KoNLPy: 쉽고 간결한 한국어 정보처리 파이썬 패키지

박은정○, 조성준
서울대학교 산업공학부
ejpark04@snu.ac.kr, zoon@snu.ac.kr

KoNLPy: Korean natural language processing in Python

Eunjeong L. Park○, Sungzoon Cho
Seoul National University, Industrial Engineering Department

요 약

파이썬은 간결한 아름다움을 추구하는 동시에 강력한 스트링 연산이 가능한 언어다. KoNLPy는 그러한 특장점을 살려, 파이썬으로 한국어 정보처리를 할 수 있게 하는 패키지이다. 꼬꼬마, 한나눔, MeCab-ko 등 국내외에서 개발된 여러 형태소 분석기를 포함하고, 자연어처리에 필요한 각종 사전, 말뭉치, 도구 및 다양한 튜토리얼을 포함하여 누구나 손쉽게 한국어 분석을 할 수 있도록 만들었다.

주제어: 말뭉치 언어학, 말뭉치 활용 도구, 파이썬

1. 서론

최근 전산언어학에서 말뭉치의 활용이 높아지면서 다양한 오픈소스 라이브러리와 패키지가 공개되고 있다. 스탠포드 대학에서 개발한 Stanford NLP 도구[8]부터 시작해서, OpenNLP[3], LingPipe[1], MALLET[9] 등 유명한 라이브러리 들이 있다. 2006년경에는 전산언어학의 교육을 더욱 용이하게 하기 위해 NLTK[2] 같은 패키지도 생겨나게 되었다.

한국어 정보처리를 위한 오픈소스 라이브러리도 이미 1995년부터 KTS[14]를 시작으로 다수 배포되고 있다. 그 중 대다수는 C/C++, Java 등으로 개발이 되어 있으며, 형태소 분석기가 중심이다[7, 13, 6]. 한편 프로그램의 실행 속도 등이 중요한 제품 개발의 관점에서 봤을 때는 이들도 좋은 선택이지만, 교육, 또는 비전공자 연구의 관점에서는 프로그래밍 언어를 습득하는데 들이는 노력을 줄일 수 있는 언어를 선택하고 싶을 수 있다.

본 논문에서는 이러한 관점 하에서 파이썬으로 한국어 정보처리를 할 수 있는 패키지인 KoNLPy(Korean NLP in Python)를 제안한다. KoNLPy는 파이썬의 범언어적 자연어처리 패키지 NLTK와 마찬가지로, 자연어처리를 갓 배우기 시작한 학생이나 자연어처리를 연구 목적으로 사용하려는 연구자에게 적합하다.

KoNLPy는 http://konlpy.org에서 찾을 수 있으며, GNU GPL 라이센스를 채택하고 있다. 또한 윈도우, 리눅스, 맥 OS 등 파이썬이 지원하는 운영체제에서 사용할 수 있으며, 파이썬 2와 파이썬 3로 사용 가능하다.[1)]

본 논문의 구조는 다음과 같다. 먼저 2장에서는 KoNLPy를 설계하고 개발하는데 도움이 된 다양한 오픈소스 라이브러리를 소개한다. 3장에서는 KoNLPy의 설계 철학을 제시한 후, 4장에서 KoNLPy를 간단하게 이용하는 방법에 대해 소개한다. 5장은 KoNLPy를 다른 패키지와 함께 이용할 수 있는 확장적 사용 방법에 대해 논의하며, 마지막으로 6장에서 요약 및 결론을 내린다.

2. 관련 연구

2.1. 한국어 정보처리 패키지

한국어 정보처리 패키지 중 오픈소스로 공개되었고 빈번히 사용되는 것들을 모아 표 1에 정리하였다.

[표 1] 여러 오픈소스 한국어 형태소 분석기와 개발 언어.

