Soynlp 소개

soynlp는 한국어 처리를 위한 파이썬 패키지 중 하나다. koNLPy에서 제공하는 형태소분석기는 형태소 기반으로 문서를 토큰화할 수 있는 기능을 제공하지만 새롭게 만들어진 미등록 단어들은 인식이 잘 되지 않는 단점을 가지고 있다. 이를 해결하기 위해서는 사용자 사전에 단어를 등록하는 절치를 거쳐야 한다. soynlp는 이러한 과정을 돕기 위해 사용자 사전과 형태소분석 없이 cohesion 기반으로 토큰화를 할 수 있는 기능을 제공한다.

- https://github.com/lovit/soynlp (https://github.com/lovit/soynlp)

패키지 설치

패키지는 다음과 같이 설치할 수 있다.

```
pip install soynlp
```

말뭉치 다운로드

soynlp는 koNLPy와 달리 패키지 내에서 말뭉치를 제공하지 않는다. 대신 github 레포에 예제 말뭉치 파일이 있으므로 이를 다운로드 받아서 사용하자.|

In [1]:

```
!wget https://raw.githubusercontent.com/lovit/soynlp/master/tutorials/2016-10-20.txt -O 2016-10-20
```

```
--2019-09-25 16:09:31--  https://raw.githubusercontent.com/lovit/soynlp/master/tutor
ials/2016-10-20.txt (https://raw.githubusercontent.com/lovit/soynlp/master/tutorial
s/2016-10-20.txt)
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 151.101.228.133
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|151.101.228.133
|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 43694449 (42M) [text/plain]
Saving to: ‘2016-10-20.txt’

2016-10-20.txt      100%[===================>]  41.67M  1.08MB/s    in 15s

2019-09-25 16:09:49 (2.78 MB/s) - ‘2016-10-20.txt’ saved [43694449/43694449]
```

이 파일은 하나의 문서가 한 줄로 되어 있고 각 줄 내에서 문장은 두 개의 공백으로 분리되어 있는 형식의 말뭉치다. DoublespaceLineCorpus 클래스로 이 말뭉치를 사용할 수 있다.

In [2]:

```
from soynlp import DoublespaceLineCorpus

# 문서 단위 말뭉치 생성
corpus = DoublespaceLineCorpus("2016-10-20.txt")
len(corpus)  # 문서의 갯수
```

Out[2]:

30091

In [3]:

```
# 앞 5개의 문서 인쇄
i = 0
for d in corpus:
    print(i, d)
    i += 1
    if i > 4:
        break
```

