Dictionary based Part of Speech Tagger

Hyunjoong Kim

soy.lovit@gmail.com

github.com/lovit

텍스트를 단어로 표현하는 방법 (tokenization)

- Subwords tokenization은 단어를 유한한 subwords units으로 표현
  - 번역, 임베딩에 기반한 document representation에 이용됩니다

‘복면가왕’ 꽃새우는 아이오아이 출신 가수 청하였다. 3일 오후 4시 50분 방송된 MBC ‘미스터리 음악쇼-복면가왕’에서는 복어아가씨와 꽃새우의 1라운드 무대가 펼쳐졌다. 55대 44로 복어아가씨가 승리를 거뒀고, 꽃새우는 이효리의 ‘텐미닛’을 부르며 청하임을 밝혔다. 이날 청하는 아이오아이 당시 춤으로 인기가 있지 않았냐는 질문에 “당시엔 노래보다 춤에 더 자신이 있었다. 연정과 세정이 메인보컬이라 보여줄 기회가 없었고, 이번에 보여줘서 좋다”라고 말했다.

‘ 복면 가왕 ’_ 꽃 새우 는_ 아이 오 아이_ 출 신_ 가수 청 하 였다._ 3 일_ 오후 _ 4 시_ 50 분_ 방송 된_ …

< Word Piece Model 예시 >

텍스트를 단어로 표현하는 방법 (tokenization)

- 키워드 추출 / 토픽 모델링을 위해서는 **단어가 제대로 인식**되야 합니다

‘복면가왕’ 꽃새우는 아이오아이 출신 가수 청하였다. 3일 오후 4시 50분 방송된 MBC ‘미스터리 음악쇼-복면가왕’에서는 복어아가씨와 꽃새우의 1라운드 무대가 펼쳐졌다. 55대 44로 복어아가씨가 승리를 거뒀고, 꽃새우는 이효리의 ‘텐미닛’을 부르며 청하임을 밝혔다. 이날 청하는 아이오아이 당시 춤으로 인기가 있지 않았냐는 질문에 “당시엔 노래보다 춤에 더 자신이 있었다. 연정과 세정이 메인보컬이라 보여줄 기회가 없었고, 이번에 보여줘서 좋다”라고 말했다.

(‘, 기호), (복면가왕, 명사), (’, 기호), (꽃새우, 명사), (는, 조사), (아이오아이, 명사), (출신, 명사), (가수, 명사), (청하, 명사), (였다, 동사), (., 기호), …

< 품사 판별에 의한 토크나이징 예시 >

품사 판별과 형태소 분석

- 한국어 단어의 품사는 5언 9품사로 구성되어 있습니다.

SENT: 재공연을 했어요

POS: (**재공연**, 명사), (**을**, 조사), (**했어요**, 동사)

<table>
<tr><th colspan="5">한국어 품사</th></tr>
<tr><td rowspan="4">불변어</td><td>체언</td><td>명사</td><td>대명사</td><td>수사</td></tr>
<tr><td>수식언</td><td colspan="2">관형사</td><td>부사</td></tr>
<tr><td>관계언</td><td colspan="3">조사</td></tr>
<tr><td>독립언</td><td colspan="3">감탄사</td></tr>
<tr><td>가변어</td><td>용언</td><td colspan="2">동사</td><td>형용사</td></tr>
</table>

품사 판별과 형태소 분석

- **품사 판별**은 텍스트 데이터 분석을 위한 전처리 과정 중 하나입니다

```
from konlpy.tag import Kkma

kkma = Kkma()

kkma.pos('오류보고는 실행환경, 에러메세지와함께 설명을 최대한상세히!^^')
```

```
[(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)]
```

품사 판별과 형태소 분석

- **품사 판별을 위하여** 형태소 분석이 이용될 수 있습니다

SENT: 재공연을 했어요

POS: (**재공연**, 명사), (**을**, 조사), (**했어요**, 동사)

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

- **형태소 분석**은 **단어의 구성 요소**들을 분해하여 인식하는 과정입니다

품사 판별과 형태소 분석

- 품사 사전이 잘 구축된다면, **사전기반으로도 품사판별**을 할 수 있습니다

SENT: 재공연을 했어요

POS: (**재공연**, 명사), (**을**, 조사), (**했어요**, 동사)

명사사전: {, … 재공연, … }

동사사전: {, … 했어요, 했엉, 해써용, … }

- 품사 판별이 목적이라면 형태소분석 과정이 필수는 아닙니다

미등록단어 문제

- 사전 기반으로 작동하는 형태소/품사 분석은 **사전 구성이 핵심**입니다

이 예제에서는 사전을 수정해 보겠습니다. 사전 파일들은

src/main/resources/com/twitter/penguin/korean/util/ 에 있습니다.

