「정보시스템연구」 제25권 제4호
한국정보시스템학회
2016년 12월, pp. 121～139

https://doi.org/10.5859/KAIS.2016.25.4.121

오픈소스 소프트웨어를 활용한 자연어 처리 패키지 제작에 관한 연구

이 종 화* · 이 현 규**

〈목 차〉

Ⅰ. 서 론

영화 속처럼 인간의 기관들을 인공적으로 개조하거나 이식하여 가상적인 인조인간의 캐릭터가 등장하곤 한다. 현대인들은 모바일기기를 통해 네트워크 세상을 읽고 표현하고 감성을 입히고 생활한다. 인간의 기관에 직접적으로 이식한 것은 아니지만 이젠 인간의 신체 한 부분인 것처럼 모바일과 네트워크를 통해서 세상을 보고 읽고 감성을 표현한다. 이러한 ICT 환경에서의 느낌이나 감성을 자연어로 표현하는 것은 많은 기업들이 소비자의 니즈를 분석하는 재료로 사용되고 있다(장영재, 2015). 사람들이 사회생활에서 자유롭게 표현하는 방대한 말에서 의미나 대화들을 컴퓨터로 분석하는 작업들이 자연어 처리이다. 단어들의 형태소 분석을 하는 기본적 알고리즘 과정들은 C언어나 Java, Python에서 제공되는 라이브러리로 처리되고 있다.

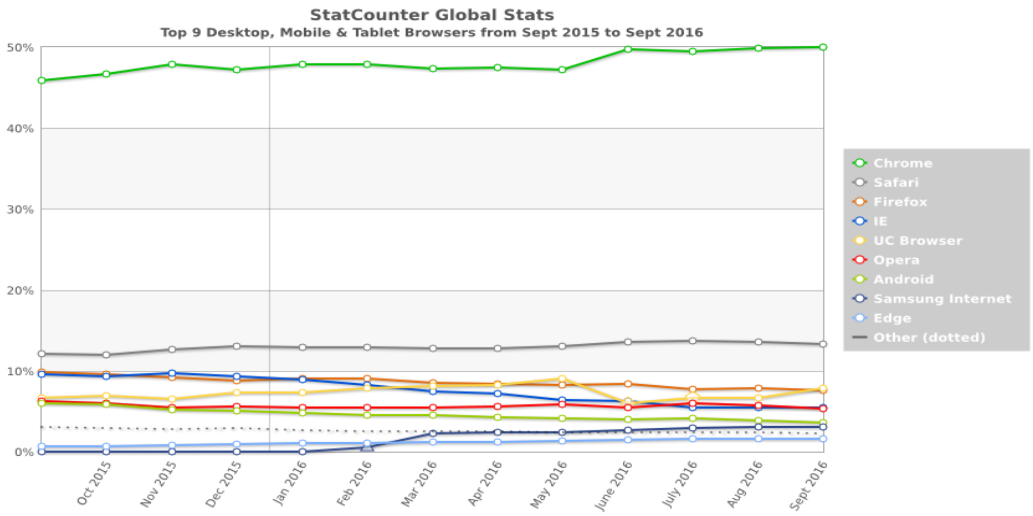
자바 개발 환경인 JDK(Java Development Kit)는 2007년에 소스 코드가 오픈되면서, 소프트웨어 개발 키트(SDK)로 발전되어 현재 가장 널리 사용되고 있다. SDK는 자바 ME, SE, EE 및 FX 등의 개발 환경을 제공하며 Linux, Mac, Windows 등 다양한 운영체제 환경에서 개발되고 있다. 또한, C언어 환경에서 개발된 유닉스(Unix)와 유닉스 기반 개인용 컴퓨터 공개 운영체제인 리눅스(Linux)는 소스 코드를 공개하여 프로그램 개발자나 전공자 내에서 사용이 확대되고 있다(권순창, 2007; 김상현 · 송영미, 2009; 김성용 · 이상민, 2008). 이와 같은 배경에서 정

* 부경대학교 경영학부 박사수료, newjwcom@daum.net, 주저자
** 부경대학교 경영학부 교수, hyunqlee@pknu.ac.kr, 교신저자

보통신산업진흥원(2013)은 리눅스 커널 개발을 통해 서버가상화 기술과 통합전산센터에 공개소프트웨어 적용 기술 지원, 커뮤니티 지원, 인력 양성 등 국내 오픈소스 소프트웨어 정책을 추진하고 있다. 즉, 개인 사용자 환경을 좌우하는 운영체제 시장을 오픈소스 소프트웨어가 더욱 빠르게 팽창시키고 있는 것이다(문상식 · 김홍기, 2014; http://www.nipa.kr).

오픈소스 소프트웨어(Open Source Software)는 프로그램 소스를 공개하였으며, 이는 이후 개발자의 자유로운 소프트웨어 사용 및 추가 개발 · 응용을 용이하게 해 주었다. 이와 더불어 웹 2.0 환경을 통해 다수의 개발자들이 서로 협력하고 참여하여 제작한 결과물들은 집단지성을 이루고 있다. 이러한 추세는 운영체제 점유율에서 잘 나타난다. 프로그램 소스 내용을 공개하지 않고 구매하는 방식의 독점 소프트웨어(Proprietary software)인 마이크로소프트사의 운영체제 Windows는 지난 2008년엔 95.3%를 차지했지만 2013년 조사한 결과로는 75%를 차지함으로써 윈도우 시장이 20% 가파른 감소 추세에 있다. 또한, 모바일의 등장으로 오픈소스 운영체제인 안드로이드가 5%, Linux가 1.5%를 차지하고 있다(문상식 · 김홍기, 2014).

<그림 1> Stat Counter Global Stats에 따르면 2016년 9월 Desktop, Tablet, Mobile등에 탑재된 웹브라우저 시장점유율은 구글의 Chrome이 55%를 넘어섰고 모질라의 Firefox는 10%대를 유지하였으며, 마이크로소프트사의 인터넷 익스플로러(Internet Explorer), 애플의 사파리(Safari)가 그 뒤를 잇는다. 마이크로소프트사의 인터넷 익스플로러(Internet Explorer)는 액티브엑스(ActiveX)의 무분별한 사용으로 인하여 감소 추세에 있으며, 애플의 사파리는 Mac 운영체제의 브라우저로 빠른 검색 성능과 웹 응용프로그램간의 반응성이 높은 것으로 나타났다. 하지만 운영체제간의 호환성이 낮고 구글의 Chrome에 비해 편리한 개발 환경을 제공하지 못하여 다소 시장점유율이 낮은 것으로 나타났다(http://gs.statcounter.com/).



