NLTKo 1.0: 한국어 언어처리 도구

홍성태[O], 차정원

창원대학교 컴퓨터공학과
{ghdchlwls123, jcha}@changwon.ac.kr

Natural Language Toolkit _ Korean

Seong-Tae Hong[O], Jeong-Won Cha
Changwon National University

요 약

NLTKo는 한국어 분석 도구들을 NLTK에 결합하여 사용할 수 있게 만든 도구이다. NLTKo는 전처리 도구, 토크나이저, 형태소 분석기, 세종 의미사전, 분류 및 기계번역 성능 평가 도구를 추가로 제공한다. 이들은 기존의 NLTK 함수와 동일한 방법으로 사용할 수 있도록 구현하였다. 또한 세종 의미사전을 제공하여 한국어 동의어/반의어, 상/하위어 등을 제공한다. NLTKo는 한국어 자연어처리를 위한 교육에 도움이 될 것으로 믿는다.

주제어: 자연어 처리 도구

1. 서론

딥러닝 기술의 접목으로 인해 자연어처리가 현실에 사용할 수 있는 수준으로 발전을 거듭하여 다양한 응용분야에서 자연어처리 도구들이 활용되고 있다. 이에 따라 자연어처리 교육에 대한 요구도 증가되었다.

NLTK[1]는 Natural Language ToolKit의 약자로 교육용으로 개발된 자연어처리 및 문서 분석용 파이썬 패키지이고, 다양한 기능 및 예제들을 포함하고 있다. 주요한 기능으로는 토큰화, 형태소 분석, 구문 분석 및 의미 추론을 위한 텍스트 처리 라이브러리 및 예제로 활용할 수 있는 말뭉치와 워드넷(WordNet)[2]을 제공하고 있다. NLTK는 실무에서는 물론 연구 분야에서도 많이 사용되고 있으며, 많은 자연어처리 기능들이 NLTK의 모듈로 구현되어 있다. 그리고 NLTK는 언어 독립적으로 이용할 수 있는 메소드도 다수 제공하기 때문에 한국어 정보처리에도 유용하게 이용할 수 있다.

한국어는 영어와 차이점도 많이 존재하기 때문에 영어를 처리하기 위한 NLTK는 한국어 분석을 하는데 도움을 받을 수는 있지만 부족한 부분이 많다. 그렇기 때문에 한국어 정보 처리를 위한 라이브러리가 필요하며 이를 위해 NLTKo를 개발하고자 하였다. 표 1은 NLTK와 KoNLPy, KoalaNLP 및 NLTKo를 정보처리 기능과 제공하는 자원 등으로 구분한 것이다[3,4,5].

표 1 언어처리 도구

	형태소 분석	구문 분석	말뭉치	워드넷	세종 의미사전	전처리	평가 도구
NLTK	○	○	○	○	×	○	○
KoNLPy	○	×	○	×	×	×	×
KoalaNLP	○	○	×	×	×	×	×
NLTKo 1.0	○	×	×	○	○	○	○

NLTKo는 한국어 정보 처리를 위한 Toolkit으로, 한국어 정보처리에 사용되는 모든 기능들을 손쉽게 하나의 라이브러리에서 제공하도록 하는 것이 목적이다. 가장 큰 장점으로는 NLTK와 동일한 함수명을 사용하여 한국어 정보들을 손쉽게 사용할 수 있으며 세종의미사전을 활용하여 단어에 대한 다양한 정보들을 이용할 수 있다. 그림 1은 NLTK와 동일한 함수를 사용하여 토크나이징 결과를 보여주는 예시이다.

```
>>> from nltk.tokenize import word_tokenize, syllable_tokenize, sent_tokenize
>>> eng_text= "Hello, my name is 000. I am currently attending Changwon University."
>>> kor_text="안녕하세요 저는 000입니다. 창원대학교에 재학 중입니다."
>>> sent_tokenize(eng_text)
['Hello, my name is 000.', 'I am currently attending Changwon University.']
>>> sent_tokenize(kor_text,'korean')
['안녕하세요 저는 000입니다.', '창원대학교에 재학 중입니다.']
>>> word_tokenize(eng_text)
['Hello', ',', 'my', 'name', 'is', '000', '.', 'I', 'am', 'currently', 'attending', 'Changwon', 'University', '.']
>>> word_tokenize(kor_text, 'korean')
['안녕하세요', '저는', '000입니다.', '창원대학교에', '재학', '중입니다.']
```

그림 1 토크나이저 사용 예시

분류 모델과 문장 생성 모델 결과의 정량 평가를 적용할 수 있도록 각 성능 지표에 대한 기능들도 추가하였다. 기계학습 데이터를 수집하여 데이터 정제 과정은 학습 전 선행되어야 할 필수 요소이며, 자연어 처리 라이브러리를 사용해 진행한다. NLTKo는 언어 처리와 관련된 전반적인 모듈들을 하나의 라이브러리로 사용함으로 자연어처리 연구자들이 쉽게 사용할 수 있을 것이다.