`src/main/resources/com/twitter/penguin/korean/util/noun/wikipedia_title_nouns.txt` 에 동사가 들어가 있네요. 삭제했습니다. (이런 경우가 많이 있습니다. 수작업으로 없애 주어야 하는데요 여러분의 도움을 구합니다. 아울러 복합명사도 최대한 분리되어야 합니다. 하동청룡리석불좌상 -> 하동 청룡리 석불 좌상)

< 트위터 한국어 분석기의 contribution guide snapshot >

https://github.com/twitter/twitter-korean-text/blob/master/docs/contribution-guide.md

미등록단어 문제

- 좋은 품질을 위하여 **사용자 사전**을 추가하여 사용합니다

  - 품사판별이 제대로 이뤄지지 않은 단어를 사전에 추가
  - 이 **노동집약적인 과정을 최대한 자동화**하는 것이 **soynlp** 의 목표입니다

soynlp

- soynlp 는 다음 그림처럼, 통계 기반으로 단어와 품사를 추정하는 기능이 포함되어 있습니다.

soynlp 는 통계 기반 단어/품사 추정 기능이 포함되어 있습니다

문장: 아이오아이는이번공연에서좋은것모습을보였습니다이뻐이뻐

단어 추출을 통한
토크나이징

단어열: [아이오아이, 는, 이번, 공연, 에서,
좋은, 것, 모습, 을, 보였습니다, 이뻐, 이뻐]

품사 추정을 통한
품사 사전 업데이트

명사 사전 += [아이오아이, ...]
동사 사전 += [잘했어용, ...]

품사 사전을 이용한
품사 판별

품사열: [(아이오아이, 명사), (는, 조사), (이번, 명사), (공연, 명사),
(에서, 조사), (좋은, 형용사), (것, 명사), (모습, 명사), (을, 조사),
(보였습니다, 동사), (이뻐, 형용사), (이뻐, 형용사)]

후처리

사전 기반으로 작동하는 품사 판별기를 만들 수 있습니다

문장: 아이오아이는이번공연에서좋은것모습을보였습니다이뻐이뻐

단어 추출을 통한
토크나이징

단어열: [아이오아이, 는, 이번, 공연, 에서,
좋은, 것, 모습, 을, 보였습니다, 이뻐, 이뻐]

품사 추정을 통한
품사 사전 업데이트

명사 사전 += [아이오아이, ...]
동사 사전 += [잘했어용, ...]

품사 사전을 이용한
품사 판별

품사열: [(아이오아이, 명사), (는, 조사), (이번, 명사), (공연, 명사),
(에서, 조사), (좋은, 형용사), (것, 명사), (모습, 명사), (을, 조사),
(보였습니다, 동사), (이뻐, 형용사), (이뻐, 형용사)]