0
1 19 1990 52 1 22
2 오패산터널 총격전 용의자 검거 서울 연합뉴스 경찰 관계자들이 19일 오후 서울 강북구 오패산 터널 인근에서 사제 총기를 발사해 경찰을 살해한 용의자 성모씨를 검거하고 있다 성씨는 검거 당시 서바이벌 게임에서 쓰는 방탄조끼에 헬멧까지 착용한 상태였다 독자제공 영상 캡처 연합뉴스 서울 연합뉴스 김은경 기자 사제 총기로 경찰을 살해한 범인 성모 46씨는 주도면밀했다 경찰에 따르면 성씨는 19일 오후 강북경찰서 인근 부동산 업소 밖에서 부동산업자 이모 67 씨가 나오기를 기다렸다 이씨와는 평소에도 말다툼을 자주 한 것으로 알려졌다 이씨가 나와 걷기 시작하자 성씨는 따라가면서 미리 준비해온 사제 총기를 이씨에게 발사했다 총알이 빗나가면서 이씨는 도망갔다 그 빗나간 총알은 지나가던 행인 71 씨의 배를 스쳤다 성씨는 강북서 인근 치킨집까지 이씨 뒤를 쫓으며 실랑이하다 쓰러뜨린 후 총기와 함께 가져온 망치로 이씨 머리를 때렸다 이 과정에서 오후 6시 20분께 강북구 번동 길 위에서 사람들이 싸우고 있다 총소리가 났다 는 등의 신고가 여러건 들어왔다 5분 후에 성씨의 전자발찌가 훼손됐다는 신고가 보호관찰소 시스템을 통해 들어왔다 성범죄자로 전자발찌를 차고 있던 성씨는 부엌칼로 직접 자신의 발찌를 끊었다 용의자 소지 사제총기 2정 서울 연합뉴스 임헌정 기자 서울 시내에서 폭행 용의자가 현장 조사를 벌이던 경찰관에게 사제총기를 발사해 경찰관이 숨졌다 19일 오후 6시28분 강북구 번동에서 둔기로 맞았다 는 폭행 피해 신고가 접수돼 현장에서 조사하던 강북경찰서 번동파출소 소속 김모 54 경위가 폭행 용의자 성모 45 씨가 쏜 사제총기에 맞고 쓰러진 뒤 병원에 옮겨졌으나 숨졌다 사진은 용의자가 소지한 사제총기 신고를 받고 번동파출소에서 김창호 54 경위 등 경찰들이 오후 6시 29분께 현장으로 출동했다 성씨는 그사이 부동산 앞에 놓아뒀던 가방을 챙겨 오패산 쪽으로 도망간 후였다 김 경위는 오패산 터널 입구 오른쪽의 급경사에서 성씨에게 접근하다가 오후 6시 33분께 풀숲에 숨은 성씨가 허공에 난사한 10여발의 총알 중 일부를 왼쪽 어깨 뒷부분에 맞고 쓰러졌다 김 경위는 구급차가 도착했을 때 이미 의식이 없었고 심폐소생술을 하며 병원으로 옮겨졌으나 총알이 폐를 훼손해 오후 7시 40분께 사망했다 김 경위는 외근용 조끼를 입고 있었으나 총알을 막기에는 역부족이었다 머리에 부상을 입은 이씨도 함께 병원으로 이송됐으나 생명에는 지장이 없는 것으로 알려졌다 성씨는 오패산 터널 밑쪽 숲에서 오후 6시 45분께 잡혔다 총격현장 수색하는 경찰들 서울 연합뉴스 이효석 기자 19일 오후 서울 강북구 오패산 터널 인근에서 경찰들이 폭행 용의자가 사제총기를 발사해 경찰관이 사망한 사건을 조사 하고 있다 총 때문에 쫓던 경관들과 민간인들이 몸을 숨겼는데 인근 신발가게 직원 이모씨가 다가가 성씨를 덮쳤고 이어 현장에 있던 다른 상인들과 경찰이 가세해 체포했다 성씨는 경찰에 붙잡힌 직후 나 자살하려고 한 거다 맞아 죽어도 괜찮다 고 말한 것으로 전해졌다 성씨 자신도 경찰이 발사한 공포탄 1발 실탄 3발 중 실탄 1발을 배에 맞았으나 방탄조끼를 입은 상태여서 부상하지는 않았다 경찰은 인근을 수색해 성씨가 만든 사제총 16정과 칼 7개를 압수했다 실제 폭발할지는 알 수 없는 요구르트병에 무언가를 채워두고 심지를 꽂은 사제 폭탄도 발견됐다 일부는 숲에서 발견됐고 일부는 성씨가 소지한 가방 안에 있었다
3 테헤란 연합뉴스 강훈상 특파원 이용 승객수 기준 세계 최대 공항인 아랍에미리트 두바이국제공항은 19일 현지시간 이 공항을 이륙하는 모든 항공기의 탑승객은 삼성전자의 갤럭시노트7을 휴대하면 안 된다고 밝혔다 두바이국제공항은 여러 항공 관련 기구의 권고에 따라 안전성에 우려가 있는 스마트폰 갤럭시노트7을 휴대하고 비행기를 타면 안 된다 며 탑승 전 검색 중 발견되면 압수할 계획 이라고 발표했다 공항 측은 갤럭시노트7의 배터리가 폭발 우려가 제기된 만큼 이 제품을 갖고 공항 안으로 들어오지 말라고 이용객에 당부했다 이런 조치는 두바이국제공항 뿐 아니라 신공항인 두바이월드센터에도 적용된다 배터리 폭발문제로 회수된 갤럭시노트7 연합뉴스자료사진
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 2050년까지 전기생산량의 80 를 재생에너지로 충당하는 것을 목표로 세웠다 이날 내각을 통과한 법안은 원전 운영자들이 원전 해체와 핵폐기물 처리를 위한 포장을 책임지고 정부는 핵

