한글 토크나이징 라이브러리 모듈 분석

이재경 · 서진범 · 조영복*

대전대학교

Analysis of the Korean Tokenizing Library Module

Jae-kyung Lee · Jin-beom Seo · Young-bok Cho*

Daejeon University

E-mail : forever408916@gmail.com

요 약

현재 자연어 처리(NLP)에 대한 연구는 급속히 발전하고 있다. 자연어 처리는 인간이 일상생활에서 사용하는 언어의 의미를 분석하여 컴퓨터가 처리할 수 있도록 하는 기술로 음성인식, 맞춤법 검사, 텍스트 분류 등 여러 분야에 사용하고 있다. 현재 가장 많이 사용되는 자연어처리 라이브러리는 영어를 기준으로 한 NLTK로 한글처리에 단점을 가지고 있다. 따라서 본 논문에서는 한글 토크나이징(Tokenizing) 라이브러리인 KonLPy와 Soynlp를 소개 후 형태소 분석 및 처리 기법을 분석하고, KonLPy의 단점을 보완한 Soynlp와의 모듈을 비교·분석하여 향후 의료분야에 적합한 자연어 처리 모델로 활용하고자 한다.

ABSTRACT

Currently, research on natural language processing (NLP) is rapidly evolving. Natural language processing is a technology that allows computers to analyze the meanings of languages used in everyday life, and is used in various fields such as speech recognition, spelling tests, and text classification. Currently, the most commonly used natural language processing library is NLTK based on English, which has a disadvantage in Korean language processing. Therefore, after introducing KonLPy and Soynlp, the Korean Tokenizing libraries, we will analyze morphology analysis and processing techniques, compare and analyze modules with Soynlp that complement KonLPy's shortcomings, and use them as natural language processing models.

키워드

NLP, NLTK, KoNLPy, Soynlp

Ⅰ. 서 론

최근 빅데이터 분석과 인공지능을 활용한 자연어 처리의 요구가 증대되고 있다. 인공지능을 기반으로 만든 자연어 처리(NLP)는 컴퓨터를 이용해 사람의 자연어를 분석하고 처리하는 기술을 뜻한다. 자연어 처리는 감정분석 · 카테고리 세분화 · 관련 상품 추천 등의 여러 분야에 사용되고 있다. 자연어 처리를 하기 위해서는 텍스트를 숫자로 변환하는 Word Embedding 과정이 필요하다. 여기서 Word Embedding의 가장 대표적인 모델 중 하나인 word2vec은 한국어 처리에 적용할 때 적합하여 최근 인기를 끌고 있다[1, 2].

자연어 처리에는 다양한 토크나이징 도구들이 이 있으며, 작업 수행에 따라 영어 토크나이징 라이브러리와 한글 토크나이징 라이브러리로 구분된다. NLTK는 가장 많이 쓰이는 영어 토크나이징 라이브러리로 텍스트 전처리 작업이 가능하며 50여개가 넘는 말뭉치 리소스를 활용해 영어 텍스트를 분석할 수 있게 제공한다. NLTK는 영어 자연

* corresponding author

어 처리에는 불편함 없이 사용가능하나, 영어를 기준으로 만들어졌기에 한글 자연어 처리를 할 때는 다소 미흡한 부분이 보인다. 이 점을 보완하기 위해 만들어진 KonLPy는 한글 토크나이징 라이브러리로, 한글 자연어 처리에 적합하며. 5개의 형태소 분석기를 포함하고 있다. Soynlp는 미등록 단어 인식이 되지 않는 KoNLPy의 단점을 보완하여 나온 모델로, 미등록 단어를 번거롭게 사전에 등록할 필요가 없는 비지도학습 기반 모델이다.

본 논문에서는 파이썬에서 가장 많이 사용되는 영어 라이브러리 모듈인 NLTK와 한글 라이브러리 모듈인 KoNLPy와 Soynlp를 각각 소개 및 비교 분석하여 향후 의료분야에 적합한 자연어 처리 모델로 활용하고자 한다.