이름	연도	언어	라이센스
KTS [14]	1995	C/C++	GPL v2
한나눔 [13]	1999	Java	GPL v3
MACH [11]	2002	C/C++	custom
Arirang	2009	Java	Apache v2
꼬꼬마 [7]	2010	Java	GPL v2
KoNLP [5]	2011	R	GPL v3
MeCab-ko [6]	2013	C/C++	GPL v2, LGPL, BSD
KOMORAN	2013	Java	custom

이 중 상당수는 C/C++ 또는 자바로 개발되었고, 각종 프로그래밍 언어에 대한 플러그인(plug-in)을 제공하는

1) 2014년 9월 현재, 버전 0.3.3이 공개되어 있다.

MeCab를 제외하고는 지원하는 언어가 개발 언어와 같다.[2)]

2.2. 자연어처리 파이썬 패키지, NLTK

NLTK[2]는 본디 교육용으로 개발된 파이썬 패키지이다. Porter, snowball, Lancaster 등 다양한 스테밍(stemming) 알고리즘과 그 외 chunking, NER, classification 알고리즘을 내장하고 있으며 50개가 넘는 다양한 언어의 말뭉치도 패키지 안에 포함되어 있다. 또한 문서가 풍부한 예제로 가득 차 있고, 사용하는 플랫폼과 무관하게 사용할 수 있어서 많은 이용자를 확보하고 있다.

게다가 NLTK는 언어독립적(language independent)으로 이용할 수 있는 메소드도 다수 제공하기 때문에 한국어 정보처리에도 유용하게 이용할 수 있다.

3. 설계 철학

KoNLPy는 파이썬 프로그래밍 언어로 한국어 정보처리를 할 수 있게 한다. 파이썬을 이용해 한국어 정보처리를 지원하는 것은 다음의 이점을 가진다.

첫째, 프로그래밍 초심자도 쉽게 시작할 수 있다. 1999년, 파이썬을 발명한 귀도 반 로썸(Guido van Rossum)이 DARPA에 보낸 서신만 봐도 알 수 있듯, 파이썬은 애초에 교육용으로 고안된 소프트웨어이다.[12] 그만큼 중고등학생이나 컴퓨터 비전공 학생도 쉽게 습득하고 사용할 수 있도록 설계 단계에서부터 고안되었다는 것이다. "실행 가능한 의사코드(executable pseudocode)"라는 별명을 가지고 있을 정도로 코드가 직관적이며, 최근 이러한 이점 때문에 미국 상위권 대학을 중심으로 초심자 대상의 프로그래밍 교육을 파이썬으로 하기 시작했다는 보고도 있다.[4] 한국어 정보처리 소프트웨어는 실제로 전산언어학 전공자 뿐 아니라 언어학, 교육학, 사회학, 정치학 등의 연구에도 널리 사용된다. 이 때, 비전공자가 프로그래밍 언어, 혹은 패키지 사용법을 학습하는데 투자하는 시간을 줄여줄수록, 연구 분야 자체에 투자할 수 있는 시간은 많아진다.

둘째, 빠른 스크립팅이 가능하다. 특히 연구자들의 경우 환경과 파라미터를 다양하게 바꿔가며 실험을 해야하는데, 스크립팅이 용이하고 인터프리터로 직접 코드를 돌릴 수 있는 환경은 연구자의 시간을 아껴준다. 뿐만 아니라, OOP를 지원하면서도 그것을 강제하지 않는 것은 큰 이점이다.

셋째, PyPI에 공개된 파이썬의 풍부한 라이브러리들 덕에 다양한 태스크를 하나의 언어로 모두 처리할 수 있다는 장점을 가지고 있다. 가령 requests, lxml 등의 라이브러리를 이용해 웹문서를 크롤링 해온 후, KoNLPy를 이용해 한국어 처리를 해서, matplotlib 등의 시각화 라이브러리로 그래프를 그릴 수 있다.

KoNLPy는, 이와 같은 파이썬의 이점 위에 다음의 설계 철학을 바탕으로 개발되었다.