<그림 1> 웹브라우저(Web Browser)의 시장점유율 흐름

이처럼 개발과 사용의 개방성(오픈 환경: openness)은 그 중요성이 커지고 있으며, 정보의 홍수 속에서 데이터 분석과 통계, 그리고 그 데이터에서 새로운 의미를 찾아내는 데이터마이닝의 분야로 좁혀 생각하면 이는 더욱 주목해 볼 만하다. 이러한 현상은 오픈환경은 사용자가 자유롭게 스스로에게 필요한 데이터를 유연하게 분석할 수 있는 사용자 중심의 환경이기 때문이라고 볼 수 있다. 이 중에서 오픈소스로서 무료로 제공되는 R은 통계분석, 시뮬레이션 등에 뛰어난 소프트웨어이다. R은 처음 다루는 사람도 쉽게 접근할 수 있는 간단한 계산을 기반으로 하며, 집단 지성을 통해 빠른 속도로 늘어나는 패키지들과 기본 함수를 활용하면 큰 어려움 없이 코딩이 가능하다. 또한, 윈도우용 R, Mac용 R, Linux용 R로 서비스를 제공하고 프로그래밍의 효율성을 위하여 R studio 에디터 환경을 별도 지원한다. 추가적으로, 효율적으로 소프트웨어를 개발하기 위한 코드 편집기, 디버그, 도움말 지원 기능으로 어플리케이션 인터페이스를 제공하며 특히, 빅데이터 통계분석과 그래픽용 프로그래밍언어로 많은 주목을 받고 있다(Lee & Lee, 2015; 김용현・허의남, 2014; 사공원 등, 2016).

본 연구는 이와 같은 개발 및 사용 환경의 중요성을 인지하고, 데이터마이닝을 통해 자연어 처리의 성능과 효과성을 검증하고 소스 코드를 공개하고자 한다. 기존의 자연어 처리 패키지보다 훨씬 유용하며 효과적인 오픈 소스 통계분석용 소프트웨어인 R 프로그램의 진보된 자연어 처리 패키지의 알고리즘을 연구하였다. 특히, 본 연구가 자연어의 처리과정에서 주목하는 패키지인 “KoNLP”는 소셜 데이터를 연구하는 많은 연구자들의 시간과 노력을 줄여주었다. 그러나 소셜 데이터의 비중이 커지면서 한글의 무분별한 표현이 자연어 처리의 어려움을 가중시키고, 품사 태깅의 오류 발생으로 명사 추출 함수의 비정상적 처리가 이루어지는 것을 발견하였다. 따라서 본 연구는 기존에 개발된 KoNLP() 패키지를 활용하여 명사를 추출하는 extractNoun 명령과 단어들의 형태소를 9개의 품사로 분리 가능한 SimplePos09 함수의 주요 기능을 분석하고, 새로운 통합 명사 처리 패키지(new_Noun)를 제공함으로써 한글 자연어 처리에 도움이 되고자 한다.

Ⅱ. 선행 연구

2.1 자연어 처리

자연어 처리란 일상생활에서 자연스럽게 사용하는 언어들을 분석하여 컴퓨터가 이해할 수 있는 형태로 표현하는 것이다. 일상생활에서 표현되는 언어를 이해하고 자연어 분석 시 생성되는 문제를 다루는 형태소 분석, 구문분석, 화행분석, 대화처리 등 언어 처리 모델들은 자연어처리에 필요한 기반 기술들이다(한만휘 등, 2015, 안정국・김희웅, 2015). 이에 대한 분석 수단으로는 데이터 속에서 숨겨진 관계나 패턴을 탐색하여 의미 있는 정보를 변환하는 데이터마이닝과 텍스트의 주제를 판단하고 유용한 정보를 가공, 추출하는 텍스트마이닝, 텍스트에서 감성, 뉘앙스, 태도 등을 판단하여 의미 있는 정보로 변환하고 의사결정에 활용하는 과정을 오피니언 마이닝 등이 있다(Lee & Lee, 2016;

LE et al., 2015; Lee et al., 2015).

품사 부착(POS tagging) 기술에는 단계별 전이모델, 음절 단위 N-gram과 확률 모델 등이 있으며, 일반적으로 형태소 분석 후에 품사 부착이 일반적이다. 주어진 문장이 정의된 문법 구조에 따라 정당하게 하나의 문장으로 사용될 수 있는가를 확인하는 작업인 구문 분석(parsing) 기술은 스탠퍼드 구문분석기, 버클리 구분분석기 등이 있다(한만휘 등, 2016)

이러한 자연어 처리는 인공지능의 도움을 받아 진행할 수 있다. 인간보다 더 빠르고 넓은 사고와 학습 능력, 그리고 지식 데이터베이스를 갖고 있는 시스템들이 등장하고 있다. 복잡한 일들을 컴퓨터 프로그래밍을 통해 효과적으로 수행할 수 있도록 구현한 기술, 인공지능(Artificial Intelligence)이 그 중 하나이다. 인공지능은 수백 개나 수천 개의 '조건-시행문(if-then)'의 논리적 과정을 거쳐 특정 분야의 지식을 얻어내며, 이 과정에서 컴퓨터가 스스로 학습하며 지식체계를 축적하고 있다. 인공지능의 기술에는 사람의 언어를 이해하고 분석이 가능한 자연어 처리 기술과 그래픽 패턴을 분별해내는 이미지 인식 기술 등이 있다.

