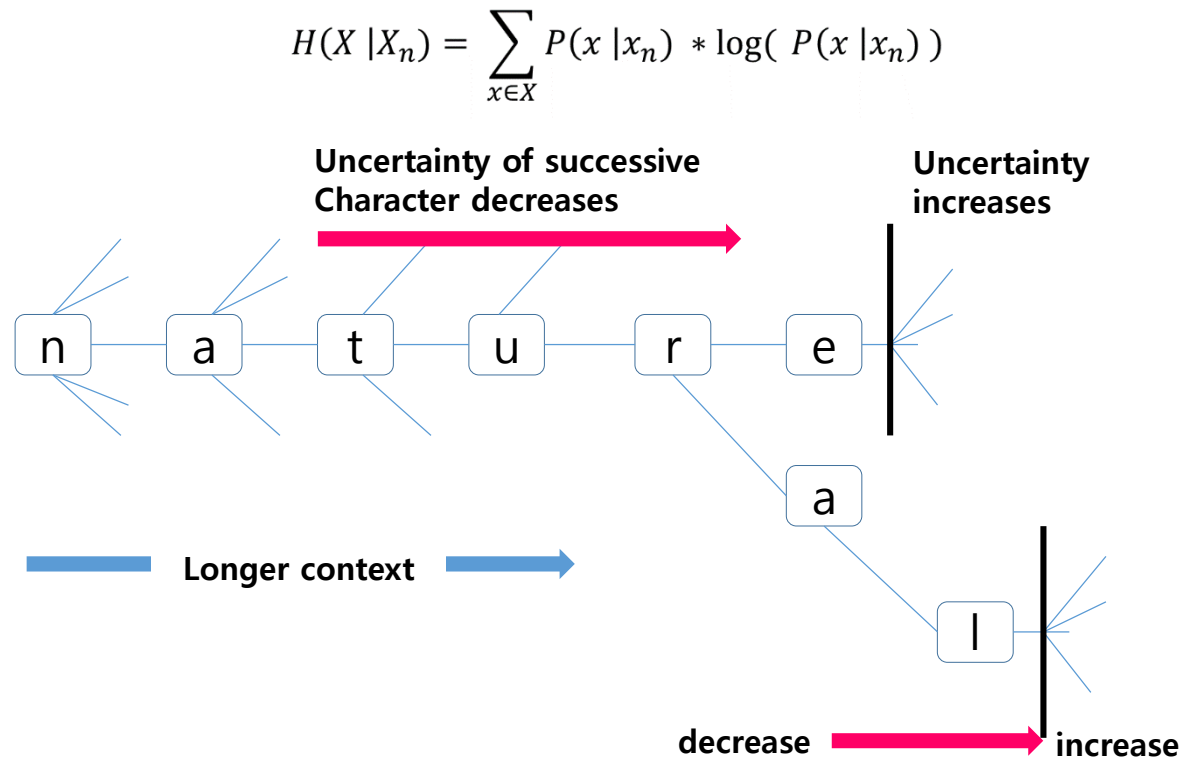
- Cohesion 은 문서 집합에서 자주 등장한 단어들을 잘 추출합니다.
 - 자주 등장한 단어는 문서 집합 내에서 중요한 단어일 가능성이 높습니다.
 - 자주 등장하지 않았던 단어들은 vectorising 과정에서 min count 에 의하여 버려질 가능성도 높습니다.
- 수작업을 하지 않으면서도 질 좋은 features 의 term frequency vector 를 만들 수 있습니다.

Unsupervised Word Extraction

Branching Entropy and Accessor Variety

Branching Entropy

- 단어의 경계 부분에서는 다음 글자의 불확실성이 증가합니다.
 - 연속된 글자의 각 부분에서 다음 글자의 불확실성을 entropy 로 정의합니다



Branching Entropy

- Entropy 는 확률 분포의 불확실성을 정의하는 방법입니다.
- Prob: {a: 0.99, b: 0.005, c: 0.005} 에서 임의의 한 개를 선택했을 때 대부분 a 입니다. 확실합니다.
- Entropy = - { 0.99 * log(0.99) + 0.005 * log(0.005) + 0.005 * log(0.005) }
= 0.063

Branching Entropy

- Entropy 는 확률 분포의 불확실성을 정의하는 방법입니다.
- Prob: {a: 0.3, b: 0.4, c: 0.3} 에서 임의의 한 개를 선택하면 어떤 글자가 등장할지 예상하기 어렵습니다. 불확실성이 큼니다.
- Entropy = - { 0.3 * log(0.3) + 0.4 * log(0.4) + 0.3 * log(0.3) }
= 1.089

Branching Entropy

- 불확실성으로 단어의 경계를 표현할 수도 있습니다
 - 불확실성은 **Entropy**를 이용하여 수치화 할 수 있습니다

아 ?

↑

'아' 다음에 올 수 있는
글자는 매우 많습니다

> 아니 17.15 %
> 아이 14.86 %
> 아시 8.06 %
> 아닌 4.74 %
> 아파 4.43 %
> 아직 3.85 %
...

H = 3.43

Branching Entropy

- 불확실성으로 단어의 경계를 표현할 수도 있습니다

아이 ?

- > 아이폰 16.60 %
- > 아이들 13.37 %
- > 아이디 9.66 %
- > 아이돌 6.77 %
- > 아이뉴 6.77 %
- > 아이오 6.53 %
- ...

H = 3.11

Branching Entropy

- 불확실성으로 단어의 경계를 표현할 수도 있습니다

아이오 ?

문맥이 명확해 질수록
다음 글자를 예상할 수 있습니다
(= 불확실성이 줄어듭니다)

- > 아이오아 87.95 %
 - > 아이오닉 7.49 %
 - > 아이오와 3.26 %
 - > 아이오빈 0.65 %
 - > 아이오페 0.33 %
 - > 아이오케 0.33 %
- H = 0.49**

Branching Entropy

- 불확실성으로 단어의 경계를 표현할 수도 있습니다

> 아이오아이 100.00 %

아이오아 ?

오로지 한 가지 경우만 가능하면
불확실성 (entropy)는 0 입니다

Branching Entropy

- 단어의 경계를 넘으면 다음 글자에 대한 불확실성이 다시 커집니다

아이오아이 ?

- > 아이오아이의 31.97 %
- > 아이오아이는 27.21 %
- > 아이오아이와 13.61 %
- > 아이오아이가 12.24 %
- > 아이오아이에 9.52 %
- > 아이오아이까 1.36 %
- ...