쉽고 간단한 사용법. 직관적인 함수명을 사용한다. 매뉴얼을 따로 읽는데 많은 시간을 보내지 않아도 설치를 받자마자 바로 실행해서 결과를 볼 수 있게 한다.

확장가능성. 형태소 분석기 뿐 아니라 다양한 자연어 처리 기능과 말뭉치를 포괄하는 것을 목표로 하며, 따라서 말뭉치, 메소드 등이 추가되는 것을 감안하여 개발을 진행한다. NLTK가 다양한 스테머(stemmer)를 지원하는 것과 마찬가지로, 여러 형태소 분석기 중에서 목적과 취향에 맞는 것을 쉽게 선택할 수 있도록 한다.

친절하고 상세한 문서. 패키지에서 중요도가 다소 저평가될 수 있는 부분이 문서화인데, 사실 상세한 문서와 풍부한 예제는 초심자에게 가장 큰 도움이 되는 부분이기도 하다. 특히 NLTK, Gensim[10] 등 다른 파이썬 패키지들과 함께 사용할 때, 어떻게 응용하여 사용할 수 있는지 보여주는 것도 중요하다.

개방과 공유. 패키지는 사용 환경이 계속 변할 수 있기 때문에 지속가능성(sustainability)을 유지할 수 있는지가 중요하다. 또한, 실질적인 사용자의 니즈(needs)를 파악하기 위해서는 누구나 개발에 참여를 할 수 있게 하는 것도 중요하다. 따라서 본 패키지의 소스는 온라인에 공개하여 참여를 독려하고 있다.[3)]

4. 간단한 한국어 정보처리

4.1. 용례검색

KoNLPy에서 특정 문구의 용례를 찾는 방법은 아래와 같다. 아래는 대한민국 헌법에서 "대한민국"이라는 문구가 어디에서 등장하는지 찾는 코드이다. 이는 konlpy.utils안에서 concordance 메소드를 통해 찾을 수 있다. kolaw는 실험적 용도로 KoNLPy에 내장된 대한민국 법률 말뭉치이며, 대한민국 헌법이 'constitution.txt'라는 파일로 들어있다.

2) MeCab는 자바, 파이썬, 루비, 펄을 지원하는 플러그인이 있다.

3) https://github.com/e9t/konlpy

```
>>> from konlpy.corpus import kolaw
>>> from konlpy.utils import concordance
>>> doc = kolaw.open('constitution.txt').read()
>>> concordance(u'대한민국', doc, show=True)
0       대한민국헌법 유구한 역사와
9       대한국민은 3·1운동으로 건립된 대한민국임시정부의
98      총강 제1조 ① 대한민국은 민주공화국이다. ②대한민국의
100     ① 대한민국은 민주공화국이다. ②대한민국의 주권은
110     나온다. 제2조 ① 대한민국의 국민이 되는
126     의무를 진다. 제3조 대한민국의 영토는 한반도와
133     부속도서로 한다. 제4조 대한민국은 통일을 지향하며,
147     추진한다. 제5조 ① 대한민국은 국제평화의 유지에
787     군무원이 아닌 국민은 대한민국의 영역안에서는 중대한
1836    파견 또는 외국군대의 대한민국 영역안에서의 주류에
3620    경제 제119조 ① 대한민국의 경제질서는 개인과
[0, 9, 98, 100, 110, 126, 133, 147, 787, 1836, 3620]
```

4.2. 형태소 분석과 품사 태깅

KoNLPy 0.3.3에서는 꼬꼬마, 한나눔, MeCab-ko 등 세 가지 형태소 분석기를 사용할 수 있다. 각각 konlpy.tag에서 Kkma, Hannanum, Mecab 클래스를 호출하여 사용할 수 있으며, 모두 동일한 형태의 입력을 받아 동일한 형태의 출력을 받을 수 있도록 통일된 인터페이스를 사용한다. 다음은 그 중에서 꼬꼬마를 사용한 예시이다.[4)]

```
>>> from konlpy.tag import Kkma
>>> from konlpy.utils import pprint
>>> kkma = Kkma()
>>> pprint(kkma.pos(u'한국어 분석은 재밌습니다!'))
[(한국어, NNG),
 (분석, NNG),
 (은, JX),
 (재밌, VA),
 (습니다, EFN),
 (!, SF)]
```