폐기물 보관을 책임지도록 했다 독일 경제부는 전력회사들과 공식적인 접촉은 아직 합의
되지 않았다고 밝혔다 독일 원자력 발전소 연합뉴스 자료사진

In [4]:

```
# 문장 단위 말뭉치 생성
corpus = DoublespaceLineCorpus("2016-10-20.txt", iter_sent=True)
len(corpus)  # 문장의 갯수
```

Out[4]:

223357

In [5]:

```
# 앞 5개의 문장 인쇄
i = 0
for d in corpus:
    print(i, d)
    i += 1
    if i > 4:
        break
```

```
0 19
1 1990
2 52 1 22
3 오패산터널 총격전 용의자 검거 서울 연합뉴스 경찰 관계자들이 19일 오후 서울 강북구
오패산 터널 인근에서 사제 총기를 발사해 경찰을 살해한 용의자 성모씨를 검거하고 있다
성씨는 검거 당시 서바이벌 게임에서 쓰는 방탄조끼에 헬멧까지 착용한 상태였다 독자제공
영상 캡처 연합뉴스
4 서울 연합뉴스 김은경 기자 사제 총기로 경찰을 살해한 범인 성모 46 씨는 주도면밀했다
```

단어 추출

WordExtractor 클래스를 사용하면 형태소에 해당하는 단어를 분리하는 학습을 수행한다. 실제로는 각 단어 후보에 대해 cohesion 등의

In [6]:

```
%%time
from soynlp.word import WordExtractor

word_extractor = WordExtractor()
word_extractor.train(corpus)
```

```
training was done. used memory 0.806 Gbse memory 0.846 Gb
CPU times: user 44.3 s, sys: 520 ms, total: 44.8 s
Wall time: 44.8 s
```

extract() 메서드로 각 cohesion, branching entropy, accessor variety 등의 통계 수치를 계산할 수 있다.

In [7]:

```
word_score = word_extractor.extract()
```

```
all cohesion probabilities was computed. # words = 223348
all branching entropies was computed # words = 360721
all accessor variety was computed # words = 360721
```

Cohesion

Cohesion은 문자열을 글자단위로 분리하여 부분문자열(substring)을 만들 때 왼쪽부터 문맥을 증가시키면서 각 문맥이 주어졌을 때 그 다음 글자가 나올 확률을 계산하여 누적곱을 한 값이다.

$$\text{cohesion}(n) = \left(\prod_{i=1}^{n-1} P(c_{1:i+1} | c_{1:i}) \right)^{\frac{1}{n-1}}$$

예를 들어 "연합뉴스가"라는 문자열이 있는 경우, 각 부분문자열의 cohesion은 다음과 같다. 한 글자는 cohesion을 계산하지 않는다.

- $cohesion(2) = P(\text{연합}|\text{연})$
- $cohesion(3) = \sqrt[2]{\left(P(\text{연합}|\text{연}) \cdot P(\text{연합뉴}|\text{연합})\right)}$
- $cohesion(4) = \sqrt[3]{\left(P(\text{연합}|\text{연}) \cdot P(\text{연합뉴}|\text{연합}) \cdot P(\text{연합뉴스}|\text{연합뉴})\right)}$
- $cohesion(5) = \sqrt[4]{\left(P(\text{연합}|\text{연}) \cdot P(\text{연합뉴}|\text{연합}) \cdot P(\text{연합뉴스}|\text{연합뉴}) \cdot P(\text{연합뉴스가}|\text{연합뉴스})\right)}$

하나의 단어를 중간에서 나눈 경우, 다음 글자를 예측하기 쉬우므로 조건부확률의 값은 크다. 하지만 단어가 종료된 다음에 여러가지 조사나 결합어가 오는 경우에는 다양한 경우가 가능하므로 조건부확률의 값이 작아진다. 따라서 cohesion값이 가장 큰 위치가 하나의 단어를 이루고 있을 가능성이 높다.

In [8]:

```
word_score["연합"].cohesion_forward
```

Out[8]:

0.19433632536341 25

In [9]:

```
word_score["연합뉴"].cohesion_forward
```

Out[9]:

0.43154839105434084

In [10]:

```
word_score["연합뉴스"].cohesion_forward
```

Out[10]:

0.5710254410737682

In [11]:

```
word_score["연합뉴스는"].cohesion_forward
```

Out[11]:

0.1535595043355021

Branching Entropy

Branching Entropy는 조건부 확률의 값이 아니라 확률분포의 엔트로피값을 사용한다. 만약 하나의 단어를 중간에서 끊으면 다음에 나올 글자는 쉽게 예측이 가능하다 즉, 여러가지 글자 중 특정한 하나의 글자가 확률이 높다. 따라서 엔트로피값이 0에 가까운 값으로 작아진다. 하지만 하나의 단어가 완결되는 위치에는 다양한 조사나 결합어가 올 수 있으므로 여러가지 글자의 확률이 비슷하게 나오고 따라서 엔트로피값이 높아진다.