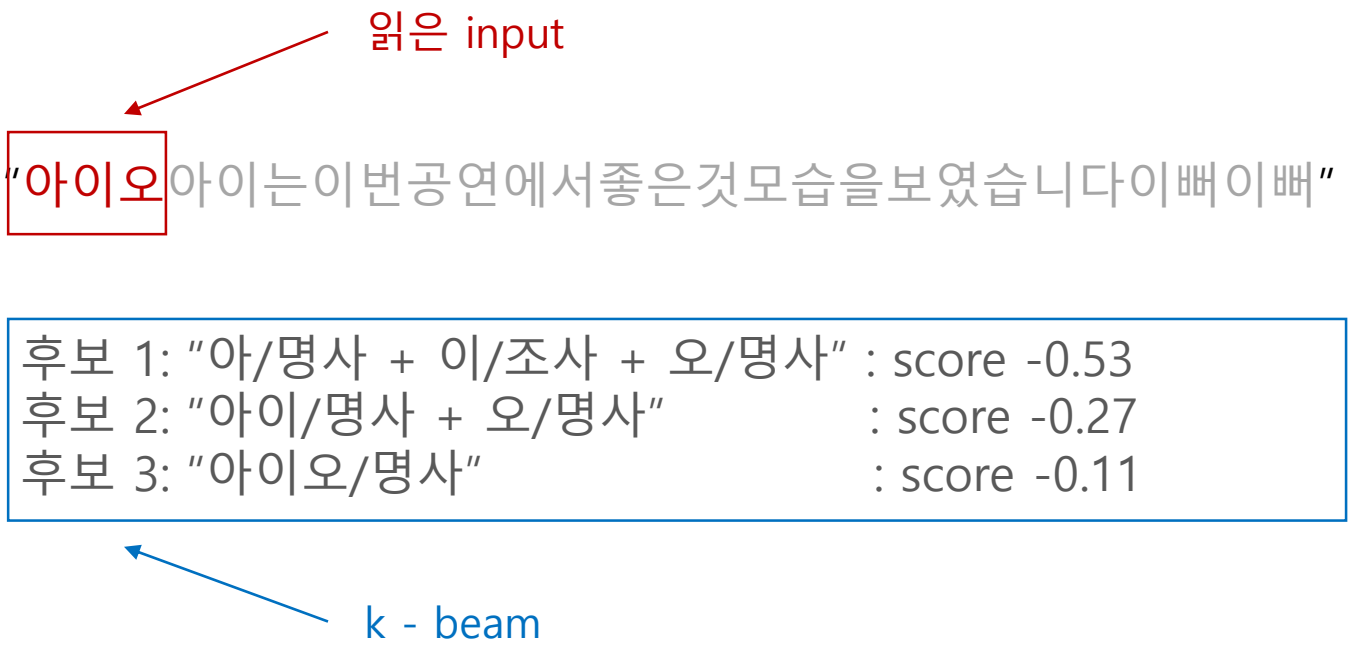
후처리

품사 판별

- 사전 기반 품사 판별은 세 가지 과정으로 구성되어 있습니다.
  - **1 단계: 후보 생성**
    - 사전을 이용하여 문장에서 가능한 품사열 후보를 만듭니다
    - 가능성이 적은 후보들을 제거한다면 계산 속도가 빨라집니다
  - **2 단계: 후보 평가**
    - 후보들 중에서 가장 적절한 품사열을 선택합니다
  - **3 단계: 후처리**
    - 사전에 포함되지 않는 단어들 처리 및 그 외의 후처리를 수행합니다

품사 판별

- Finite State Model 처럼 순차적으로 후보를 만들 수도 있습니다.



품사 판별

- 하지만 Max Score Tokenizer 와 같이, **알고 있는 단어부터 품사 판별**을 수행하도록 하였습니다

  - 긴 문장이 주어지면 사람은 아는 단어부터 눈에 보입니다
  - **확신이 있는 단어부터 품사를 판별**합니다

품사 판별

Step 1: 어절의 "**명**사/**형**용사/**동**사/**부**사"를 사전과 매칭 합니다

- 단어가 겹치더라도 가능한 모든 후보를 만듭니다

```
sent = '아이오아이는이번공연에서좋은컁모습을보였습니다이뻐이뻐'
candidates = _initialize_L(sent)
```

```
[ ['아이',          'Noun',   0, 2],
  ['아이오',        'Noun',   0, 3],
  ['아이오아이', 'Noun',   0, 5],
  ['이오',          'Noun',   1, 3],
  ['아이',          'Noun',   3, 5],
  ['이는',          'Verb',   4, 6],
  ['이번',          'Noun',   6, 8],
  ['공연',          'Noun',   8, 10],
    … ]
```

[단어, 품사, 시작 index, 종료 index]

품사 판별

Step 2: 포함 관계에 있는 **같은 품사의 단어**중, **가장 긴 것**만 남깁니다

```
sent = '아이오아이는이번공연에서좋은컁모습을보였습니다이뻐이뻐'
candidates = _remove_l_subsets(candidates)
```

```
[ ['아이',        'Noun',   0, 2],
  ['아이오',      'Noun',   0, 3],
  ['아이오아이',  'Noun',   0, 5],
  ['아오',        'Noun',   1, 3],
  ['아이',        'Noun',   3, 5],
  ['이는',        'Verb',   4, 6],
  ['이번',        'Noun',   6, 8],
  ['공연',        'Noun',   8, 10],
    … ]
```