2.2 오픈 소스 소프트웨어

일반적인 개발 소프트웨어는 유료로 거래되며, 저작권이 있는 창작물이다보니 개발 환경과 기술의 핵심인 코딩 내용은 극비 내용이다. 하지만 오픈소스 소프트웨어는 누구나 자유롭게 소프트웨어를 코딩하여 유용한 기술을 공유하는 것이다. 집단지성으로 사용자들이 서로의 기술을 함께 사용하며 업데이트하는 것이다. 데이터 분석 도구인 "R" 또한 오픈 소스이며 통계 계산과 마이닝 처리에 많은 연구자들이 활용하고 있다. 특히, 선행 연구자의 패키지 등록으로 이후 연구자들이 함께 사용하며 등록된 패키지들은 전 세계 연구자에 의하여 업데이트 되고 있다(Cachia et al., 2007; Black, 2008). 또한, 등록된 패키지 수가 무려 1만여 개에 이르고 있다(이상준・이동훈, 2016; https://www.r-project.org/). 코딩 기술이 소프트웨어 기술을 넘어 하드웨어에도 결합되어 피지컬(Physical) 컴퓨팅으로 새로운 전성기를 맞고 있는(박정웅 등, 2014) 대표적 사례가 IoT(사물인터넷)이며, 다양한 ICT(정보통신기술)의 등장으로 더욱 확산될 전망이다. 특히, 전자 컨트롤러 보드에 오픈 소스 기반으로 개발한 도구와 환경을 적용하여 다양한 센서기술들을 제어할 수 있도록 설계된 오픈 하드웨어가 아두이노(Arduino)이다. 기존 농사 기술과 ICT를 융합하여 지능화된 농촌 마을을 만들려는 정부 추진 사업인 스마트 팜(Smart Farm) 사업(손수아・박석천, 2015)도 이와 같은 추세를 반영하는 대표적 사례이다. 스마트 팜을 통해 다양한 농작물 재배 시설의 온도, 습도, 이산화탄소, 토양 등을 측정하는 제어 장치부를 활용할 수 있고, 네트워크 기술, 웹 어플리케이션 기술 등을 융합하여 스마트한 경작을 할 수 있다. 이는 오픈소스가 한 몫을 담당하였기에 가능해진 것이다.(박정웅 등, 2014).

2.3 알고리즘

일상생활에서 코딩과 알고리즘은 바늘과 실과도 같다. 아래의 <문장1>, <문장2>는 무슨

차이가 있을까?

문장1> 그녀는 훌륭한 박물관을 운영하고 있어요.
문장2> 그녀는 박물관을 훌륭하게 운영하고 있어요.

단어의 순서에 따라 내용과 표현의 의도가 달라진다. 코딩이란 바로 이러한 차이를 이해하는 것이다. 코딩은 위 문장의 단어처럼 알고리즘의 한 요소이며 표현 도구라 할 수 있다. 단어에 그 의미가 있듯 서로 약속된 명령어인 코딩으로 시간적 흐름, 즉 알고리즘을 찾는 것이 중요하다. 그렇다면 알고리즘이란 무엇인가?

문장에서 단어의 배열이 의미 전달에 주요한 역할을 하듯 아래의 <문장3>에서 순서가 다르게 표현된다면 문제의 해답을 찾기 어려울 것이다.

문장3> 코끼리를 냉장고에 넣는 방법은?
냉장고 문을 연다.
코끼리를 넣는다.
문을 닫는다.

최적화된 문제의 해답을 찾기 위한 방법론들을 반복, 탐욕, 분할정복, 동적계획, 백트랙킹 등 5가지 알고리즘 기법으로 표현하였고, 스토리텔링을 활용한 문제 상황, 프로그램 코딩(구안), 코딩 수정 및 보안, 알고리즘 검토 및 정리 등 4단계의 알고리즘 구현 절차를 주장하였다(신수범, 2015).

본 연구의 진행 과정은 다음의 단계로 진행하였다. 기존 extractNoun() 함수의 알고리즘을 분석하고 비정상적으로 명사 추출이 제외되는 단어들의 2차 명사 판정을 위한 새로운 함수를 개발한다. 즉, 기존 extractNoun() 함수와 본 연구의 업그레이드된 알고리즘인 new_Noun() 함수를 이용하여 명사 추출 알고리즘을 구현한다. 또한 본 연구는 뉴스와 블로그 데이터를 활용하여 기존 명사 추출 방법과 품사 태그 처리 과정을 포함한 명사 추출 방법을 비교해 보고자 한다.

Ⅲ. 연구방법과 프레임워크

텍스트마이닝은 분석 주제에 맞는 텍스트 형태의 연구 대상 자료를 수집하고, 문서별 말뭉치인 Corpus 생성 이후 명사 추출 과정을 거친다. 하지만 연구 목적에 불필요한 불용어 및 기타 의미 없는 기호 제거 과정을 거치기도 한다. 그 이후에는, Corpus 내의 한글 형태소 단위를 인식하고 명사들의 빈도를 이용하여 다양한 Table 구조(Array, List, Matrix, Vector)의 선택과 유사한 문서의 그룹화를 거쳐서 키워드와 복합 명사들을 추출한다(Lee et al., 2015).

```
> extractNoun("한글날을 하루 앞두고 개최된 이 행사는 외국인 어린이들의 한글에 대한 흥미를 높이고 마련되었다.")
 [1] "한글날을" "하루"     "개최"     "행사"     "외국"     "어린이"   "들"       "한글에"   "흥미"     "마련"
>
```

<그림 2> 1차 명사 분리 작업(extractNoun)

```
> SimplePos09("한글날을")
$`한글날을`
[1] "한글날/N+을/J"
```

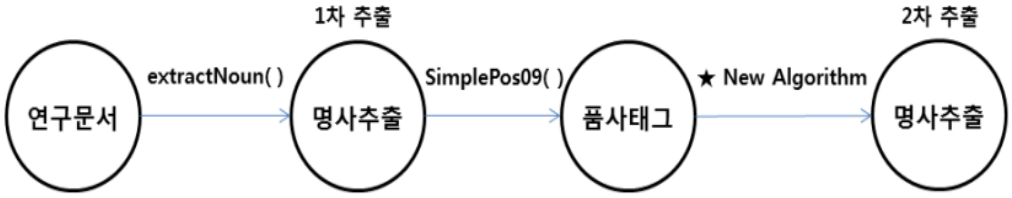
```
> SimplePos09("한글에")
$`한글에`
[1] "한글/N+에/J"
```

<그림 3> 품사 태킹 작업으로 명사 단어 확인(SimplePos09)

본 연구에서는 빅데이터 분석 도구인 R의 장점인 패키지를 재설계하려고 한다. 텍스트마이닝 처리의 기본 패키지이며 한글 자연어 처리에 많이 사용되는 KoNLP 패키지를 분석하였다. KoNLP 패키지의 함수는 명사 추출 사용자 단어 사전 등록 함수(mergeUserDic), 한글의 자음, 모음의 분리 함수(HangulAutomata), 한글을 아스키 문자를 기준으로 음절 단위 변환 함수(convertHangulStringToKeyStrokes), 한글을 자음, 모음 분리하여 음절 단위 변환 함수(convertHangulStringToJamos), 9품사 태그 처리 함수(SimplePos09), 문장에서 명사 추출 함수(extractNoun) 등의 다양한 30여개의 함수로 구성되어 있다(https://www.r-project.org/).