2. 한국어 언어처리 도구

NLTKo는 korChar, eval, metric 모듈 및 sejong 패키지를 추가하였으며 tag, tokenize 패키지를 수정하여 기존의 NLTK 함수와 동일하게 사용 가능하게 만들었다. sejong 패키지는 체언 및 용언 상세사전을 포함하고 있다.

2.1 세종 전자사전

NLTKo는 21세기 세종계획의 의미사전을 이용하여 단어의 파생어, 복합어, 숙어, 동의어, 반의어, 동위어, 상위어, 하위어, 전체어, 부분어, 관련어, 영어, 예시, 형용사/명사/동사 결합, 선택 제약, 경로, 유사도 정보를 제공한다. 각 단어의 상세사전 xml tag에 따른 획득 정보에 따라 사용 가능한 기능들을 구분하였다. 그림 2는 체언 및 용언 상세사전에서 활용한 태그들의 구조를 나타낸다. 문형 구성 구획은 체언에서만 획득 가능하며 동사 정보 구획은 용언에서 활용 가능하다.

```
superEntry                  최상위 표제항 구획
    entry                   표제항 구획
        morph_grp           형태 정보 구획
            comp            합성어 형성
            der             파생어 형성
        idm_grp             숙어 구획
            idm             숙어
        sense               센스 구획
            sem_grp         의미 정보 구획
                sem_class   의미 부류
                trans       영어 대역어
                lr          어휘 의미 관계 구획
                    syn     동의어
                    ant     반의어
                    hyper   상위어
                    hypo    하위어
                    holo    전체어
                    mero    부분어
                    rel     관련어
            frame_grp       문형 구성 구획 (체언)
                frame       문형
                subsense    하위 센스 구획
                sel_rst     선택 제약
                eg          용례
            syn_grp         통사 정보 구획 (용언)
                comb_aj     형용사 결합 정보
                comb_n      명사 결합 정보
                comb_v      동사 결합 정보 구획
                    form    동사 결합 형태
                    frame   문형
```

그림 2 체언 및 용언 상세전자사전의 미시 구조

superEntry는 동형어 분할 이전의 표제항을 제시하는 구획이므로 하나의 superEntry는 동형어 구분이 된 표제항 구획을 나타내는 entry를 여러 개 가질 수 있다. sense는 표제항이 갖는 각각의 형태 의미를 구별하여 의미 정보 및 통사 정보를 제공하는데 entry는 여러 개의 sense를 가질 수 있으므로 NLTKo에서 entry와 sense를 object로 표현하였다. 그림 3은 기입된 단어의 표제항 구획 리스트의 목록을 확인하고 특정 entry의 복합어와 숙어 정보를 확인하는 예제 코드이다.

```
>>> from nltk.sejong import ssem
>>> word = '이'
>>> entrys=ssem.entrys(word)
>>> entrys
[Entry('이.nng_s.1'), Entry('이.nng_s.2'),
Entry('이.nng_s.3'), Entry('이.nng_s.4')]
>>> entrys[0].senses()
[Sense('이.nng_s.1.1'), Sense('이.nng_s.1.2')]
>>> entrys[0].comp()[:5]
['이ㅅ몸', '이뿌리', '틀이', '덧이', '송곳이']
>>> entrys[0].idm()[:4]
['이가 갈리다', '이를 갈다', '이를 악물다', '이 빠진 호랑이']
```