H = 1.72

Branching Entropy

- 하루치 뉴스로부터 학습한 결과입니다
 - '아이'가 다양한 단어의 subword 여서 Branching Entropy가 큼니다

subword	frequency	P(AB A)	Cohesion score
아이	4,910	0.15	3.11
아이오	307	0.10	0.49
아이오아	270	0.20	0
아이오아이	270	0.30	1.72
아이오아이는	40	0.26	0

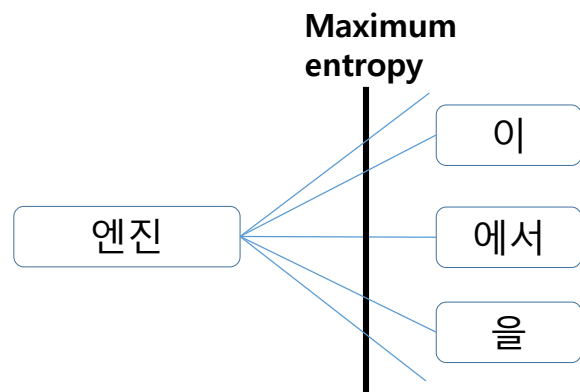
Branching Entropy

- Cohesion, Branching Entropy 를 조합하여 다양한 토큰나이지어를 만듭니다.
 - (예시) Branching Entropy가 다음 글자에서 떨어지는 부분들 중에서 Cohesion이 가장 큰 부분 선택

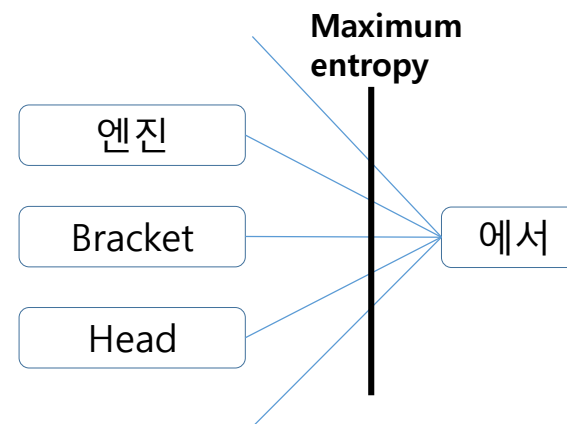
subword	frequency	Cohesion score	Branching Entropy
아이	4,910	0.15	3.11
아이오	307	0.10	0.49
아이오아	270	0.20	0
아이오아이	270	0.30	1.72
아이오아이는	40	0.26	0

Branching Entropy

- 어절의 양방향 접근이 가능하기 때문에, 단어의 왼쪽 경계도 찾습니다



Right-side branch entropy

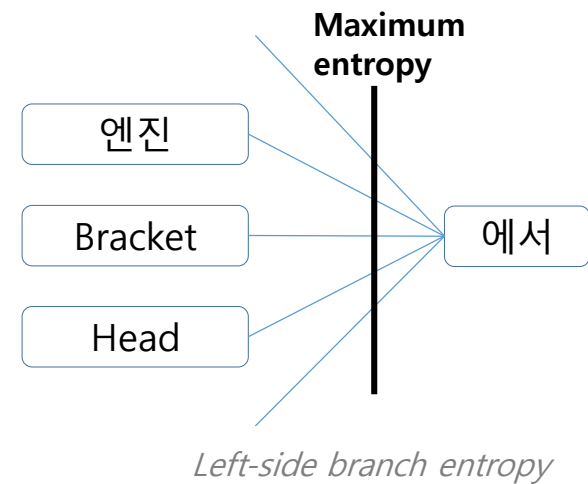
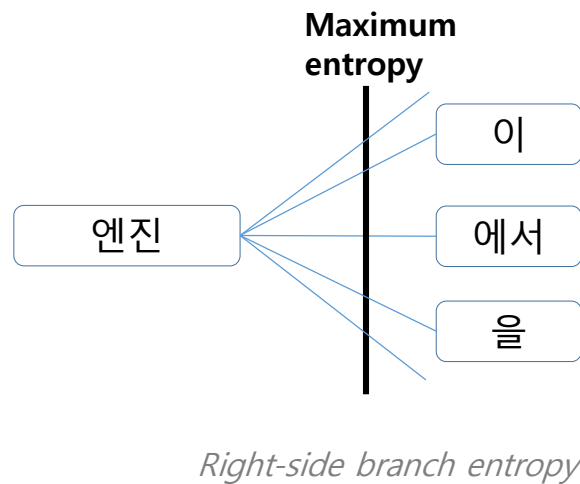


Left-side branch entropy

엔진에서 → 엔진 + 에서

Branching Entropy

- Accessor Variety는 경계에 등장하는 글자의 종류수로 단어 점수를 표현
 - AV 와 BE는 중국어/일본어의 word segmentation에서 자주 이용되었음



엔진에서 → 엔진 + 에서

단어 추출

- 통계 기반 단어 추출 방법은 exterior / interior score 로 분류합니다.
 - Cohesion 은 단어 내 글자의 연관성을 단어 점수로 이용합니다. (interior)
 - Branching Entropy 는 단어 좌/우의 다른 글자를 이용하여 단어 점수를 정의합니다. (exterior)
 - 두 방법은 더 서로 다른 종류의 정보를 이용합니다. 어떤 방법이 우수한 것이 아닙니다.

단어 추출

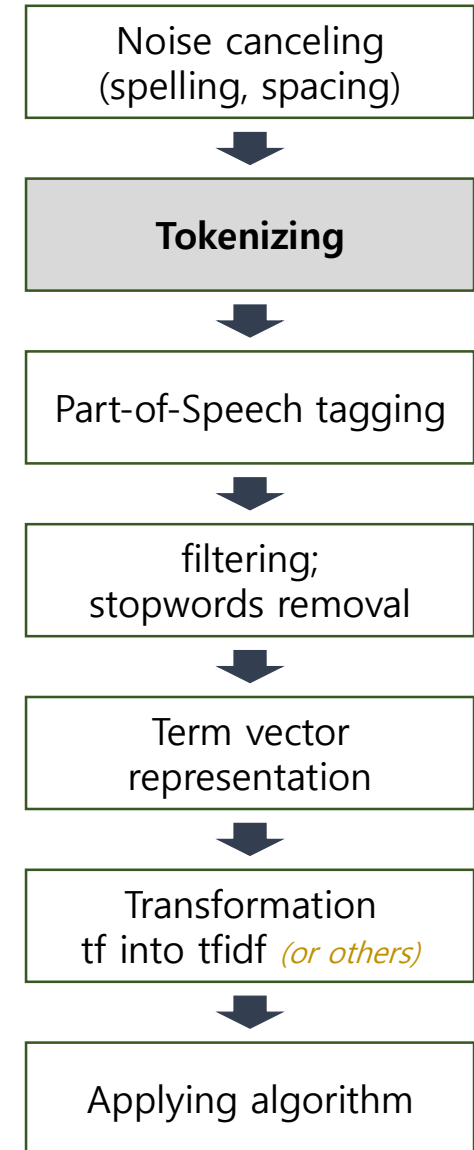
- 단어 추출은 문서 집합의 도메인이 homogeneous 할 때 잘 작동합니다.
 - 통계 기반 방법은 패턴이 잘 드러날 때 유리합니다.
 - 아프리카 내전과 관련된 문서집합이었다면 '카메룬'의 cohesion은 '카메라'보다 높았을 것입니다.
 - (tip) 가능한 분석할 문서의 종류를 나눠놓은 뒤 단어 추출을 수행합니다.