명사 추출용 extractNoun() 함수의 비정상적인 명사 처리로 연구 비교 대상에서 제외되는 경우가 발생하여 마이닝 결과에 영향을 주는 것으로 나타났다(그림 2). 그래서 명사 추출의 신뢰성을 높이기 위한 방법으로 명사 추출용 함수의 결과에 다시 형태소 태그 부착을 통한 여과망 역할의 SimplePos09() 함수를 적용해 보았다(그림 3).

본 연구는 명사 추출 함수인 extractNoun()와 품사 태그 처리 함수인 SimplePos09()을 구성하는 알고리즘과 두 함수의 결합 함수의 연구를 진행하였다. 먼저 기존 명사 추출 함수를 사용하여 1차 명사 추출 과정을 정상적으로 처리한다. 그 결과로 도출된 명사 단어들 중 <그림 2>의 “한글날을”, “한글에”와 같은 비정상적인 명사 추출 단어를 SimplePos09() 함수를 사용하여 걸러낸 후, 품사 태그를 이용하여 명사 단어 확인 철차를 추가로 하게 되는 것이다(그림 4, 5).



<그림 4> 품사 태그를 활용한 명사 추출 프레임워크

<그림 5> 본 연구에 사용된 분석 흐름도

<그림 6>에서는 본 연구에서 제시된 품사 태그를 활용한 명사 추출 알고리즘의 함수별 코딩 과정을 제시한다. 연구 대상 문서와 컴퓨터 개발 환경의 인코딩이 동일한지를 확인하기 위한 함수인 명사 추출 함수와 품사 태그 함수에 사용되는 전처리 함수들을 살펴볼 수 있다.

명사 추출 전처리 과정으로 연구 대상 문서 내 불필요한 공백 제거와 단어의 구분 없이 20 글자 이상 띄어쓰기 없는 비정상적 단어 추출을 목적으로 포함된 함수이다.

```
checkEncoding <- function(inputs){
  if(Encoding(inputs) == "unknown"){
    expectenc <- detectInputEncoding(inputs)
    if(is.null(expectenc)){
      return(F)
    }
    if(expectenc != localeToCharset()[1]){
      stop("Please check input encoding!")
    }
  }
  return(T)
}
# R 환경 인코딩과 연구 대상문서의 인코딩의 동일 유무를 판단하는 과정이다.
# detectInputEncoding() 함수를 통하여 연구 대상문서의 인코딩을 확인하고
  localeToCharset() 함수를 통하여 R 프로그래밍 환경의 인코딩을 확인할 수 있다.
```

<그림 6> checkEncoding() 함수 알고리즘

```
preprocessing <- function(inputs){
  newInput <- gsub("[[:space:]]", " ", inputs)
  if(nchar(newInput) > 20 & length(strsplit(newInput, " ")[[1]]) <= 3){
    warning(sprintf("It's not kind of right sentence : '%s'", inputs))
    return(FALSE)
  }
  return(newInput)
}
# gsub(찾을 값, 바꿀 값, 문자열)함수는 특정 문자열을 검색하여 지정된 문자로 바꾸는 함수로 [/@().,:], [&], [#], [[:space:]] 등의 특수문자는 특정문자로 치환하는 과정이다. 또한 원하지 않는 내용은 필터링한다([[:space:]] - 공백들).
# nchar함수는 데이터들의 글자 수를 확인 할 수 있는 매크로이며 newInput 내의 글자수가 20글자 초과하는지를 확인한다.
# strsplit함수는 글자별, 단어별, 문장별 등 분리 기준 문자 즉, 빈값(“”), 공백(“ ”)이나 마침표(“.”) 등을 넣어서 문자열을 분리해 내는 함수이며 조사 제거도 가능하다.
```

<그림 7> preprocessing() 함수 알고리즘

명사 추출과정으로 먼저 현 프로그램 개발 환경과 같은 인코딩 연구 대상 문서인지 확인하고 이후 불필요한 공백 제거, 및 비정상적 단어 추출 과정을 거친다. 이 과정은 <그림 8>에서 확인가능하다.

```
  if(!checkEncoding(sentence)){
    return(sentence)
  }
  if(!is.character(sentence) | nchar(sentence) == 0) {
    stop("Input must be legitimate character!")
  }else{
    sentence_pre <- preprocessing(sentence)
    if(sentence_pre == FALSE){
      return(sentence)
    }
      # 명사 처리 모듈 및 Tagging 과정 수행<그림 9>
  }
# checkEncoding() 함수에서 R 개발환경과 연구 대상문서의 인코딩을 유무를 확인한다.
# 문자형 자료와 연구 대상 문서의 글자의 수가 하나 이상으로 확인되면 proprocessing() 함수를 거쳐 명사처리 과정을 거친다.
```

