제목: 한글 텍스트 분석을 위한 명사 추출 방법 제안

이상엽

연세대학교 언론홍보영상학부 조교수

요약

인터넷 이용 인구의 증가로 인해 온라인 텍스트 데이터 분석을 통한 사회 현상 연구의 중요성이 커지고 있다. 컴퓨터 기반 방법을 이용해 대용량의 텍스트 데이터를 분석하고자 하는 경우, 형태소 분석기를 통해 주요 단어를 추출하는 작업이 선행되어야 한다. 하지만, 형태소 분석기가 사용하는 사전에 등록되어 있지 않은 단어들은 제대로 추출되지 않는 미등록 단어 문제가 발생하게 된다. 이러한 미등록 단어는 대부분 명사의 단어들이다. 이러한 명사 미등록 단어의 문제를 개선하기 위해, 본 연구에서는 명사의 사용·배제 정보와 Komoran 이라는 기존 형태소 분석기를 기반으로한 명사 추출 알고리즘을 제안한다. 그리고 독자들이 쉽게 사용할 수 있도록 관련 알고리즘을 Python 을 통해 공개하고 설명 방법을 제공한다. 본 알고리즘은 고유 명사, 신조어, 외래어 등의 미등록 명사 단어를 추출하는데 있어 기존 형태소 분석기 보다 성능이 좋은 것으로 나타났다.

1. 서론

인터넷이 우리 사회의 중요한 부분을 차지하게 됨으로써 온라인 데이터 분석은 인문사회 분야의 연구자들에게도 그 중요성이 커지고 있다. 인터넷에 존재하는 다양한 형태의 데이터 중에서, 가장 많은 부분을 차지하는 것이 텍스트 데이터이다. 신문기사, 신문기사의 댓글, 블로그, 상품평, 소셜미디어에 사용자들이 올리는 글 등이 대표적인 온라인 텍스트 데이터의 예이다.[1] 이는 인터넷과 연관된 사회 현상 연구를 위해서는 온라인 텍스트 데이터 분석이 중요하다는 것을 의미한다.

온라인 텍스트 데이터는 보통 그 양이 방대하기 때문에, 내용 분석과 같은 전통적인 방법은 적용되기 어렵다. 따라서, 대용량 텍스트 데이터의 효과적인 분석을 위해서 컴퓨터 기반의 방법을 사용하게 된다. 이러한 대용량 온라인 텍스트 데이터의 분석은 내용 분석과 같은 전통적인 분석

[1] 본 연구에서는 한글 텍스트만 다룬다.

방법이 적용되기 어려운 연구 문제를 해결하는 실마리를 제공할 뿐 아니라, 인터넷의 발달과 새롭게 발생하고 있는 사회 현상들을 이해할 수 있는 기회를 제공한다.

비정형(unstructured)의 텍스트 데이터를 컴퓨터를 이용해서 분석하기 위해서는 정형 데이터와 비슷한 형태로 변경하는 작업이 필요하다. 이를 위해 텍스트 데이터를 구성하고 있는 각 문서(document, 예: 하나의 신문기사)를 단어의 집합으로 표현하고, 각 단어를 숫자 정보로 변환하게 된다. 즉, 텍스트 데이터를 컴퓨터 기반 방법으로 분석하기 위해서는 우선적으로 텍스트를 단어 단위로 구분하는 작업이 필요한 것이다.

한글의 경우, 텍스트 데이터를 단어 단위로 구분하기 위해 형태소 분석 방법을 사용한다. 형태소 분석이란, 텍스트 데이터를 형태소 단위로 구분하고 각 형태소의 품사를 태깅하는 과정을 의미한다 (강승식, 2002).[2] 형태소 분석은 형태소 분석기라는 툴을 이용해서 이뤄진다.

형태소 분석기는 형태소와 형태소의 품사 정보를 저장하고 있는 형태소 사전과 형태소 형성 규칙 정보를 기반으로 텍스트에서 형태소를 구분하고, 각 형태소의 품사를 찾는다. 하지만, 형태소 분석기는 사전을 기반으로 작동하기 때문에, 사전에 등록되지 않은 형태소는 잘 찾지 못하는 문제가 발생하는데, 이러한 문제를 미등록 단어 문제라고 한다 (심준혁 외, 1999). 이러한 미등록 단어 문제를 해결하지 않은 형태소 분석 결과를 그대로 최종 분석 방법에 적용하게 되면, 그 분석 결과의 정확성이 떨어진다. 따라서, 텍스트 데이터를 컴퓨터로 분석하여 보다 정확한 결과를 얻기 위해서는 미등록 단어 문제를 해결하는 것이 필요하다. 미등록 단어의 대부분은 고유명사, 신조어, 전문 용어 등과 같은 명사이기 때문에 (이도길 외, 2003), 미등록 단어 문제를 해결하기 위해서는 명사를 정확하게 추출하는 것이 중요하다.