4.3. 최빈 형태소 찾기

형태소 찾기를 하고 나면 가장 빈번하게 등장하는 형태소가 뭔지 찾고 싶을 수 있다. 이럴 때는 collections의 Counter 클래스를 이용할 수 있다.

```
>>> from collectiosn import Counter
>>> import konlpy
>>> doc = konlpy.corpus.kolaw.open('constitution.txt').read()
>>> pos = konlpy.tag.Hannanum().pos(doc)
>>> cnt = Counter(pos)
>>> konlpy.utils.pprint(cnt.most_common(5))
[((의, J), 398),
 ((., S), 340),
 ((하, X), 297),
 ((에, J), 283),
 ((ㄴ다, E), 242)]
```

4) 파이썬 3에서는 pprint를 사용할 필요가 없다. 파이썬 2에서는 리스트 형태를 출력하면 콘솔에 유니코드로 나타나기 때문에 편의를 위해 함수가 추가되었다.

5. KoNLPy와 다른 패키지 함께 사용하기

파이썬의 패키지 인덱스인 PyPI에는 KoNLPy를 비롯하여 상당수의 서드파티(3rd party) 라이브러리들이 있다. 다국어 분석이 가능한 NLTK나, 토픽모델링에 사용되는 Gensim, 시각화를 위한 matplotlib는 그 중 일부일 뿐이다. 이들을 KoNLPy와 동시에 활용하면 더욱 다양한 분석을 할 수 있다.[5)]

여기서는 KoNLPy의 Mecab 클래스를 이용해 제 25회 한글 및 한국어 정보처리 학술대회 논문집 (2013) 본문에서 명사를 추출하여 pytagcloud 패키지를 이용해 30줄 이내의 파이썬 스크립트로 [그림 1]의 워드클라우드를 생성한다.

```
1  from collections import Counter
2  import random; random.seed(0)
3  import webbrowser
4
5  from konlpy.tag import Mecab
6  import pytagcloud
7
8  r = lambda: random.randint(0,255)
9  color = lambda: (r(), r(), r())
10
11 # Read file
12 with open('2013-hclt.txt', 'r') as f:
13     text = f.read().decode('utf-8')
14
15 # Get nouns and count
16 m = Mecab()
17 nouns = m.nouns(text)
18 count = Counter(nouns)
19
20 # Draw word cloud
21 tags = [{'color': color(), 'tag': n, 'size': c/10}\
22                 for n, c in count.most_common(50)]
23 pytagcloud.create_tag_image(\
24     tags, 'wordcloud.png', size=(800, 600),
25     fontname='Noto Sans CJK')
26 webbrowser.open('wordcloud.png')
```

줄 1-6에서 관련 패키지를 불러온 후, 줄 8-9에서 색을 랜덤으로 생성하는 함수를 선언한다. 그 후 줄 11-13에서 텍스트 파일을 불러와 줄 16-18에서 빈도 높은 명사를 찾는다. 여기서는 KoNLPy의 Mecab 클래스를 사용했지만, Kkma, Hannanum 클래스로 바꿔 사용하여 결과물을 손쉽게 비교해볼 수도 있다. 줄 21-22에서는 상위 빈도 50개의 명사를 pytagcloud에 맞는 포맷으로 변환하는 과정이며, 색, 태그(명사), 태그의 크기를 전달한다. 마지막으로 줄 23-26에서 pytagcloud를 이용해 이미지를 생성한 후, 사용자의 화면에 생성된 이미지를 띄운다.[6)]

5) http://konlpy.org/ko/latest/examples/에서 더 많은 예제를 확인할 수 있다.

6) 이 때, pytagcloud에 한글 폰트를 추가해서 사용한다. 자세한 내용은 http://konlpy.org/ko/latest/examples/wordcloud/를 참고하자.