'아이오아이'에 포함되는 모든 명사는 제거합니다

"이는/Verb"는 '아이오아이'와 다른 품사이며, 포함되지 않습니다

품사 판별

Step 3: “조사/형용사/동사”를 **확장**합니다

- 조사, 어미보다 명사/형용사/동사/부사를 잘 인식하는 것이 중요합니다

```
sent = '아이오아이는이번공연에서좋은컁모습을보였습니다이뻐이뻐'
candidates = _initialize_LR(sent, candidates)
```

```
[ [('아이오아이', 'Noun'), ('', ''),          0, 5, 5],
  [('아이오아이', 'Noun'), ('는', 'Josa'),    0, 5, 6],
  [('이는', 'Verb'),       ('', ''),          4, 6, 6],
  [('이번', 'Noun'),       ('', ''),          6, 8, 8],
  [('공연', 'Noun'),       ('', ''),          8, 10, 10],
  [('공연', 'Noun'),       ('에', 'Josa'),    8, 10, 11],
  [('공연', 'Noun'),       ('에서', 'Josa'),  8, 10, 12],
  …
]
```

품사 판별

Step 4: 확장된 단어 중 같은 품사는 가장 긴 것만 남깁니다

```
sent = '아이오아이는이번공연에서좋은컁모습을보였습니다이뻐이뻐'
candidates = _remove_r_subsets(candidates)
```

```
[ [('아이오아이', 'Noun'), ('', ''),          0, 5, 5],
  [('아이오아이', 'Noun'), ('는', 'Josa'),    0, 5, 6],
  [('이는', 'Verb'),       ('', ''),          4, 6, 6],
  [('이번', 'Noun'),       ('', ''),          6, 8, 8],
  [('공연', 'Noun'),       ('', ''),          8, 10, 10],
  [('공연', 'Noun'),       ('에', 'Josa'),    8, 10, 11],
  [('공연', 'Noun'),       ('에서', 'Josa'), 8, 10, 12],
  …
]
```

품사 판별

Step 4: 최종 후보입니다

```
sent = '아이오아이는이번공연에서좋은컁모습을보였습니다이뻐이뻐'
candidates = _initialize(sent)
```

```
[[('아이오아이', 'Noun'), ('는', 'Josa'),    0, 6, 6],
 [('아이오아이', 'Noun'), ('', ''),          0, 5, 5],
 [('이는', 'Verb'),       ('', ''),          4, 6, 2],
 [('이번', 'Noun'),       ('', ''),          6, 8, 2],
 [('공연', 'Noun'),       ('에서', 'Josa'), 8, 12, 4],
 [('공연', 'Noun'),       ('', ''),          8, 10, 2],
 [('좋은', 'Adjective'),  ('', ''),          12, 14, 2],
 [('모습', 'Noun'),       ('을', 'Josa'),   15, 18, 3],
 [('모습', 'Noun'),       ('', ''),         15, 17, 2],
 [('보였습니다', 'Verb'), ('', ''),          18, 23, 5],
 [('다이', 'Noun'),       ('', ''),          22, 24, 2],
 … ]
```

품사 판별

Step 5: “L + R”을 고려하여 **scoring**을 합니다

```
sent = '아이오아이는이번공연에서좋은컁모습을보였습니다이뻐이뻐'
scores = _scoring(candidates)
```

```
[[('아이오아이', 'Noun'), ('는', 'Josa'),    0, 6, 6,   3.39],
 [('아이오아이', 'Noun'), ('', ''),          0, 5, 5,   2.54],
 [('이는', 'Verb'),       ('', ''),          4, 6, 2,   1.90],
 [('이번', 'Noun'),       ('', ''),          6, 8, 2,   2.24],
 [('공연', 'Noun'),       ('에서', 'Josa'), 8, 12, 4,   2.77],
 [('공연', 'Noun'),       ('', ''),          8, 10, 2,  1.73],
 [('좋은', 'Adjective'),  ('', ''),          12, 14, 2, 2.25],
 [('모습', 'Noun'),       ('을', 'Josa'),   15, 18, 3,  3.26],
 [('모습', 'Noun'),       ('', ''),         15, 17, 2,  2.01],
 [('보였습니다', 'Verb'), ('', ''),          18, 23, 5, 2.21],
 [('다이', 'Noun'),       ('', ''),          22, 24, 2, 1.11],
 … ]
```