Unsupervised Tokenizers

L-Tokenizer / Max Score Tokenizer / Regex Tokenizer / Word Piece Model

Tokenizing

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



Tokenizing

- 토큰나이징은 문서집합의 성격과 목적에 따라서 적절한 전략이 다릅니다.
 - 모든 경우에 완벽한 토큰나이저는 만들기 어렵습니다.
 - 그렇기 때문에 우리는 단어 추출부터 살펴보았습니다.
- 단어 추출 방법을 이용한 토큰나이징 전략
 - 띄어쓰기가 잘 되어 있는 경우: 어절의 왼쪽 부분만을 추출하는 L-Tokenizer
 - 띄어쓰기 오류를 포함하는 경우: Max Score Tokenizer

L-Tokenizer

- 띄어쓰기가 잘 되어 있다면 어절의 구조는 L + [R] 입니다.
 - 길이가 2 이상인 L 중에서 score 가 가장 높은 단어를 선택합니다.
 - 1음절 단어는 해석이 잘 되지 않아서 이용하기 어렵습니다 (제거합니다)

```
def ltokenize(w):  
    n = len(w)  
    if n <= 2: return (w, '')  
    tokens = []  
    for e in range(2, n+1):  
        tokens.append(w[:e], w[e:], cohesion(w[:e]))  
    tokens = sorted(tokens, key=lambda x:-x[2])  
    return tokens[0][:2]
```

Max Score Tokenizer

- 띄어쓰기 오류가 많다면 어절의 구조가 $L + [R]$ 이 아닙니다.
- 사람이 띄어쓰기가 잘 되지 않은 문장을 읽은 때에는 잘 아는 단어부터 눈에 들어옵니다.
 - 어절 내에서 단어라는 확신이 높은 글자부터 분리합니다.

이런문장을직접토크나이징을해볼게요

이런문장을직접토크나이징해볼게요



이런문장을직접**[토크나이징]**해볼게요

이런문장을직접토크나이징을해볼게요

이런문장을직접[토크나이징]을해볼게요



이런[문장]을직접[토크나이징]을해볼게요

이런문장을직접토크나이징해볼게요

이런문장을직접[토크나이징]해볼게요

이런[문장]을직접[토크나이징]해볼게요



이런[문장]을**[직접]**[토크나이징]해볼게요

이런문장을직접토크나이징을해볼게요

이런문장을직접[토크나이징]을해볼게요

이런[문장]을직접[토크나이징]을해볼게요

이런[문장]을[직접][토크나이징]을해볼게요



이런[문장]을[직접][토크나이징]을[해볼]게요

이런문장을직접토크나이징을해볼게요

이런문장을직접[토크나이징]을해볼게요

이런[문장]을직접[토크나이징]을해볼게요

이런[문장]을[직접][토크나이징]을해볼게요

이런[문장]을[직접][토크나이징]을[해볼]게요



[이런, 문장, 을, 직접, 토크나이징, 을, 해볼, 게요]

이런문장을직접토크나이징을해볼게요



이런문장을직접[토크나이징]을해볼게요



이런[문장]을직접[토크나이징]을해볼게요



이런[문장]을[직접][토크나이징]을해볼게요



이런[문장]을[직접][토크나이징]을[해볼]게요



[이런, 문장, 을, 직접, 토크나이징, 을, 해볼, 게요]

Max Score Tokenizer

- 띄어쓰기가 잘 되어있지 않다면 **아는 단어부터** 자릅니다

```
cohesions = {'파스': 0.3, '파스타': 0.7, '좋아요': 0.2, '좋아': 0.5}
```

```
score = lambda x: cohesions.get(x, 0)
```

```
tokenize('파스타가좋아요')
```

[('파스', 0, 2, 0.3),
(('파스타', 0, 3, 0.7),
(('스타', 1, 3, 0),
(('스타가', 1, 4, 0),
(('타가', 2, 4, 0),
(('타가중', 2, 5, 0),
(('가중', 3, 5, 0),
(('가중아', 3, 6, 0),
(('좋아', 4, 6, 0.5),
(('좋아요', 4, 7, 0.2),
(('아요', 5, 7, 0)]

Subword 별 score 계산
(subword, begin, end, score)

[('파스타', 0, 3, 0.7),
(('좋아', 4, 6, 0.5),
(('파스', 0, 2, 0.3),
(('좋아요', 4, 7, 0.2),
(('스타', 1, 3, 0),
(('스타가', 1, 4, 0),
(('타가', 2, 4, 0),
(('타가중', 2, 5, 0),
(('가중', 3, 5, 0),
(('가중아', 3, 6, 0),
(('아요', 5, 7, 0)]

Score 기준으로 정렬

[('파스타', 0, 3, 0.7),
(('좋아', 4, 6, 0.5),
~~(('파스', 0, 2, 0.3),~~
(('좋아요', 4, 7, 0.2),
~~(('스타', 1, 3, 0),~~
~~(('스타가', 1, 4, 0),~~
~~(('타가', 2, 4, 0),~~
~~(('타가중', 2, 5, 0),~~
(('가중', 3, 5, 0),
(('가중아', 3, 6, 0),
(('아요', 5, 7, 0)]

최고점수의 단어 선택,
위치가 겹치는 단어 제거

Max Score Tokenizer

- 띄어쓰기가 잘 되어있지 않다면 **아는 단어부터** 자릅니다

```
cohesions = {'파스': 0.3, '파스타': 0.7, '좋아요': 0.2, '좋아': 0.5}
score = lambda x: cohesions.get(x, 0)
```

[파스타]가좋아요

[('파스타', 0, 3, 0.7),
('좋아', 4, 6, 0.5),
~~('파스', 0, 2, 0.3),~~
('좋아요', 4, 7, 0.2),
~~('스타', 1, 3, 0),~~
~~('스타가', 1, 4, 0),~~
~~('타가', 2, 4, 0),~~
~~('타가좋', 2, 5, 0),~~
('가좋', 3, 5, 0),
('가좋아', 3, 6, 0),
('아요', 5, 7, 0)]



[파스타]가[좋아]요

[('파스타', 0, 3, 0.7),
('좋아', 4, 6, 0.5),
~~('파스', 0, 2, 0.3),~~
('좋아요', 4, 7, 0.2),
~~('스타', 1, 3, 0),~~
~~('스타가', 1, 4, 0),~~
~~('타가', 2, 4, 0),~~
~~('타가좋', 2, 5, 0),~~
('가좋', 3, 5, 0),
('가좋아', 3, 6, 0),
('아요', 5, 7, 0)]



[파스타, 가, 좋아, 요]

Regex Tokenizer

- 언어가 바뀌는 부분도 규칙 기반으로 띄어둘 수 있습니다.
- 한국어의 완전글자/자음/모음/숫자/기호 등이 바뀌는 부분은 단어의 경계
 - 아이고ㅋㅋ진짜? → [아이고, ㅋㅋ, 진짜, ?]
 - 아이고ㅋㅋㅌㅌ진짜? = [아이고, ㅋㅋ, ㅌㅌ, 진짜, ?]