그림 3 표제항 구획 이하 태그 활용 예시

"단어.품사 태그.엔트리 번호.센스 번호"를 통하여 해당 센스에 직접 접근 가능하다. 그림 4는 sense 구획에 직접 접근하여 표제항 이하의 해당 단어의 sense에 접근하여 의미 부류, 동의어, 동위어, 상위어, 하위어, 관련어, 영어 대역어, 동사 결합 정보 및 세종 의미사전 의미 분류 체계를 이용한 경로 정보 사용 예시이다.

```
>>> from nltk.sejong import ssem
>>> sense=ssem.sense('이.nng_s.1.1')
>>> sense.sem()
['신체부위']
>>> sense.syn()
['치아', '이빨']
>>> sense.coord()
['손', '발', '눈썹', '손톱', '머리카락']
>>> sense.hyper()
['신체부위']
>>> sense.hypo()
['틀', '덧', '송곳', '어금', '사랑', '앞ㄴ이;충치']
>>> sense.rel()
['치과', '치통', '틀니']
>>> sense.trans()
['tooth']
>>> sense.comb_v()[:5]
['이가 나다', '이가 들뜨다', '이가 쑤시다', '이가 썩다', '이가 빠지다']
>>> sense.sem_path()
['1_구체물', '1.4_관계구체물', '1.4.1_부분', '1.4.1.2_신체부위']
```

그림 4 센스 구획 이하 태그 활용 예시

2.2 성능측정

NLTKo는 분류 모델의 성능을 평가하는 분류 성능 평가 지표와 기계 번역 모델의 성능을 평가하는 기계번역 평가 지표를 함께 제공한다.

2.2.1 기계번역 평가

기계 번역과 같이 문장 간의 품질을 측정하는데 사용

되는 BLEU[6], ROUGE[7], METEOR[8], WER/CER, CIDER[9] 평가 지표들을 제공한다. 영어 NLTK에서는 WordNet을 이용한 METEOR Score를 제공하며, NTLKo에서는 세종 전자사전의 동의어 정보를 활용한 METEOR Score를 제공한다. 그림 5는 두 문장 사이의 품질 측정 예시이다.

```
>>> from nltk import eval
>>> can="농산물 가격이 추석에 가까워질수록 가파르게 상승했다."
>>> ref="한가위가 가까워질수록 농산물 가격이 가파르게 올랐다."
>>> eval.bleu_n([ref],[can],1)
0.666666666666667
>>> eval.rouge_l([ref],can)
0.5
>>> eval.wer(ref,can)
0.8333333333333334
>>> eval.cider([ref],[can])
0.2166667
>>> eval.meteor([ref],can)
0.5260416666666665
```

그림 5 기계번역 평가지표 사용 예시

2.2.2 분류 평가

분류 모델의 예측력 검증 및 평가를 위하여 평가 지표 Accuracy, Precision, Recall, F1_Score 4가지를 제공한다. 그림 6은 분류 평가 모듈 사용 예시이다. Macro, Micro-average 모두 사용 가능하다.

```
>>> from nltk import metric
>>> y_true=[0,0,1,1,2,0,1,2,2,1,1,0,1]
>>> y_pred=[0,1,0,2,0,0,2,2,2,0,1,0,2]
>>> metric.accuracy_score(y_true,y_pred)
0.46153846153846156
>>> metric.precision_score(y_true,y_pred,avg="macro")
0.4666666666666666
>>> metric.recall_score(y_true,y_pred,avg="macro")
0.5277777777777778
>>> metric.f1_score(y_true,y_pred,avg="macro")
0.4953445065176908
```

그림 6 분류 평가지표 사용 예시

2.3 한국어 형태소 분석 및 띄워쓰기

NLTKo는 딥러닝을 이용한 한국어 형태소 분석기, 품사 태깅, 보조 동사 결합 태깅, 명사 추출, 띄워쓰기 기능을 제공한다. NLTK의 형태소 분석 함수와 동일하게 사용 가능하다. 그림 7은 품사 태깅 사용 방법 예제 코드이다.

```
>>> from nltk import pos_tag, nouns, word_segmentor,
pos_tag_with_verb_form
>>> from nltk.tokenize import syllable_tokenize
>>> sentence = "추석에 반가운 얼굴들을 만났다."
>>> sentence2 = "한가위가가까워질수록농산물가격이가파르게올랐다."
>>> pos_tag(syllable_tokenize(sentence,'kor'),lang='kor')
[('추석', 'NN'), ('에', 'JJ'), ('반갑', 'VB'), ('ㄴ', 'EE'),
('얼굴', 'NN'), ('들', 'XN'), ('을', 'JJ'), ('만나', 'VB'),
('았다', 'EE'), ('.', 'SY')]
>>> pos_tag_with_verb_form(sentence)
[('추석', 'NN'), ('에', 'JJ'), ('반갑', 'VB'), ('ㄴ', 'EE'),
('얼굴', 'NN'), ('들', 'XN'), ('을', 'JJ'), ('만나', 'VB'),
('았다', 'EE'), ('.', 'SY')]
>>> nouns(sentence)
['추석', '얼굴']
>>> word_segmentor(sentence2)
['한가위가', '가까워질수록', '농산물', '가격이', '가파르게', '올랐다']
```