‘아이오아이/명사 + 는/조사’ 점수

품사 판별

Step 5: 점수 계산 feature를 만든 뒤, weight를 곱하여 scoring을 합니다

```
profile = OrderedDict([

        ('cohesion_l', 0.5),
        ('droprate_l', 0.5),
        ('log_count_l', 0.1),

        ('prob_l2r', 0.1),
        ('log_count_l2r', 0.1),
        ('known_LR', 1.0),

        ('R_is_syllable', -0.1),
        ('log_length', 0.5)
])
```

L의 cohesion score * 0.5
+ L의 droprate score * 0.5
+ L의 log 빈도수 * 0.1

+ 분석텍스트의 P(L → R) * 0.1
+ 분석텍스트의 Freq(L → R) * 0.1
+ L과 R이 모두 알려진 품사 * 1.0

+ 1음절 조사/어미 * -0.1
+ “L+R” 길이의 log * 0.5

품사 판별

Step 5: 점수 계산 feature를 만든 뒤, **weight를 곱**하여 **scoring**을 합니다

```
profile = OrderedDict([

        ('cohesion_l', 0.5),
        ('droprate_l', 0.5),
        ('log_count_l', 0.1),
        ('log_count_l2r', 0.1),
        ('log_count_l2r', 0.1),
        ('known_LR', 1.0),

        ('R_is_syllable', -0.1),
        ('log_length', 0.5)
])
```

분석하려는 텍스트 도메인의 특성을 반영하기 위한 장치

```
   L의 cohesion score          * 0.5
+  L의 droprate score          * 0.5
+  L의 log 빈도수              * 0.1

+  분석텍스트의 P(L → R)       * 0.1
+  분석텍스트의 Freq(L → R)    * 0.1
+  L과 R이 모두 알려진 품사    * 1.0

+ 1음절 조사/어미               * -0.1
+ "L+R" 길이의 log             * 0.5
```

품사 판별

Step 6: 높은 점수의 단어부터 품사를 부여 / 겹치는 부분은 제거합니다

```
sent = '아이오아이는이번공연에서좋은컁모습을보였습니다이뻐이뻐'
words = _find_best(scores)
```

```
[ [('아이오아이', 'Noun'), ('는', 'Josa'), 0, 6, 6, 3.39],
  [('모습', 'Noun'), ('을', 'Josa'), 15, 18, 3, 3.26],
  [('공연', 'Noun'), ('에서', 'Josa'), 8, 12, 4, 2.77],
 [('아이오아이', 'Noun'), ('', ''), 0, 5, 5, 2.54],
  [('좋은', 'Adjective'), ('', ''), 12, 14, 2, 2.25],
  [('이번', 'Noun'), ('', ''), 6, 8, 2, 2.24],
  [('보였습니다', 'Verb'), ('', ''), 18, 23, 5, 2.21],
 [('모습', 'Noun'), ('', ''), 15, 17, 2, 2.01],
 [('아는', 'Verb'), ('', ''), 4, 6, 2, 1.90],
 [('공연', 'Noun'), ('', ''), 8, 10, 2, 1.73],
 [('다아', 'Noun'), ('', ''), 22, 24, 2, 1.11],
  ... ]
```

품사 판별

Step 7: 사전에 등록되지 않은 단어는 아직 인식되지 않았습니다

```
sent = '아이오아이는이번공연에서좋은컁모습을보였습니다이뻐이뻐'
```

```
[('아이오아이', 'Noun'),
 ('는', 'Josa'),
 ('이번', 'Noun'),
 ('공연', 'Noun'),
 ('에서', 'Josa'),
 ('좋은', 'Adjective'),
 ('모습', 'Noun'),
 ('을', 'Josa'),
 ('보였습니다', 'Verb'),
 ('이뻐', 'Adjective'),
 ('이뻐', 'Adjective')]
```