Regex Tokenizer

- 블로그/채팅에서 자주 발생하는 typo 역시 규칙으로 해결 가능합니다

- 으익ㅋㅋㅋㅋㅋㅋ큐ㅠㅠㅠㅠㅠ → [으익, ㅋㅋㅋ, ㅠㅠㅠ]

(1) 큐 정규화: 으익ㅋㅋㅋㅋㅋㅋㅋㅋㅠㅠㅠㅠㅠ

(2) 반복되는 글자 단순화: 으익ㅋㅋㅋㅠㅠㅠ

(3) RegexTokenizer: [으익, ㅋㅋㅋ, ㅠㅠㅠ]

Regex Tokenizer

- Tokenizer는 아래의 순서대로 적용할 수 있습니다
 1. Normalizer
 2. Regex tokenizer
 3. (Spacing correction): 아직 다루지 않았습니다
 4. L-Tokenizer / MaxScoreTokenizer / KoNLPy 등의 단어를 인식하는 토크나이저

Word Piece Model

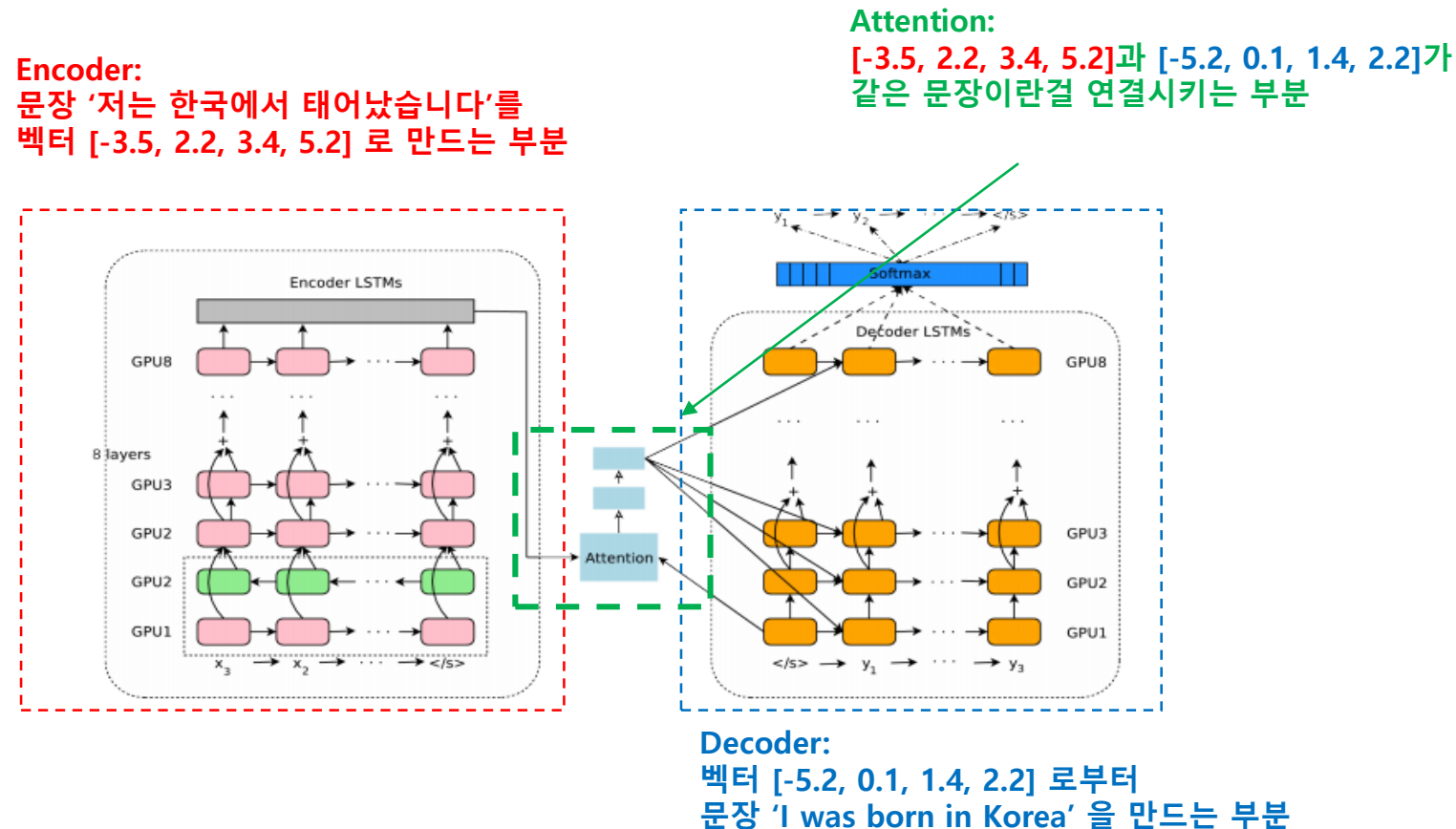
- 모든 경우에 단어를 제대로 인식해야 하는가?
 - 다루는 대상이 문서인가? 단어인가?
- 키워드 추출 / 토픽 추출은 분석의 대상이 단어이기 때문에 단어가 제대로 인식되어야 이해가 가능함

Word Piece Model

- 모든 경우에 단어를 제대로 인식해야 하는가?
 - 다루는 대상이 문서인가? 단어인가?
- 질 좋은 문서 벡터의 표현이 중요하며, 그 과정에서 단어가 반드시 제대로 나뉘어질 필요는 없습니다.

Word Piece Model

- 두 언어의 문장이 벡터로 잘 표현되면 번역은 좋은 성능을 보여줍니다.
- 알고리즘의 역할은 두 문장을 벡터로 만들고, 두 벡터 공간을 연결하는 것입니다.