<그림 8> extractNoun() 함수 전처리 알고리즘

```
if(!exists("HannanumObj", envir=KoNLP:::.KoNLPEnv)){
    assign("HannanumObj",.jnew("HannanumInterface"), KoNLP:::.KoNLPEnv)
}
    out <- .jcall(get("HannanumObj",envir=KoNLP:::.KoNLPEnv),
                  "[S", "extractNoun",get("SejongDicsZip", envir =
                   KoNLP:::.KoNLPEnv), sentence_pre, get("CurrentUserDic",
                   envir=KoNLP:::.KoNLPEnv))
Encoding(out) <- "UTF-8"
return(out)
```

```
# .rJava는 R에서 Java를 불러오는 기능을 가진 패키지이며 자바의 프로그래밍적 능력을 R의 통계
  처리 능력과 결합할 수 있다.
# .jnew() - 실행클래스 객체를 만들어준다.
# .jcall() - 클래스명, 리턴타입, 메소드명 등을 기재하여 “SimplePos09" 메소드를 갖고 있다.
※. "S" : String, “[S" : String array
'HannanumInterface"라는 이름을 가진 Java 클래스를 생성한다. 그 안에 "out()"메소드를 만드는
  과정이다. "UTF-8"로 인코딩하여 리턴한다.
```

<그림 9> extractNoun() 함수 알고리즘

rJava 패키지는 객체지향프로그램인 JAVA 환경을 사용하며 객체를 만들고 메소드를 정의하는 과정을 거친다. 또한 사용자 정의 추출과정으로 먼저 현 프로그램 개발 환경과 같은 인코딩 연구 대상 문서인지 확인하고 이후 불필요한 공백 제거 및 비정상적 단어 추출 과정을 거친다. 대규모 언어 자료를 축적 가공하는 국립국어원 언어정보나눔터(21세기세종) 사전과 사용자 등록 사전을 함께 사용한다. jcall()함수에 extractNoun()과 SimplePos09() 함수를 메소드로 읽어 들여 명사 추출과 단어별 품사를 부여한다. 이는 <그림 9>에서 확인가능하다.

extractNoun()함수에서 처리한 명사리스트를 인코딩하여 SimplePos09() 함수를 사용하여 2차 명사 확인 절차를 거친다. 문자열 제어 함수인 str관련 함수를 사용하여 품사 태그 결과 리스트에서 명사 단어만을 추출하였다. 추출된 단어들의 뭉치를 벡터 테이블로 저장하여 리턴한다<그림 10>.

```
new_Noun <- function(sentence){
  sentence_pre <- preprocessing(sentence[1])
  if(sentence_pre == FALSE){
    return(sentence)
  }
  if(!exists("HannanumObj", envir=KoNLP:::.KoNLPEnv)){
      assign("HannanumObj",.jnew("HannanumInterface"), KoNLP:::.KoNLPEnv)
  }
  out <- .jcall(get("HannanumObj",envir=KoNLP:::.KoNLPEnv),
                "[S", "extractNoun",get("SejongDicsZip", envir=
                KoNLP:::.KoNLPEnv),sentence_pre, get("CurrentUserDic",
                envir=KoNLP:::.KoNLPEnv))
  out1<-out
  Encoding(out) <- "UTF-8"
  out2 <- NULL
  for (i in 1:length(out1)) {
    out0 <- .jcall(get("HannanumObj",envir=KoNLP:::.KoNLPEnv),
                   "S", "SimplePos09",get("SejongDicsZip", envir=
                   KoNLP:::.KoNLPEnv),out1[i], get("CurrentUserDic",
                   envir=KoNLP:::.KoNLPEnv))
    out2 <- c(out2, out0)
    Encoding(out2) <- "UTF-8"
  }
  ex_str<-NULL
  for (i in 1:length(out2)) {
    st=str_locate(out2[i],"\t")
    en=str_locate(out2[i],"/N")
    ex_str = rbind(ex_str, str_sub(out2[i], st[1]+1, en[1]-1))
  }
  return(ex_str)
```

```
# extractNoun() 함수와 SimplePos09() 함수를 결합한 new_Noun()함수를 제시한다.
# extractNoun()에서 명사처리 과정의 결과 데이터를 SimplePos09() 함수를 이용하여 Tagging 과정
  을 거쳐 명사 부분만을 추출하여 보다 정확도 높은 명사 처리 수행율을
 보이고 있다.
```

<그림 10> 연구 논문 new_Noun() 함수 알고리즘

Ⅳ. 연구 알고리즘 실험과 결과

본 연구에서는 KoNLP() 패키지를 활용하여 명사를 추출하는 extractNoun() 함수와 단어들의 형태소를 9개의 품사로 분리 가능한 SimplePos09() 함수의 주요 기능을 분석하고 새로운 통합 명사처리 new_Noun() 패키지를 개발하여 실험 연구를 하였다.

실험은 R-3.3.1 버전으로 진행되었으며 데이터 구성은 <표 1>과 같이 구성하였다.

본 연구의 실험에 사용된 R의 소스 코드는 <그림 11>과 같으며 먼저 연구 대상인 문서를 Corpus의 말뭉치로 결합하여 의미 전달과 상관없는 불용어 제거 과정이나 세종사전에 등록되지 않은 새로운 단어를 등록하는 과정을 통하여 보다 정확도 높은 명사 처리를 위한 전처리 과정을 진행한다. 하지만 이번 실험에서는 모든 전처리 과정을 생략한 상태로 기존에 사용하는 함수에만 의존하여 마이닝 처리를 하였다. 명사 추출 함수인 extractNoun()의 성능을 평가하기 위한 본 연구자의 의도 때문이다.

<표 1> 연구 대상 데이터 통계

	실험1	실험2	실험3
구분	뉴스	블로그	블로그
출처	헤럴드경제	네이버	네이버
자료형	텍스트	텍스트	텍스트
글자	1,783	1,531	1,933
단어	406	306	473

```
library(rJava)                                          # 필요한 패키지 로딩
library(KoNLP)
library(tm)

cname=file.path("./data")                               # 말뭉치 경로 설정
dir(cname)
docs = Corpus(DirSource(cname))

write(unlist(docs),"test.txt")                          # 데이터 확인 차원의 저장
place = readLines("test.txt")

place = sapply(place, extractNoun, USE.NAMES=F) # 명사 추출 처리
write(unlist(place),"test_mining.txt")                  # 결과 저장
```

<그림 11> 기존 extractNoun()함수를 사용한 마이닝 알고리즘

<그림 12>에서 알 수 있듯, 위 실험은 본 연구과정에서 개발된 new_Noun() 함수로 진행하였다. new_Noun()함수는 기존 extractNoun() 함수의 명사처리 이후 추출된 명사를 다시 SimplePos09() 함수를 사용하여 품사 태그 작업을 진행하였다. extractNoun()함수의 비정상적인 명사 추출 오류를 바로 잡기 위한 새로운 명사 추출 함수를 적용한 것이다.