품사 판별

Step 7: 사전에 없는 단어로 구성된 sub-sentence를 **후처리** 합니다

```
sent = '아이오아이는이번공연에서좋은컁모습을보였습니다이뻐이뻐'
```

```
[('아이오아이', 'Noun'),
 ('는', 'Josa'),
 ('이번', 'Noun'),
 ('공연', 'Noun'),
 ('에서', 'Josa'),
 ('좋은', 'Adjective'),
 ('컁', None),
 ('모습', 'Noun'),
 ('을', 'Josa'),
 ('보였습니다', 'Verb'),
 ('이뻐', 'Adjective'),
 ('이뻐', 'Adjective')]
```

품사 판별

Step 7: 길이가 긴 부분은 Max Score Tokenizer로 토크나이징까지 합니다

```
sent = '아이오아이는이번공연에서좋은모습을보였습니다앙순이들이죠아'
```

```
[('아이오아이', 'Noun'),
 ('는', 'Josa'),
 ('이번', 'Noun'),
 ('공연', 'Noun'),
 ('에서', 'Josa'),
 ('좋은', 'Adjective'),
 ('모습', 'Noun'),
 ('을', 'Josa'),
 ('보였습니다', 'Verb'),
 ('앙순이들', None),
 ('이', None),
 ('죠아', None)]
```

- 품사 사전에 ['앙순이들', '죠아']가 등록되어 있지 않더라도 가능한 단어로 나눠둡니다
- "앙순이들 + 이"에서 조사 사전을 바탕으로 "앙순이들"의 품사를 추정하는 것은 현재 개발중입니다

품사 판별 성능: vs. 트위터 한국어 분석기

```
twitter.pos(sent)
```

```
Process time: 102 ms

[('아이오', 'Noun'),
 ('아이', 'Noun'),
 ('는', 'Josa'),
 ('이번', 'Noun'),
 ('공연', 'Noun'),
 ('에서', 'Josa'),
 ('좋', 'Adjective'),
 ('은', 'Eomi'),
 ('컁', 'Noun'),
 ('모습', 'Noun'),
 ('을', 'Josa'),
 ('보였', 'Verb'),
 ('습니다', 'Eomi'),
 ('이뻐', 'Adjective'),
 ('이뻐', 'Adjective')]
```

```
proposed.pos(sent)
```

```
Process time: 3.05 ms

[('아이오아이', 'Noun'),
 ('는', 'Josa'),
 ('이번', 'Noun'),
 ('공연', 'Noun'),
 ('에서', 'Josa'),
 ('좋은', 'Adjective'),
 ('컁', None),
 ('모습', 'Noun'),
 ('을', 'Josa'),
 ('보였습니다', 'Verb'),
 ('이뻐', 'Adjective'),
 ('이뻐', 'Adjective')]
```

- 알고리즘은 예외가 발생하며, 사용자는 예외를 쉽게 수정하고 싶어합니다

  - 사전의 단어 추가 및 삭제
  - 반드시 보존하고 싶은 단어의 손쉬운 보호

사전의 단어 추가 및 삭제

- 컴파일을 다시 하지 않으면서 단어를 추가/삭제 해야 합니다

```
from soynlp.pos import LRMaxScoreTagger


my_dictionary_folders=['folder1', 'folder2']
tagger = LRMaxScoreTagger(my_dictionary_folders)


tagger.add_words_into_dictionary( ['아이오아이'], 'Noun')
tagger.remove_words_from_dictionary( ['아이오아이'], 'Noun')
```

반드시 보존하고 싶은 단어의 손쉬운 보호

- 도매인의 키워드, 혹은 중요한 단어들은 선호도를 설정합니다

```
tagger. set_word_preferance( ['아이오아이', '너무너무너무'], 'Noun', 10)
```

```
[[('아이오아이', 'Noun'), ('는', 'Josa'),    0, 6, 6,    3.39],
 [('아이오아이', 'Noun'), (', '),            0, 5, 5,    2.54],
 [('이는', 'Verb'),        (', '),            4, 6, 2,    1.90],
 … ]
```

```
[[('아이오아이', 'Noun'), ('는', 'Josa'),    0, 6, 6,    13.39],
 [('아이오아이', 'Noun'), (', '),            0, 5, 5,    12.54],
 [('이는', 'Verb'),        (', '),            4, 6, 2,    1.90],
 … ]
```

다른 단어로 인식될 가능성을
원천 봉쇄