Word Piece Model

- RNN 기반 모델은 계산비용 때문에 이용가능한 단어 개수가 제한됩니다.
 - 30 ~ 80k 의 단어가 현실적이며, 이로인해 발생하는 미등록단어 문제를 해결하기 위해 단어를 subword units 으로 표현하는 토큰라이저를 이용합니다.
 - WPM 은 희귀한 단어인 Jet과 feud를 [_J, et], [_fe ud] 인 subword units 으로 나누며, 자주 나오는 단어는 앞에 _를 붙여서 그대로 이용합니다

Sentence: Jet makers feud over seat width with big orders at stake

Tokens : _J et _makers _fe ud _over _seat _width _with _big _orders _at _stake

Word Piece Model

- k_1 개의 단어와 k_2 개의 subwords 를 섞어 이용할 수 있습니다.
 - k_1 개의 단어가 아닌 경우에 subwords unit 으로 표현합니다.
- WPM은 최소한의 units을 이용하여 문서의 모든 문장을 표현합니다.
 - 데이터 압축입니다.

Sentence: Jet makers feud over seat width with big orders at stake

Tokens : _J et _makers _fe ud _over _seat _width _with _big _orders _at _stake

Word Piece Model

```
import re, collections

def get_stats(vocab):
    pairs = collections.defaultdict(int)
    for word, freq in vocab.items():
        symbols = word.split()
        for i in range(len(symbols)-1):
            pairs[symbols[i],symbols[i+1]] += freq
    return pairs

def merge_vocab(pair, v_in):
    v_out = {}
    bigram = re.escape(' '.join(pair))
    p = re.compile(r'(?!\S)' + bigram + r'(?!\S)')
    for word in v_in:
        w_out = p.sub(' '.join(pair), word)
        v_out[w_out] = v_in[word]
    return v_out
```

- `get_state()` 는 bigram 빈도수를 계산합니다.
- `merge_vocab()` 은 가장 빈번한 bigram 을 하나의 unit 으로 묶습니다.

Word Piece Model

```
vocab = {'l o w </w>' : 5,  
        'l o w e r </w>' : 2,  
        'n e w e s t </w>':6,  
        'w i d e s t </w>': 3}  
  
num_merges = 10  
  
# training units  
for i in range(num_merges):  
    pairs = get_stats(vocab)  
    best = max(pairs, key=pairs.get)  
    vocab = merge_vocab(best, vocab)
```

```
# subword tokens  
vocab = {  
    'low</w>': 5,  
    'low e r </w>': 2,  
    'newest</w>': 6,  
    'wi d est</w>': 3  
}
```

```
# merging  
('e', 's')  
('es', 't')  
('est', '</w>')  
('l', 'o')  
('lo', 'w')  
('n', 'e')  
('ne', 'w')  
('new', 'est</w>')  
('low', '</w>')  
('w', 'i')
```

```
# final units  
{'low</w>': 5,  
 'low': 2,  
 'e': 2,  
 'r': 2,  
 '</w>': 2,  
 'newest</w>': 6,  
 'wi': 3,  
 'd': 3,  
 'est</w>': 3}
```

Word Piece Model

- WPM은 토큰으로부터 문장으로의 복원이 쉽습니다.

Sentence: Jet makers feud over seat width with big orders at stake

Tokens : _J et _makers _fe ud _over _seat _width _with _big _orders _at _stake

```
def recover(piece_str):  
    sent = piece_str.replace(' ', '')  
    words = sent.split('_')  
    return words
```


Word Piece Model

- WPM 은 translator 처럼 단어를 직접 이용하지 않는 모델에 적합합니다.
- Labeling 이나 토픽 모델링에는 적합하지 않습니다. 분석의 목적에 맞게 토큰나이지어를 선택해야 합니다.

Word Piece Model

- Units 의 개수는 충분히 크게 잡는 것이 좋습니다. Infrequent words 가 음절로 분해될 수 있습니다. (2016-10-20 뉴스, n units = 5000)

[doc=4]: 브뤼셀 연합뉴스 김병수 특파원 독일 **정부는** 19일 원자력발전소를 폐쇄하기로 함에 따라 원자력 **발전소** 운영자들에게 **핵폐기물** 처리를 지원하는 펀드에 235억 유로 260억 달러 29조 원 를 지불하도록 하는 계획을 승인했다고 언론들이 보도했다 앞서 독일은 5년 전 일본 후쿠시마 원전사태 이후 오는 2022년까지 원전 17기를 모두 폐쇄하기로 하고 오는 205

[BPE 5000 tokens]: 브뤼셀 _ 연합뉴스_ 김 병 수_ 특 파 원_ 독 일_ **정 부는**_ 19일_ 원 자 력 발전 소를_ 폐 쇄 하기로_ 함 에_ 따라_ 원 자 력_ **발 전 소**_ 운 영 자 들에게_ **핵 폐 기 물**_ 처 리를_ 지원 하는_ 펀 드 에_ 23 5 억_ 유로 _ 26 0억 _ 달 러 _ 29 조_ 원 _ 를 _ 지 불 하 도록_ 하는_ 계 획 을_ 승 인 했다고_ 언 론 들이_ 보 도 했 다_ 앞 서_ 독 일 은_ 5년_ 전_ 일 본_ 후 쿠 시 마_ 원 전 사 태_ 이 후_ 오 는_ 20 22 년 까 지_ 원 전_ 17 기 를_ 모 두_ 폐 쇄 하 기 로_ 하 고_ 오 는_ ...

Word Piece Model

- Units 의 개수는 충분히 크게 잡는 것이 좋습니다. Infrequent words 가 음절로 분해될 수 있습니다. (2016-10-20 뉴스, n units = 50000)

[doc=4]: 브뤼셀 연합뉴스 김병수 특파원 독일 **정부는** 19일 원자력발전소를 폐쇄하기로 함에 따라 원자력 **발전소** 운영자들에게 **핵폐기물** 처리를 지원하는 펀드에 235억 유로 260억 달러 29조 원 를 지불하도록 하는 계획을 승인했다고 언론들이 보도했다 앞서 독일은 5년 전 일본 후쿠시마 원전사태 이후 오는 2022년까지 원전 17기를 모두 폐쇄하기로 하고 오는 205

[BPE 50000 tokens]: 브뤼셀_연합뉴스_김병수_특파원_독일_정부는_19일_원자력발전소를_폐쇄하기로_함에_따라_원자력_발전소_운영자들에게_핵폐기물_처리를_지원하는_펀드에_235억_유로_260억_달러_29조_원_를_지불하도록_하는_계획을_승인했다고_언론들이_보도했다_앞서_독일은_5년_전_일본_후쿠시마_원전사태_이후_오는_2022년_까지_원전_17기를_모두_폐쇄하기로_하고_오는_2050년까지_전기생산량의_80_를_재생에너지로_...