In [12]:

```
word_score["연합"].right_branching_entropy
```

Out[12]:

0.42721236711742844

In [13]:

```
# '연합뉴' 다음에는 항상 '스'만 나온다.
word_score["연합뉴"].right_branching_entropy
```

Out[13]:

-0.0

In [14]:

```
word_score["연합뉴스"].right_branching_entropy
```

Out[14]:

3.8967810761022053

In [15]:

```
word_score["연합뉴스는"].right_branching_entropy
```

Out[15]:

0.4101163182884O9

Accessor Variety

Accessor Variety는 확률분포를 구하지 않고 단순히 특정 문자열 다음에 나올 수 있는 글자의 종류만 계산한다. 글자의 종류가 많다면 엔트로피가 높아지리 것이라고 추정하는 것이다.

In [16]:

```
word_score["연합"].right_accessor_variety
```

Out[16]:

42

In [17]:

```
# '연합뉴' 다음에는 항상 '스'만 나온다.
word_score["연합뉴"].right_accessor_variety
```

Out[17]:

1

In [18]:

```
word_score["연합뉴스"].right_accessor_variety
```

Out[18]:

158

In [19]:

```
word_score["연합뉴스는"].right_accessor_variety
```

Out[19]:

2

soynlp는 이렇게 계산된 통계수치를 사용하여 문자열을 토큰화하는 방법도 제공한다. soynlp가 제공하는 토큰화 방법은 두 가지다.

- 띄어쓰기가 잘 되어 있는 경우: L-토큰화
- 띄어쓰기가 안되어 있는 경우: Max Score 토큰화

L-토큰화

한국어의 경우 공백(띄어쓰기)으로 분리된 하나의 문자열은 'L 토큰 + R 토큰; 구조인 경우가 많다. 왼쪽에 오는 L 토큰은 체언(명사, 대명사)이나 동사, 형용사 등이고 오른쪽에 오는 R 토큰은 조사, 동사, 형용사 등이다. 여러가지 길이의 L 토큰의 점수를 비교하여 가장 점수가 높 L단어를 찾는 것이 L-토큰화(L-tokenizing)이다. soynlp에서는 `LTokenizer` 클래스로 제공한다.

In [20]:

```
from soynlp.tokenizer import LTokenizer

scores = {word:score.cohesion_forward for word, score in word_score.items()}
l_tokenizer = LTokenizer(scores=scores)

l_tokenizer.tokenize("안전성에 문제있는 스마트폰을 휴대하고 탑승할 경우에 압수한다", flatten=False)
```

Out[20]:

```
[('안전', '성에'),
 ('문제', '있는'),
 ('스마트폰', '을'),
 ('휴대', '하고'),
 ('탑승', '할'),
 ('경우', '에'),
 ('압수', '한다')]
```

최대 점수 토큰화

최대 점수 토큰화(max score tokenizing)는 띄어쓰기가 되어 있지 않는 긴 문자열에서 가능한 모든 종류의 부분문자열을 만들어서 가장 점수가 높은 것을 하나의 토큰으로 정한다. 이 토큰을 제외하면 이 위치를 기준으로 전체 문자열이 다시 더 작은 문자열들로 나누어지는데 이 문자열들에 대해 다시 한번 가장 점수가 높은 부분문자열을 찾는 것을 반복한다.

In [21]:

```
from soynlp.tokenizer import MaxScoreTokenizer

maxscore_tokenizer = MaxScoreTokenizer(scores=scores)
maxscore_tokenizer.tokenize("안전성에문제있는스마트폰을휴대하고탑승할경우에압수한다")
```

Out[21]:

```
['안전',
 '성에',
 '문제',
 '있는',
 '스마트폰',
 '을',
 '휴대',
 '하고',
 '탑승',
 '할',
 '경우',
 '에',
 '압수',
 '한다']
```