II. 라이브러리 모듈 소개

2-1 NLTK

NLTK는 자연어 처리를 위한 Python 패키지로 자연어 분석 작업을 위해 만든 샘플 문서 집합인 50여개의 말뭉치와 긴 문자열의 분석을 위한 작은 단위로 나누는 작업인 토큰 생성을 제공하며, 단어로부터 어근·접두사·접미사·품사 등 형태소를 찾아내는 작업인 형태소 분석, 품사를 다는 작업인 품사 태깅을 제공한다. 또한 NLTK는 N Poster, snowball 등 다양한 스테밍 알고리즘과 그 외, chucking, NER, classfication 알고리즘을 내장하고 있다. 영어와 한국어 자연어 처리 모두 가능하지만, 영어를 기준으로 만들어진 토크나이징 라이브러리이기에 한국어 처리를 하기엔 미숙한 부분을 보여준다[3].

2-2 KoNLPy

KoNLPy는 C++, JAVA 등으로 구현된 형태소 분석기들을 모아 파이썬에서 사용 가능 하도록 만든 오픈 패키지로, 자연어 처리에서 형태소를 분리하는 데이터 전처리가 필요할 때 많이 사용되며, 라이센스에 따라 자유롭게 코드를 이용할 수 있다.

KoNLPy는 Mecab·kkma·Hannanum·Okt·Komoran 5개의 형태소 분석기를 통합적으로 지원한다. KoNLPy는 띄어쓰기가 없어도 토큰화 작업이 된다는 장점은 있지만, 고유 명사 추출 및 미등록 단어 인식이 힘들고 아직 완벽하지 않다는 단점을 가지고 있다[1, 4].

2-3 Soynlp

Soynlp는 KoNLPy와 같이 한국어 처리를 위한 파이썬 패키지 중 하나로, 학습데이터를 이용하지 않으면서 데이터에 존재하는 단어를 찾거나 문장을 단어열로 분해 또는 품사 판별을 할 수 있는 비지도학습 접근법을 사용한다. 비지도학습 접근법은 통계적 패턴을 이용하여 단어를 추출하기 때문에 하나의 문장보다는 어느 정도 규모가 있는 동일한 집단의 문서에서 효율적으로 작동한다[5].

Soynlp는 기본적으로 Python 3.5+버전을 지원하며 Noun Extracter, Word Extracter, Tokenizer 등 많은 기능을 제공한다[5]. Tokenizer는 어미, 조사 등이 붙어 구분이 불가능한 한국어를 명사, 동사, 어미, 조사 등으로 형태소를 분석해 Word2vec에 적합한 데이터를 만드는 과정을 뜻한다[1].

Soynlp는 비지도학습 방법을 통해 KoNLPy가 지니던 미등록 단어 인식 문제는 해결하였으나, 띄어쓰기가 없으면 토큰화가 제대로 되지 않는다는 단점이 있다[5].

III. 자연어 처리를 위한라이브러리

3-1 KoNLPy

KoNLPy는 한국어 정보처리를 위한 파이썬 패키지로, 형태소 분석기를 중심으로 구축되었으며, 간단한 말뭉치를 제공하기도 한다[4]. 본 논문에서는 KoNLPy의 패키지에 포함되어 있는 Okt 형태소 분석기를 사용하여 형태소 분석을 한 후, 장단점에 대해 설명하고자 한다.

```
1 from konlpy.tag import Okt
2 okt = Okt()

1 x = okt.morphs("한국어 자연어 처리는 konlpy가 쉽고 간편하다")
2 print(x)

['한국어', '자연어', '처리', '는', 'konlpy', '가', '쉽고', '간편하다']
```

그림 1 morphs 메서드를 이용한 형태소 처리

그림 1과 같이 okt 형태소 분석기와 morphs 메서드를 이용하여 문장을 형태소 단위로 나눌 수 있다.

```
1 from konlpy.tag import Okt
2 okt = Okt()

1 x = okt.pos("한국어 자연어 처리는 konlpy가 쉽고 간편하다")
2 print(x)

('한국어', 'Noun'), ('자연어', 'Noun'), ('처리', 'Noun'), ('는', 'Josa'), ('konlpy', 'Alpha')
```

그림 2 pos 메서드를 이용한 형태소 처리

그림 2와 같이 Okt 형태소 분석기와 pos 메서드를 이용하여 형태소 단위로 나누고, 형태소의 종류까지 나타낼 수 있다.

```
from konlpy.tag import Okt
okt = Okt()

x = okt.pos("조정조서는 조정에 관하여 상황을 기재하는 것")
print(x)
```

```
[('조정', 'Noun'), ('조서', 'Noun'), ('는', 'Josa'), ('조정', 'Noun'), ('에', 'Josa')]
```

그림 3. 미등록 단어를 포함한 문장 형태소 처리

그림 3과 같이 미등록 단어인 "조정조서"를 Okt 형태소 분석기를 통해 처리하게 되면, 명사 추출 오류가 발생한다.