Performance of tokenizers

Sentiment classification

뉴스 검색 엔진을 이용한 유사 문서 탐색

-
- 토큰나이저의 독립적인 성능평가는 잘 하지 않습니다.
 - 토큰나이저는 다른 작업의 전처리 과정에 이용됩니다.
 - 다른 작업의 성능 평가를 통하여 간접적으로 성능을 측정합니다.
 - 각 문제에 적절하게 토큰나이저의 성능 평가를 설계/진행해야 합니다.
 - 통계기반의 단어 추출 성능 측정은 특히 어렵습니다.
 - 정답 단어 사전이 없는 경우이기 때문에 테스트에 쓸 정답도 없습니다
 - 반드시 인식되어야 하는 단어들이 잘 인식되는지 정성 평가를 하기도 합니다.

Sentiment classification

- 영화 평 데이터를 이용하여 sentiment classification 을 수행합니다.
 - Binary classification 으로 문제를 단순화 합니다.
 - 1 ~ 3 점 : negative
 - 4 ~ 8 점 : ignore
 - 9 ~ 10 점 : positive

Sentiment classification

- 영화 평 데이터를 이용하여 sentiment classification 을 수행합니다.
 - 영화 리뷰는 띄어쓰기 오류가 존재하기 때문에 Max Score Tokenizer
 - 단어 점수는 (1) cohesion, (2) cohesion * right-side branching entropy
 - Word Piece Model 은 units 의 개수를 다양하게 설정
 - 트위터 한국어 분석기와 띄어쓰기를 baseline 으로 이용

Sentiment classification

- 긍/부정 분류의 핵심 features 는 bi-gram: '재미 + 없다'
- 트위터 한국어 분석기보다도 단어 추출 기법의 성능이 더 좋음

model	unigram		uni + bigram	
	accuracy	rank	accuracy	rank
WPM 3000	89.12%	10	92.67%	9
WPM 5000	89.56%	9	92.95%	8
WPM 10000	91.69%	8	93.47%	3
WPM 20000	92.23%	4	93.41%	4
WPM 30000	92.43%	2	93.35%	6
WPM 50000	92.65%	1	93.32%	7
cohesion	92.27%	3	93.48%	2
csbe	92.05%	5	93.63%	1
space	92.04%	6	92.13%	10
twitter	91.91%	7	93.39%	5

Sentiment classification

- WPM 은 units 이 어느 수준 이상 확보되어야 제대로 작동
- WPM 3000 의 uni + bigram 의 성능과 WPM 50000 unigram 이 비슷
- WPM 3000 uni + bigram 의 차원이 약 50k

model	unigram		uni + bigram	
	accuracy	rank	accuracy	rank
WPM 3000	89.12%	10	92.67%	9
WPM 5000	89.56%	9	92.95%	8
WPM 10000	91.69%	8	93.47%	3
WPM 20000	92.23%	4	93.41%	4
WPM 30000	92.43%	2	93.35%	6
WPM 50000	92.65%	1	93.32%	7
cohesion	92.27%	3	93.48%	2
csbe	92.05%	5	93.63%	1
space	92.04%	6	92.13%	10
twitter	91.91%	7	93.39%	5

뉴스 검색 엔진을 이용한 유사 문서 탐색

- 단어가 잘 추출되었다면 이를 이용한 유사 뉴스 검색이 가능합니다.
 - 문서는 몇 개의 키워드로 요약이 되며, 이를 통하여 검색된 동일 뉴스는 비슷한 뉴스일 가능성이 높습니다.
 - 네이버의 기구축된 사전과 서비스를 이용하여 단어 추출 방법의 성능을 평가합니다.

모든뉴스 ▾ 나이지리아 검색 통합검색

뉴스 상세검색 | 뉴스 라이브러리 상세검색 ?

범위

☒ 제목과 본문 ☐ 제목에서만

기간

☐ 전체 ☐ 최근 한달간 ☐ 최근 일주일간 ☒ 직접입력 2015-01-16 ~ 2015-01-16

최신뉴스 검색

뉴스 라이브러리 검색

RSS | 주소복사 | 도움말

뉴스

포토뉴스

TV뉴스

신문게재기사

보도자료

검색결과 (1 ~ 10 / 17건)

정확도순

최신순

제목과 내용보기

제목만 보기

마지막 검색 시각으로부터 00:10가 지났습니다.

편리한 실시간 기사검색을 원하시면

자동고침 ▾

 설정을 해주세요.

"나이지리아군, 보코하람 조직원 78명 사살"

연합뉴스

2015.01.16

네이버뉴스

... 신화=연합뉴스) 나이지리아군이 수니파 극단주의 테러단체 보코하람 조직원 78명을 사살했다고 치안 소식통이 15일(현지 시간) 밝혔다. 소식통은 외국인들을 포함한 보코하람 대원 수백 명이 전날 새벽 동북부 보르노 주 비우의 군기지를 습격했다가 격

뉴스 검색 엔진을 이용한 유사 문서 탐색

- 잘못된 단어가 입력되면, 비슷한 뉴스는 검색되지 않습니다.

모든뉴스 **나이** 통합검색

범위 ☒ 제목과 본문 ☐ 제목에서만
기간 ☐ 전체 ☐ 최근 한달간 ☐ 최근 일주일간 ☒ 직접입력 2015-01-16 ~ 2015-01-16

최신뉴스 검색 뉴스 라이브러리 검색 RSS | 주소복사 | 도움말

뉴스 | 포토뉴스 | TV뉴스 | 신문계재기사 | 보도자료

검색결과 (1 ~ 10 / 877건) 정확도순 최신순

마지막 검색 시각으로부터 00:06가 지났습니다.
편리한 실시간 기사검색을 원하시면 [자동고침] 설정해주세요.

 **"클라라 폴라리스 진실 공방, '父 이승규 색시-노출 이미지 좋아해, 지금 나이에 안 하면 못 한다'고"** 미디어파나뉴스 | 2015.01.16 | [🔗](#)
... '지금 네 나이에 안 하면 못한다. 다 해봐라'라고 종게 말씀해주신다"라고 말했다. 이어 "만약 색시 이미지로 질타만 받았다면 아버지도 걱정했을 텐데, 사랑받는 모습을 보여서 항상 긍정적이다. 부모님이 늘 좋은 면을 보고, 장점을 보시려고 한다"라고 덧붙였다. 한편 클라라는 소속사...

 **"'어린이집 아동학대' 이병진 '알과 같은 나이, 내가 맞은 듯 아파' 분노"** 티브이데일리 | 2015.01.16 | [🔗](#)
... 그 아이 우리 딸과 나이도 같고 내가 맞은 듯 아프고 무서웠어. 아직도 인터넷에 관련 기사와 캡처 사진을 보면 분노가 치밀어"는 댓글을 남겨 동조했다. 여기에 엄정화는 "죄가 너무 가볍다는게 슬퍼, 모든게"라는 답글을 다시 남겨 안타까움을 전했다. 이는 지난 8일 인천시 연수구의 한...