[그림 1] KoNLPy의 Mecab 클래스를 이용해 2013년 한국어 정보처리 학회 논문집에서 가장 높은 빈도로 등장하는 50개의 명사로 그린 워드클라우드.

6. 결론

앞으로는 초중고등학교에서 국어, 수학 등 뿐 아니라, 소프트웨어 교육도 의무화가 된다고 하여 어떻게 하면 그 교육이 효과적이고 효율적일 수 있을지에 대한 논의가 활발하다. 논의의 중심에 있는 이슈 중에는, 어떤 프로그래밍 언어나 소프트웨어를 이용해서 교육을 진행할 것인가도 포함된다. 그런 측면에서 쉽게 습득하고, 빠르게 사용할 수 있는 파이썬 패키지 KoNLPy는 한국어 정보처리를 가르치는데 좋은 도구가 될 수 있을 것이다.

앞으로 KoNLPy의 각종 형태소 분석 클래스에서 사용하는 사전 인터페이스를 통일하고 사용자 사전을 추가하는 모듈을 추가한다면 커스터마이즈(customize)된 사전도 손쉽게 사용할 수 있게 된다. 또한, konlpy.download() 함수를 제공하면 말뭉치, 사전 등 방대한 데이터 파일을 소스코드와 분리하여 별도로 다운로드 받을 수 있어 편리함이 높아질 수 있을 것이다.

참고문헌

[1] Alias-i, "Lingpipe", http://alias-i.com/lingpipe. [Accessed: 2014-09-29].

[2] S. Bird, "NLTK: the natural language toolkit", In Proceedings of the COLING/ACL on Interactive presentation sessions, pp.69-72, Association for Computational Linguistics, 2006.

[3] Apache Software Foundation, "Apache OpenNLP", http://opennlp.apache.org. [Accessed: 2014-09-29].

[4] P. Guo, "Python is now the most popular introductory teaching language at top U.S. universities", http://cacm.acm.org/blogs/blog-cacm/176450. [Accessed: 2014-09-12].

[5] H. Jeon, "KoNLP: Korean NLP package", http://cran.r-project.org/web/packages/KoNLP/index.html, 2013. [Accessed: 2014-09-29].

[6] T. Kudo, K. Yamamoto, and Y. Matsumoto, "Applying conditional random fields to Japanese morphological analysis", In Proceedings of the Conference on Empirical Methods on Natural Language Processing (EMNLP), volume 4, pp.230-237, 2004.

[7] D. Lee, J. Yeon, I. Hwang, and S. Lee, "꼬꼬마: 한글 형태소 분석기 (KKMA: Kool Korean morpheme analyzer)", http://kkma.snu.ac.kr. [Accessed: 2014-09-12].

[8] C.D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S.J. Bethard, and D. McClosky, "The Stanford CoreNLP natural language processing toolkit", In Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pp.55-60, 2014.

[9] A.K. McCallum, "Mallet: A machine learning for language toolkit", http://mallet.cs.umass.edu. [Accessed: 2014-09-29].

[10] R. Rehurek and P. Sojka, "Software Framework for Topic Modelling with Large Corpora", In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, pp.45-50, Valletta, Malta, May 2010.

[11] K. Shim and J. Yang, "Mach: A supersonic Korean morphological analyzer", In Proceedings of the 19th International Conference on Computational Linguistics (COLING-2002), pp.939-945, 2002.

[12] G. van Rossum, "Computer programming for everybody", http://python.org/doc/essays/cp4e. [Accessed: 2014-09-12].

[13] 박상원, 최동현, 김은경, 최기선, "플러그인 컴포넌트 기반의 한국어 형태소 분석기", 한글 및 한국어 정보처리 학술대회 논문집, pp.197-201, 2010.

[14] 이상호, 서정연, 오영환, "KTS: 미등록어를 고려한 한국어 품사 태깅 시스템", 음성통신 및 신호처리 워크샵 논문집, 한국음향학회.