본 연구과정에서 실험한 결과는 <표 2>, <표 3>, <표 4>와 같다. 먼저 <표 2>는 상대적으로 표준어를 사용하는 뉴스 데이터를 사용하여 기존 명사 추출 함수와 품사 태그 처리 과정을 포함하여 명사를 추출하는 새로 개발된 new_Noun() 함수와의 비교표이다(그림 12). 전체 397개 단어의 명사 추출에서 기존 명사 처리와 불일치한 단어가 62개로 15.6%이며 특히 명사가 아닌 단어를 명사 처리하여 배제된 단어가 29개인 7.3%에 이르렀다. 또한 불일치하거나

```
library(rJava)                                    # 필요한 패키지 로딩
library(KoNLP)
library(tm)

cname=file.path("./data")                         # 말뭉치 경로 설정
dir(cname)
docs = Corpus(DirSource(cname))

write(unlist(docs),"test.txt")                    # 데이터 확인 차원의 저장
place = readLines("test.txt")

place = sapply(place, new_Noun, USE.NAMES=F)      # 개발 함수로 명사 추출
write(unlist(place),"test_mining.txt")            # 결과 저장
```

<그림 12> new_Noun()함수를 사용한 마이닝 알고리즘

<표 2> 실험 1 결과

	extractNoun()	new_Noun()	비 고
명사	397	368	
명사 처리 일치	334		84.1%
명사 처리 불일치	62		15.6%
명사 제외 단어	29		7.3%
개선 단어	91		22.9%

헤럴드신문, 1,783글자, 406단어 중 368단어 명사화(90.6%)

제외된 단어 즉, 기존 명사 처리를 바로 잡은 단어의 수가 91개로 명사 처리가 22.9% 개선된 것으로 나타났다.

<표 3>은 인터넷 사용자들이 자유롭게 의사를 표현하는 블로그를 대상으로 진행하였다. 신문 뉴스 기사보다 상대적으로 표준어 사용이 낮은 자연어 처리를 분석한 결과이다. 전체 268개 단어의 명사 추출에서 이번 실험 결과는 명서 처리가 29.1% 개선된 것으로 나타났다. 앞 실험과는 대조적으로 7% 이상의 차이를 보이는 결과이며 배제된 단어와 불일치 단어 수의 비율이 높은 것을 알 수 있다. 인터넷 사용자들이 사용하는 이모티콘이나 비표준어 사용에 따른 결과인 것이다.

<표 4> 실험은 상대적으로 앞 <표 3>보다 개선된 단어의 빈도는 낮은 편이며 표준어를 사용하는 뉴스 데이터와 유사한 개선도를 확인할 수 있었다. 실험 단어의 수는 406개 단어였지만 실제 명사로 추출된 단어는 291개 단어로 상대적으로 명사 처리가 된 단어의 수가 낮게 나타났다. 실험 1에서는 실험 2, 실험 3과 대조적으로 대부분의 단어들이 명사 처리된 것으로 나타났다. 일반 사용자들이 블로그에 기재한 글들은 분석이 그만큼 어렵다는 것을 반증하며 뉴스 데이터와 인터넷어, 채팅어의 사용이 일반화된 블로그와 차이가 난다고 볼 수 있다.

본 연구의 실험에서 기존 명사 처리 함수와 본 연구의 연구자가 개발한 명사 추출 함수와 비교했을 때 불일치되는 단어들의 리스트는 <표 5>와 같다. 명사에서 조사 품사의 분리가

<표 3> 실험 2 결과

	extractNoun()	new_Noun()	비 고
명사	268	246	
명사 처리 일치	211		78.7%
명사 처리 불일치	56		20.9%
명사 제외 단어	22		8.2%
개선 단어	78		29.1%

네이버 블로그, 1,531글자, 306단어 중 246단어 명사화(80.4%)

<표 4> 실험 3 결과

	extractNoun()	new_Noun()	비 고
명사	291	269	
명사 처리 일치	242		83.2%
명사 처리 불일치	48		16.5%
명사 제외 단어	22		7.6%
개선 단어	70		24.1%

네이버 블로그, 1,933글자, 473단어, 269단어 기준(56.9%)

비정상적으로 된 사례들이 대부분을 차지하였다. 하지만 “한글에”, “한글은”, “한글을”, “한글이”와 같은 단어들은 대부분 조사를 분리하지 못하여 명사 “한글”의 빈도수에도 영향을 준 것으로 보인다. “초밥”의 명사 또한 상당수 조사 조합으로 인하여 정상적으로 명사 처리가 안 된 것으로 나타났다.

기존 명사 처리 함수에서 명사로 판단된 몇몇 함수들이 new_Noun()함수를 거치고 난 후 명사가 아닌 것으로 구분되었으며, 명사 리스트에서 제외된 단어들은 <표 6>과 같이 정리할 수 있다.