 **"김수로, '진짜 사나이' 군 전역 앞둔...나이 잊은 중년 병사"** 배국남닷컴 | 2015.01.16 | [🔗](#)
[배국남닷컴 이꽃들 기자] 김수로가 '진짜 사나이'를 통해 군 전역한다. 김수로가 18일 방송되는 MBC '일밤-진짜 사나이'를 통해 지난해 3월 입대한 이후 어느덧 전역을 앞두고 있다. 남자들의 진한 전우애와 유쾌한 내무생활로 군에 대한 색다른 재미와 인식을 안겨준 '진짜...

모든뉴스 **지리** 통합검색

범위 ☒ 제목과 본문 ☐ 제목에서만
기간 ☐ 전체 ☐ 최근 한달간 ☐ 최근 일주일간 ☒ 직접입력 2015-01-16 ~ 2015-01-16

최신뉴스 검색 뉴스 라이브러리 검색 RSS | 주소복사 | 도움말

뉴스 | 포토뉴스 | TV뉴스 | 신문계재기사 | 보도자료

검색결과 (1 ~ 10 / 59건) 정확도순 최신순

마지막 검색 시각으로부터 00:04가 지났습니다.
편리한 실시간 기사검색을 원하시면 [자동고침] 설정해주세요.

 **"한국풍수지리연구원 전항수 원장, 다양한 분야에 활용가능한 풍수지리 연구"** 스포츠조선 | 2015.01.16 | 네이버뉴스 | [🔗](#)
전항수 한국풍수지리원장, 한국 풍수지리 연구원(대표 전항수, www.poongsoo.net)에서는 많은 분야에 활용되고 있는 공공용지, 신도시, 부동산 개발 및 아파트 부지, 공장용지, 상업용지, 개인주택지, 별장지 그리고 공간배치 등 다양한 분야에서 활용되는 풍수지리학을 연구하고 있다....

 **"[생활속의 풍수지리] 반풍수 집안 망하게 한다"** 경남신문 | 2015.01.16 | [🔗](#)
... 되는 것이요, 한 치만 낮아도 물이 되는 것이다'라는 글귀가 있다. 스님은 이러한 이치를 알고서 조연을 했으리라 본다. 하지만 높은 곳이 지기가 좋은 곳이 되기도 하지만 물이 많은 곳이 될 수도 있음을 알아야 한다. 주재민 화산풍수지리연구소장 (화산풍수·수맥연구원 055-297-3882) ;

 **"[한국사와 사람전문 '한림학원'] 성공 대입전략 한국사와 사람에 있다"** 내일신문 | 2015.01.16 | [🔗](#)
... 영역, 지리 영역, 윤리 영역 별로 전문 강사를 두고 초·중·고 학생들에게 학년별, 과목별, 분야별로 교육을 진행하고 있다. 권 원장은 "분야별 한국사와 사회탐구 수업만을 전문으로 진행하는 학원을 찾는 것은 서울의 주요 학원가에서도 어려운 일이다. 특히 역사, 일반사회, 지리...

뉴스 검색 엔진을 이용한 유사 문서 탐색

- 잘못된 단어가 입력되면, 비슷한 뉴스는 검색되지 않습니다

모든뉴스

나이 지리

검색

통합검색

뉴스 상세검색 | 뉴스 라이브러리 상세검색 ?

범위

☒ 제목과 본문 ☐ 제목에서만

기간

☐ 전체 ☐ 최근 한달간 ☐ 최근 일주일간 ☒ 직접입력

2015-01-16

~

2015-01-16

최신뉴스 검색

뉴스 라이브러리 검색

RSS | 주소복사 | 도움말

뉴스

포토뉴스

TV뉴스

신문게재기사

보도자료

검색결과 (1 ~ 1 / 1건)

정확도순

최신순

제목과 내용보기

제목만 보기

마지막 검색 시각으로부터 00:07가 지났습니다.
편리한 실시간 기사 검색을 원하시면

자동고침

 설정을 해주세요.



"시력의 질적 향상을 극대화한 초정밀 라식이 있다?" 한국경제 | 2015.01.16 | 네이버뉴스 | [🔗](#)
... 최근엔 스마트폰 및 컴퓨터의 사용량이 늘면서 젊은 **나이**에도 시력저하 현상을 겪는 사람이 많아지고 있
다. 이에 시력이 좋지 않은 사람들... 맞은편, 1층 스타벅스)에 위치해 **지리**적인 접근성 역시 큰 장점으로 꼽
힌다. (사진출처: 영화 '투어리스트' 스틸컷) news@wstarnews.com

1

뉴스 검색 엔진을 이용한 유사 문서 탐색

- 실험에 이용한 2015-01-16 큰 이슈는 없었기에 한 사건에 대해 50 여개 뉴스를 가정.
- 3 개의 단어를 입력하여 50 개에 가까운 뉴스를 검색하는 단어셋을 이용합니다

Select this comb.

[나이지리아=10.0,
보코하람=8.0,
차드=8.0,
카메=6.0,
있는=6.0,
시위=5.0,
대통령=5.0,
있다=4.0,
우리=4.0,
말했다=4.0
...]

모든뉴스 ▼ 나이지리아 보코하람 대통령

뉴스 상세검색 | 뉴스 라이브러리 상세검색 ?

검색결과 (1 ~ 5 / 5건) 정확도순 | 최신순

마지막 검색 시각으로부터 00:14가 지났습니다.
편리한 실시간 기사 검색을 원하시면 [자동고침 ▼] 설정을 해주세요.

"나이지리아 대통령, 보코하람 피해 격심한 보르노 주 방문" 뉴스.
[마이두구리(나이지리아) = AP/뉴스스] 양문평 기자 = 국력 조나단 L
코하람의 공세로 처참한 피해를 입은 보르노 주를 방문했다. 대통령실은
보코하람과 싸우고 있는...

"나이지리아군, 보코하람 조직원 78명 사살" 연합뉴스 | 2015. 01.
... 신년 들어 보코하람은 지난 3일 보르노 주의 바가를 공격해 최대 2천
조녀선 나이지리아 대통령은 이날 군 수뇌부와 함께 보르노주 주도인 M
다. 조녀선 대통령은 다음달...

"나이지리아군, 군 기지 습격한 보코하람 조직원"
헤럴드경제 | 2015. 01. 16 | 네이버뉴스 | [🔗](#)
... 보코하람은 새해들어 보르노 주의 바가를 공격해
했다. 한편 국력 조녀선 나이지리아 대통령은 이날 군
문해 카심 셰티마 주지사를 면담했다. 조녀선 대통령.

[나이지리아=10.0,
보코하람=8.0,
차드=8.0,
카메=6.0,
있는=6.0,
시위=5.0,
대통령=5.0,
있다=4.0,
우리=4.0,
말했다=4.0
...]

모든뉴스 ▼ 보코하람 카메 대통령

뉴스 상세검색 | 뉴스 라이브러리 상세검색 ?