다수의 연구자들이 인터넷에 존재하는 한글 텍스트 데이터를 컴퓨터 기반 방법으로 분석하여 사회 현상 연구하고자 하는 시도를 했다 (예,감미아·송민, 2012; 심주영, 2017; 양희수·현은정, 2018; 윤지윤·한민규, 2016; 좌미라, 2018). 이러한 연구의 다수가 명사 단어를 기반으로 분석을 수행하였고, 명사 추출을 위해서 기존에 개발되어 있는 형태소 분석기를 사용하였다. 하지만, 기존 형태소 분석기를 사용함으로써 발생하는 미등록 단어 문제를 어떻게 해결했는지에 대해서 구체적으로 기술한 논문은 거의 존재하지 않는다. 더욱이 문제가 되는 것은 다수의 선행 연구들이

[2] 형태소 분석과 품사 태깅을 구분하는 사람들도 있지만, 본 연구에서는 형태소 분석을 품사 태깅까지 포함하는 의미로 사용합니다.

미등록 단어 문제의 큰 영향을 받을 수 있는 분석 방법들 (키워드 빈도 분석, 의미 연결망 분석 등)을 주 기법으로 사용했다는 것이다. 이는 저자들이 미등록 단어 문제의 심각성을 알지 못했거나, 알았더라도 미등록 단어 문제를 어떻게 해결해야하는지 몰라서 발생했을 가능성이 크다.

이러한 미등록 단어 문제를 해결하기 위해서는 등록되지 않은 명사 단어를 형태소 분석기가 사용하는 사전에 추가를 하면 된다. 하지만, 대용량의 텍스트 데이터에 존재하는 미등록 명사 단어가 무엇인지 모를 뿐 아니라, 그 수가 많아서 사람이 직접 사전에 등록하는 것은 거의 불가능하다 (이도길 외, 2003).

이에 따라, 미등록 단어 문제 해결을 위해 컴퓨터 기반의 다양한 명사 추출 방법들이 제시되었다 (예, 박용현 외, 2010; 안동언, 1999; 이도길 외, 2003; 이중영 외, 1999). 하지만, 이러한 방법은 실제로 사용하기가 쉽지 않다. 왜냐하면, 제시된 방법들이 사용하는 사전과 알고리즘 등을 공개하지 않기 때문이다. 더군다나 코딩에 익숙하지 않은 인문사회분야의 연구자에게는 그러한 활용이 더욱 어렵다.

따라서 본 연구는 인문사회분야 연구자들이 쉽게 사용할 수 있는 컴퓨터 기반의 명사 추출 방법을 제시하고 관련 코드를 공개함으로써 코딩 경험이 많지 않은 연구자나 학생들이 쉽게 사용할 수 있게 하고자 한다.

2. 한글 텍스트 분석 소개

본 연구에서 제안하는 명사 추출 방법을 설명하기 이전에 한글 텍스트 분석에 대한 간략한 소개를 하도록 한다.

2.1 한글의 형태론적 특성

한글 텍스트의 분석을 이해하기 위해서는 한글의 형태론적 특성을 아는 것이 필요하다. 유현경 외 (2015)에 따르면, 형태론은 “단어가 어떤 구조로 되어 있는지 단어의 하위 부류들은 어떤 것들이 있는지 등 단어와 관련한 언어학적 사실들에 대해 연구하는 언어학의 하위 분야”로 정의된다. 간단하게 얘기하면, 단어의 형태와 구조를 연구하는 학문이라고 생각할 수 있다.

한글에서의 하나의 문장은 어절로 구성되어 있고, 어절은 단어들의 조합으로 구성이 되며, 단어들은 형태소의 조합으로 구성이 된다. 예를 들어, 다음과 같은 문장이 있다라고 가정을 하자.