<표 5> extractNoun()과 new_Noun()함수와의 불일치 명사 리스트

No	extractNoun()	new_Noun()	extractNoun()	new_Noun()	extractNoun()	new_Noun()
1	10선을	10선	맛집이였어여	맛집	책테마파크에	책테마파크
2	13작품이	13작품	먹다보니	먹다보	책테마파크에서	책테마파크
3	2구역은	2구역	먹방은	먹방	초밥들이	초밥들
4	2인석에	2인석	미니우동하나를	미니우동하나	초밥먹을꺼면	초밥먹을꺼
5	2인석은	2인석	부른배에도	부른배	초밥은	초밥
6	3구역은	3구역	성남문화재단은	성남문화재단	초밥이	초밥
7	4구역으로	4구역	시켜먹었어여	시켜먹었어	초밥이랑은	초밥
8	4구역은	4구역	식감이	식감	초밥입니다	초밥
9	갈꺼에요~	갈꺼	식감자체가	식감자체	초밥집이	초밥집
10	고깃집에	고깃집	쌈밥같은	쌈밥같	초밥집입니다	초밥집
11	고깃집임에도	고깃집	쌈채소나	쌈채소	촉촉~해서	촉촉
12	굽느냐에따라	굽느냐	안되서	안되	키즈룸입니다	키즈룸
13	내안에를	내안에	안받고	안받	하시더라구	하시
14	내안에에서는	내안에	없어여	없어	한국콘텐츠진흥원과	한국콘텐츠진흥원
15	네이버가	네이버	예삿놈이	예삿놈	한글날에는	한글날
16	네이버는	네이버	용비어천가를	용비어천가	한글날을	한글날
17	대답하시더라구	대답하시	이히히히~~	이히히히	한글에	한글
18	떡~~~하니~	떡~~~하	잔뜩~	잔뜩	한글은	한글
19	뚫려있는곳은	뚫려있는곳	잘어울린다는	잘어울린다	한글을	한글
20	뜨거운물이	뜨거운물	종류만해도	종류만해	한글이	한글
21	리필을	리필	주문햇습니다	주문햇습니	한글자료도	한글자료
22	말했어여	말했어	쫙~	쫙	한글자료를	한글자료
23	맛있더라구여	맛있더라구	찜갈비도	찜갈비	햇기에	햇기
24	맛집에서는	맛집	찜갈비를	찜갈비	허형만	허형
25	맛집이였습니다	맛집	책테마파크가	책테마파크	활어로	활어

<표 6> 기존 명사에서 new_Noun()함수 사용으로 제외된 명사 리스트

No	extractNoun()	new_Noun()	extractNoun()	new_Noun()	extractNoun()	new_Noun()
1	각	"각/M"	안	"안/M"	처음	"처음/M"
2	계속	"계속/M"	애	"애/I"	체	"체/I"
3	들이	"들이/M"	약	"약/M"	하더라구	"하/P+더라구/E"
4	떡	"떡/M"	어리둥절	"어리둥절/M"	한	"하/P+ㄴ/E"
5	맛있어	"맛있/P+어/E"	요놈	"요놈/I"	합니다	"하/P+ㅂ니다/E"
6	몇	"몇/M"	위해	"위하/P+어/E"	해	"하/P+어/E"
7	물론	"물론/M"	이	"이/M"	해서	"하/P+어서/E"
8	사왔습니다	"사/P+아/E+오/P+아ㅂ니다/E"	이미	"이미/M"	회	"회/M"
9	석	"석/M"	저	"저/M"	후	"후/I"
10	순	"순/M"	전	"전/M"		
11	시	"시/I"	쪽	"쪽/M"		

M : 수식언(관형사, 부사), I : 독립언(감탄사), P : 용언(동사, 형용사), E : 어미(연결어미)

Ⅴ. 결론 및 향후 과제

본 논문은 R을 통한 텍스트마이닝 과정 중 명사 추출 함수에 대한 연구와 실험을 진행하였다. 빅데이터 분석은 방대한 데이터 내에서 패턴을 찾아 예측을 하는 것이다. 자료의 크기는 모집단인 자료 전체를 분석하는 전수조사 방식이 가능하기 때문에 무엇보다 의미가 있는 분석 방법이다. 많은 연구자들이 고객의 소리에 귀 기울이며 소비자의 패턴과 니즈 분석, 의미 기반 분석 등이 가능하다.

기존 명사 추출의 결과에서 비정상적인 명사 추출을 살펴보면 “은”, “는”, “이”, “가”와 같은 조사들의 비정상적인 분리가 이상 현상으로 나타났다. “2구역은”, “네이버가”, “네이버는”, “먹방은”, “식감이”, “초밥들이”, “초밥은”, “초밥이”, “초밥집이”, “한글은”, “한글이” 등 일부 단어의 조사 분리가 미완성된 상태에서 명사 분리 단어로 구분되어 마이닝 처리되어왔던 것으로 보이며, 이번 연구에 개발된 new_Noun() 함수를 통하여 정상적으로 명사 추출이 가능한 것으로 나타났다.

또한, 기존 명사 추출 함수에서 명사 추출 단어로 구분되어 오던 단어들을 본 연구 개발된 함수로 명사가 아님을 구분할 수 있게 되었다. “어리둥절”, “계속”, “들이”, “물론”, “석”, “순”, “안”, “이미”, “쪽”, “처음” 등은 명사로 구분되어져 왔지만 부사로 표현되어야 되므로 명사 추출에서 제외되어야 한다. 또한 “몇”, “약”, “이”, “저”는 관형사로 구분되어야 한다.

<실험 1>은 방송이나 신문 뉴스의 이야기들은 표준어로 구사된 검증된 언어들만 사용하므로 문서 내 명사 추출 확률 또한 90.6%로 나타

나 매우 높은 수준이며 명사 추출 과정 또한 그만큼 정확도가 높다고 볼 수 있다(표 2).

<실험 2>, <실험 3>은 자신의 관심에 따라 자유롭게 글과 사진, 동영상을 공유하는 블로그의 텍스트를 기반으로 실험하였다. 인터넷어, 채팅어, 신조어, 자음 모음을 활용한 이모티콘 등으로 자유롭게 표현하는 문장들이 대부분이라 상대적으로 명사 추출 확률이 56.9% 수준에 그치는 정도였다. 자세한 사항은 <표 4>를 통해 확인할 수 있다.

기존 명사 추출 과정에서 미처 발견하지 못한 명사 단어는 뉴스 기사처럼 표준어를 구사하는 텍스트 실험에서 23%에 가까이 추가 추출이 되어 개선이 이루어졌으며 일반 사용자들이 작성한 자유 형식의 블로그 게시글을 분석하는 실험에서는 30%에 가까운 명사 추출 개선이 이루어졌다. 이러한 결과를 통해 향후 자연어 처리 연구에 new_Noun() 함수가 많은 도움이 될 것으로 확인 되었고, R 공식 사이트인 CRAN을 통해 배포하여 자연어 처리 연구자들에 보탬이 되고자 한다.