그림 7 품사 태깅 사용 예시

2.4 한국어 전처리

각 문자의 한글, 영어, 한자, 숫자, 기호, 구두점, 연결 문자 판별 기능과 입력 한글 문자의 분할, 초/중/종성의 결합, 입력된 두 문자의 비교 기능을 제공한다. 그림 8은 전처리 모듈 사용 예시이다.

```
>>> from nltk import korChar
>>> korChar.kor_check('ㄱ')
True
>>> korChar.kor_syllable('가')
True
>>> korChar.kor_split('국')
('ㄱ', 'ㅜ', 'ㄱ')
>>> korChar.kor_join('ㄲ','ㅜ','ㅁ')
'꿈'
>>> korChar.hanja_syllable('韓')
True
>>> korChar.num_syllable('1')
True
>>> korChar.eng_syllable('a')
True
>>> korChar.symbol_check('★')
True
>>> korChar.punctuation_check('.')
True
>>> korChar.engConnection_check('_')
True
>>> korChar.numConnection_check('.')
True
```

그림 8 한국어 전처리 모듈 사용 예시

3. 결론

현재 NLTKo에서 제공하는 체언 및 용언 상세 의미사전의 개수는 표 2와 같다. 이를 의존명사, 대명사, 수사,

부사, 관형사 등의 추가와 동시에 기초 전자사전까지 확장한다면 사전 항목의 규모가 60만 항목을 넘어서므로 대규모 어휘들을 다룰 수 있을 것이다.

표 2 NLTKo 내 단어 파일 개수

구분		개수
체언	명사	23,016
용언	형용사	3,451
	동사	12,102
합계		38,569

기존의 NLTK에 한국어 언어처리를 위한 여러 패키지 및 모듈들을 추가하여 언어별로 각기 다른 도구들을 이용하면서 사용자에게 불편함을 제공하던 이전의 방식에서 한/영 간의 분리된 라이브러리가 아닌 통합된 인터페이스를 제공하고 기존 NTLK에 있던 함수명과 공유함으로써 새로 익히는 시간을 절약하므로 보다 더 편리하고 효율적인 이용이 가능할 것으로 예측된다. 다음 버전에서는 구문분석, 개체명 분석, SRL(Semantic Role Labeling)을 추가하여 공개할 예정이다.

Acknowledgement

이 논문은 2021년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No.2021-0-00354, 비정형 텍스트를 학습하여 쟁점별 사실과 논리적 근거 추론이 가능한 인공지능 원천기술)

참고문헌

[1] Steven B ird, Edw ard Loper, NLTK: The Natural Language Toolkit, ACL, pp. 214-217, 2004
[2] “WordNet”, https://wordnet.princeton.edu/
[3] NAM, Gyu-Hyeon; LEE, Hyun-Young; KANG, Seung-Shik. KoNLTK: Korean Natural Language Toolkit. In: Annual Conference on Human and Language Technology. Human and Language Technology, 2018. p. 611-613.
[4] Eunjeong L. Park, Sungzoon Cho. “KoNLPy: Korean natural language p rocessing in Python”, Proceedings of the 26th Annual Conference on Human & Cognitive Language Technology, Chuncheon, Korea, Oct 2014.
[5] “KoalaNLP”, https://koalanlp.github.io/python-support/html/#
[6] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu, BLEU: a Method for Automatic Evaluation of Machine Translation, ACL, pp. 311-318, 2002.
[7] Chin-Yew Lin, ROUGE: A Package f or Automatic Evaluation of Summaries, ACL, pp.74-81, 2004
[8] Satanjeev B anerjee, A lon Lavie, METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments, ACL, pp. 65-72, 2005.
[9] Ramakrishna V edantam, C. Lawrence Z itnick, Devi Parikh, CIDEr: Consensus-based Image Description Evaluation, CVPR ,2015.