!

"보코하람 카메 대통령"에 대한 검색결과가 없습니다.

· 단어의 철자가 정확한지 확인해 주세요.
· 검색어의 단어수를 줄이거나 다른 검색어로 검색해 보세요.
· 보다 일반적인 검색어로 다시 검색해 보세요.
· 정확한 검색어가 생각나지 않는다면, 각 색션에서 기사를 확인해 보세요.

이런 기능도 활용해 보세요.
[통합검색 이동](#) | [상세검색에서 찾기](#)

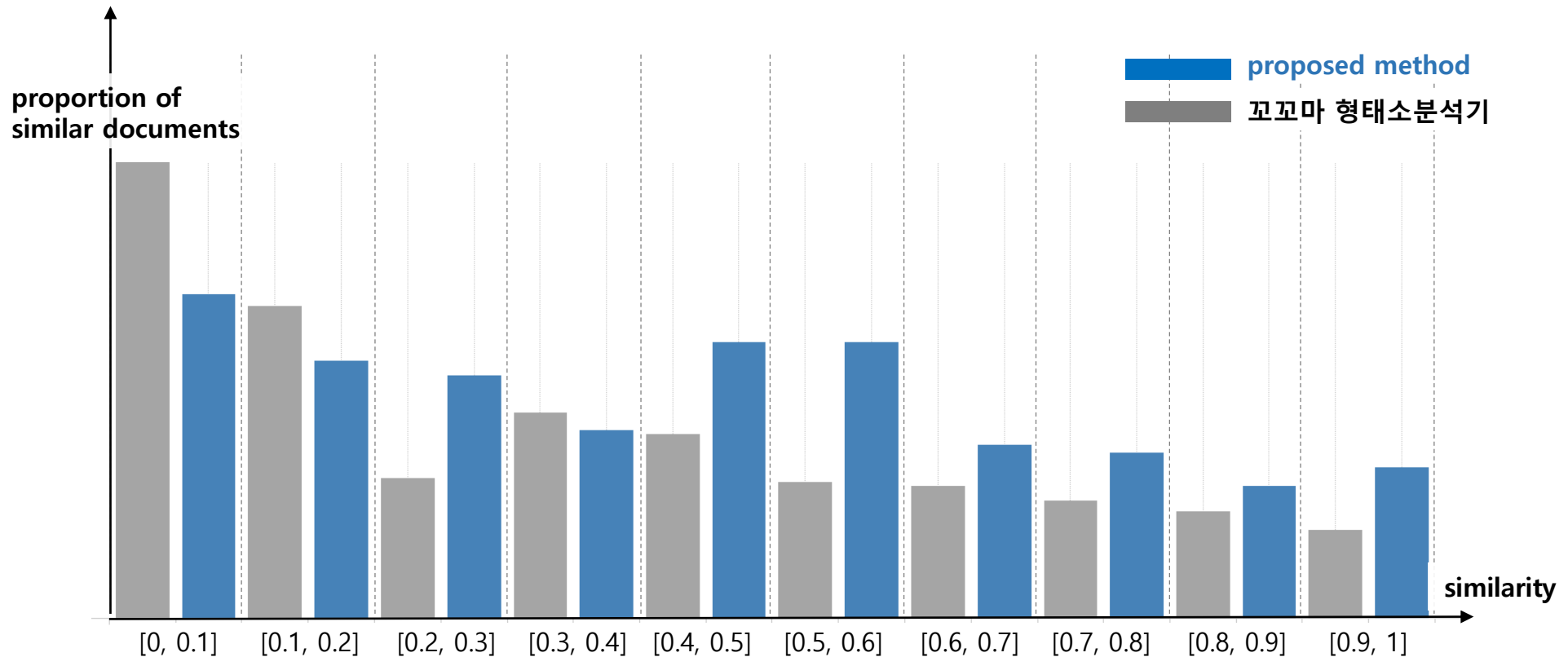
뉴스 검색 엔진을 이용한 유사 문서 탐색

- 검색된 문서의 평균 유사도를 계산하여 뉴스의 유사도를 확인합니다



뉴스 검색 엔진을 이용한 유사 문서 탐색

- Cohesion + L-Tokenizer 를 이용하여 검색한 뉴스의 유사도들이 꼬꼬마 형태소 분석기를 이용한 경우보다 높습니다. 제안된 토크나이저의 성능을 간접적으로 확인합니다.



Packages

- <https://github.com/lovit/soynlp> 에 단어추출/명사추출/토큰나이저를 구현

```
from soynlp import DoublespaceLineCorpus
from soynlp.word import WordExtractor

corpus = DoublespaceLineCorpus(fname, iter_sent=True)
word_extractor = WordExtractor(corpus, min_count=10)
words = word_extractor.extract()
words['드라마']
```

```
Scores(cohesion_forward=0.6093651029086764,
        cohesion_backward=0.5282705437953743,
        left_branching_entropy=3.6583115265560924,
        right_branching_entropy=3.675624807575614,
        left_accessor_variety=128,
        right_accessor_variety=136,
        leftside_frequency=2375,
        rightside_frequency=1284)
```


Packages

- <https://github.com/lovit/soynlp> 에 단어추출/명사추출/토큰라이저를 구현

```
from soynlp.tokenizer import LTokenizer

scores = {w:s.cohesion_forward for w, s in words.items()}
tokenizer = LTokenizer(scores=scores)
tokenizer.tokenize('뉴스의 기사를 이용했던 예시입니다')
```

```
['뉴스', '의 ', '기사', '를 ', '이용', '했던', '예시', '입니다']
```

Packages

- <https://github.com/lovit/soynlp> 에 단어추출/명사추출/토큰나이저를 구현

```
from soynlp.tokenizer import MaxScoreTokenizer
```

```
scores = {w:s.cohesion_forward for w, s in words.items()}
```

```
tokenizer = MaxScoreTokenizer(scores=scores)
```

```
tokenizer.tokenize('맛있는짜파게티파스타일식초밥소바김볶다먹고싶어라일단김밥천국으로고고')
```

```
['맛있', '는', '짜파게티', '파스타', '일식', '초밥', '소바', '김볶', '다', '먹고', '싶어', '라', '일단',  
'김밥천국', '으로', '고고']
```

Packages

- <https://github.com/lovit/soynlp> 에 단어추출/명사추출/토큰나이저를 구현

```
from soynlp.noun import LRNounExtractor
```

```
noun_extractor = LRNounExtractor(min_count=50)
```

```
nouns = noun_extractor.train_extract(corpus, minimum_noun_score=0.5)
```

```
nouns['설입']
```

```
NounScore(frequency=67, score=0.926, known_r_ratio=0.529)
```

```
nouns['드라마']
```

```
NounScore(frequency=4976, score=0.522, known_r_ratio=0.601)
```