3-2 Soynlp

Soynlp는 한국어 분석을 위한 파이썬 패키지로, 학습데이터를 이용하지 않고 비지도학습 접근법을 이용한다. Soynlp에서 사용가능한 L-토큰화는 여러 가지 길이의 L 토큰의 점수를 비교하여 가장 점수가 높은 L단어를 찾는 것을 말한다[5]. 본 논문에서는 Soynlp의 비지도학습 접근법을 이용하여, 한국어 형태소 분석 및 장단점에 대해 설명하고자 한다.

```
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("이 모듈은 단점을 보완해서 나온 모델", flatten=False)
```

```
[('이', ''), ('모듈은', ''), ('단점을', ''), ('보완해서', ''), ('나온', ''), ('모델', '')]
```

그림 4. L-토큰화를 이용한 형태소 처리

그림 4는 LTokenizer 클래스를 사용하여 L 토큰+R 토큰 구조인 문자열을 서로 각각 나누어, 형태소를 구분하는 L-토큰화 방법이다[5]. 그림 4와 같이 띄어쓰기가 되어있는 문장은 품사 구분이 정확히 구분되는 모습을 보여준다.

```
from soynlp.tokenizer import MaxScoreTokenizer

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

```
['이모듈은단', '점을', '보완해서나온', '모델']
```

그림 5. 최대 점수 토큰화를 이용한 형태소 분석

그림 5는 띄어쓰기를 하지 않은 문장을, 최대 점수 토큰화를 이용하여 형태소를 분석하였다. 그러나 KoNLPy와 달리 띄어쓰기가 제대로 되지 않은 것을 확인할 수 있다.

IV. 결 론

본 논문에서는 영어 토크나이징 라이브러리인 NLTK, 한글 토크나이징 라이브러리인 KoNLPy와 Soynlp를 각각 소개한 후, KoNLPy와 Soynlp를 각각 형태소 분석을 하여, 장단점을 비교하였다. KoNLPy는 Okt 형태소 분석기와 메서드들을 이용하여, 형태소 분석을 편리하고 용이하게 사용할 수 있었으나 그림 3과 같이 미등록 단어 문제가 발생할 수 있다는 것을 확인할 수 있었다. Soynlp는 L-토큰화를 이용하여 완벽한 문장에는 형태소 처리에 문제가 없음을 보여주었으나, 그림 5와 같이 띄어쓰기가 되지 않은 문장에는 미흡한 모습을 보여주었다.

본 논문에서는 KoNLPy와 Soynlp의 형태소 분석을 통하여 장단점을 알아보았다. KoNLPy는 띄어쓰기가 없이 토큰화가 잘된다는 장점을 가진 반면 미등록 단어에 취약한 모습을 보였고, Soynlp는 미등록 단어를 간편하게 설정한다는 장점을 가진 반면에 띄어쓰기가 없는 문장에 취약한 모습을 보였다. 또한 두 가지 모델의 장단점을 비교하여 띄어쓰기가 없는 문장에 미흡한 모습을 보였으나, 미등록 단어 등록이 용이한 Soynlp를 의료분야에 적합하다고 판단하여 향후 치매 후보 추적물질 연구에 활용하고자 한다.

Acknowledgement

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. 2018R1C1B5083789).

References

[1] Dong-Woo Ko, Jung-Jin Yang, "Korean Natural Language Processing ang Analysis", *Korea Information Science Association,* pp. 2140-2142, June. 2018.

[2] Hyungsuc Kang, Janghoon Yang, "Optimization of Word2vec Models for Korean Word Embeddings", *Korea Digital Content Association,* Vol 20, No 4, pp. 825-833, April. 2010.

[3] NLTK 3.5 documentation, Natural Language Toolkit, https://www.nltk.org/

[4] Eun-Jeong Park, Seong-Jun Cho, "KoNLPy Korean natural language processing in Python" *Korean Language and Korean Information Processing Conference,* pp. 133-136, Oct. 2014.

[5] Soynlp, lovit, https://github.com/lovit/soynlp