본 연구는 자연어 처리의 비표준어 처리 부분에서 한계점을 보이고 있다. 이는 블로그 게시글의 명사 변환에서 명확히 드러났다. 인터넷에서 쓰는 채팅어, 국어와 외국어를 조합한 합성어, 오타에서 파생된 파생어, 단어들의 함축적으로 표현하는 등 어근을 찾기 힘든 외계 단어들, 즉 인터넷에서 사용되는 비속어의 표준어 구별이 어렵기 때문이다. 따라서 다양한 매체를 통해 대중적으로 사용되는 비표준어들에 대한 사전 구축 과정을 거쳐 표준어로 변환하는 과정이 필요하다. SNS상에서 무분별하게 사용되는 단어들이 사전에 등재되어 명확히 인식될 때 자연어 처리 연구 결과가 보다 더 신뢰될 수 있을 것이다.

참고문헌

권순창, “A Study on the Use of Open Source Software in Vocational Education,” 한국전산회계학회 정기학술발표회, 2007, pp. 165-169.

김상현, 송영미, “오픈소스 소프트웨어의 지속적인 사용의도에 영향을 미치는 요인에 관한 연구,” 인터넷전자상거래연구, 제9권, 제1호, 2009, pp. 257-280.

김성용, 이상민. “무선 인터넷 망에서 임베디드 리눅스 기반 PDA 를 이용한 영상보드 원격제어 시스템 구현,” 정보시스템연구 제17권, 제1호 2008, pp. 155-171.

김용현, 허의남, “Log Analysis Supporting System based on Log Data for Efficient Big Data Analysis,” 한국정보과학회 학술발표논문집, 2014, pp. 936.

문상식, 김기홍, “IT 환경 변화에 따른 한국의 오픈소스 소프트웨어의 정책방향 연구,” 인터넷전자상거래연구, 제14권, 제1호, 2014, pp. 203-221.

사공원, 하성호, 박경배. “온라인 후기에 내재된 고객의 감성분석과 LQI 차원별 호텔서비스 품질 평가,” 정보시스템연구 제25권, 제3호, 2016, pp. 217-245.

손수아, 박석천, “IoT 기반 실시간 시각화 알고리즘을 이용한 스마트가드닝 시스템 설계 및 구현,” 정보교육학회논문지, 제16권, 제6호, 2015, pp. 31-37.

박정웅, 최영민, 박희동, “오픈 소스 하드웨어

기반의 스마트 온실관리 시스템 설계 및 구현," 디지털융복합연구, 제14권, 제2호, 2016, pp. 259-264.

신수범, "Teaching and Learning Strategies of Computer Algorithms using Robot," 한국엔터테인먼트산업학회 학술대회 논문집, 2015, pp. 39-42.

안정국, 김희웅, "Building a Korean Sentiment Lexicon Using Collective Intelligence," 지능정보연구, 제21권, 제2호, 2015, pp. 49-64.

장영재, "튜토리얼: 빅데이터, 비즈니스 애널리틱스, IoT: 경영의 새로운 도전과 기회," 정보시스템연구, 제24권, 제4호, 2015, pp. 139-152.

한만휘, 박성찬, 이한빛, 연종흠, 이상구, "한국어 언어자원에서의 자연어 처리 기술 현황 조사," 한국정보과학회 학술발표 논문집, 2015, pp. 681-683.

Black, E. W., "Wikipedia and academic peer review: Wikipedia as a recognised medium for scholarly publication?," *Online Information Revie*, Vol. 32, No. 1, 2008, pp. 73~88.

Cachia, R., R. Compano, and O. D. Costa, "Grasping the potential of online social networks for foresight," *Technological Forecasting and Social Change*, Vol. 74, No. 8, 2007, pp. 1179~1203.

Lee J, Le HS, Lee H. "Research on Methods for Processing Nonstandard Korean Words on Social Network Services," *Journal of the Korea Industrial Information Systems Research*, Vol. 21, No. 3, 2016, pp. 35-46.

Lee, J. H. and Lee, H. K., "A Study on Unstructured Text Mining Algorithm through R Programming based on Data Dictionary," *Journal of the Korea Society Industrial Information System*, Vol. 20, No. 2, 2015, pp. 113-124.

Lee, J. H, Le. H. S. and Lee, H. K., "A Study on Customer Reviews about Domestic and Imported Clothes Products through Opinion Mining," *The Journal of Internet Electronic Commerce Research*, Vol. 15, No. 3, 2015, pp. 233-234.

LE, H., LEE, J. H. and LEE, H. K., "Purchase Process Aspect-based Opinion Mining : An Application for Online Shopping Mall," *The Journal of Internet Electronic Commerce Research*, Vol. 15, No. 2, 2015, pp. 15-28.

http://www.nipa.kr

https://www.r-project.org/

이 종 화 (Jong-Hwa Lee)

부경대학교 경영학 박사과정을 수료하였으며, 현재 부경대학교 경영학부 시간강사로 재직 중이다. 주요 관심분야는 BigData, Mining, Content Analysis 등이다.

이 현 규 (Hyun-Kyu Lee)

연세대학교에서 경영학 박사학위를 취득하고, 현재 부경대학교 경영학부 교수로 재직하고 있으며, 주요 관심분야는 정보시스템전략, Data- Mining & Analysis 등이다.

<Abstract>

Research on Natural Language Processing Package using Open Source Software

Jong-Hwa Lee · Hyun-Kyu Lee

Purpose

In this study, we propose the special purposed R package named ""new_Noun()" to process nonstandard texts appeared in various social networks. As the Big data is getting interested, R - analysis tool and open source software is also getting more attention in many fields.

Design/methodology/approach

With more than 9,000 R packages, R provides a user-friendly functions of a variety of data mining, social network analysis and simulation functions such as statistical analysis, classification, prediction, clustering and association analysis. Especially, "KoNLP" - natural language processing package for Korean language - has reduced the time and effort of many researchers. However, as the social data increases, the informal expressions of Hangeul (Korean character) such as emoticons, informal terms and symbols make the difficulties increase in natural language processing.

Findings

In this study, to solve the these difficulties, special algorithms that upgrade existing open source natural language processing package have been researched. By utilizing the “KoNLP” package and analyzing the main functions in noun extracting command, we developed a new integrated noun processing package "new_Noun()" function to extract nouns which improves more than 29.1% compared with existing package.

Keywords : Text Mining, NLP, R Program, Package, Open Source Software

* 이 논문은 2016년 11월 17일 접수, 2016년 12월 1일 1차 심사, 2016년 12월 27일 게재 확정되었습니다.