"철수가 밥을 먹다"

위의 문장을 어절로 구분하면, '철수가', '밥을', '먹다'로 구분이 되며, '철수가'라는 어절은 '철수' + '가' 라는 두개의 단어로 구성이 되어 있고, '밥을'은 '밥'+ '을', '먹다'는 '먹다'라는 단어로 구성되어 있다. '철수'나 '가'라는 단어는 하나의 형태소로 구성이 된 단어이다. 하지만, '먹다'라는 단어는 '먹'이라는 형태소 (어간)과 '다' 라는 형태소 (어미)로 구성되어 있다.

단어는 "최소의 자립 형식"으로 정의 되고, 이는 의미적으로나 문법적으로 자립하여 사용될 수 있는 최소 단위를 의미한다(유현경 외, 2015). 단어는 문법적 성질에 따라 몇 가지로 구분되는데, 이를 품사라고 한다 (고창운, 2006). 단어의 품사에는 명사, 대명사, 수사, 형용사, 동사, 관형사, 부사, 조사, 감탄사 등의 9 가지가 존재한다. 품사는 단어가 문장에서 하는 역할에 따라 체언, 용언, 수식언, 관계언, 독립언으로 구분 된다 (유현경 외, 2015). 체언은 문장의 몸과 같은 주어와 목적어의 역할을 하며, 명사, 대명사, 수사 등의 품사가 포함된다. 용언은 문장의 서술어 역할을 하며, 형용사와 동사의 단어가 포함된다. 수식언은 체언이나 용언을 수식하는 역할을 하며, 부사와 관형사가 포함된다. 관계언에는 말들 간의 관계를 나타내는 조사가 있다. 감탄사는 독립언이다.

본 글에서는 명사에 대해 다루기 때문에, 명사에 대해서만 추가적인 설명을 하겠다. 다른 품사에 대해 궁금한 독자들은 관련 자료 (예, 고창운, 2006; 유현경 외, 2015 등)를 참조하기 바란다. 명사는 사물의 이름을 나타내는 단어들을 포함하는 품사이다. 명사 단어는 관형어의 수식을 받을 수 있으며, 조사와 결합하여 사용되기도 한다. 한국어의 단어 중 65% 정도가 명사에 속한다 (유현경 외, 2015). 명사는 크게 사용 범위에 따라 보통 명사 (common noun)와 고유 명사 (proper noun)로 나뉘며, 자립 가능 여부에 따라 자립 명사와 의존 명사로 구분된다. 의존 명사는 문장에서 반드시 관형어와 동반해서만 쓰일 수 있다. 예를 들어, 단위를 나타내는 명사 (개, 마리, 채 등)가 의존 명사에 속한다.

명사는 텍스트 분석에서 중요하게 간주되는데, 왜냐하면, 명사는 주요한 의미를 담고 있어 문서의 특성을 잘 나타내기 때문이다 (이도길 외, 2003). 따라서, 정확한 명사의 추출은 텍스트 분석을 위해 중요하다. 하지만, 형태소 분석에서 미등록 단어로 인식되는 대부분의 단어는 고유명사, 신조어, 전문 용어 등과 같은 명사다 (이도길 외, 2003). 따라서, 명사가 중요한 역할을 하는 텍스트 분석에서는 이러한 명사의 미등록 단어 문제를 해결해야 한다. 그렇지 않으면, 분석 결과의 정확도가 떨어지게 된다.

하나의 단어는 한 개 이상의 형태소로 구성된다.[3] 형태소는 "의미를 갖는 최소 단위"로 정의된다. 여기서의 '의미'는 어휘적인 의미뿐 아니라 형식적인 혹은 문법적인 의미도 포함이 된다 (유현경 외, 2015). 예를 들어, 어미는 단어는 아니지만, 어절을 생성하는데 있어서 문법적인 의미를 갖기 때문에 형태소로 구분이 된다.

형태소는 자립성의 여부와 의미의 허실에 따라 그 유형이 나뉜다 (유현경 외, 2015). 자립 형태소는 다른 형태소와 결합하지 않고도 사용될 수 있는 형태소로, 위의 예에서 '철수' 와 '밥' 등이 포함된다. 반대로, 의존 형태소는 다른 형태소와 결합되어 사용되는 형태소를 의미하고, 위의 예에서 '가', '먹' 등이 포함된다. 실질적인 의미를 갖느냐에 따라도 구분이 되는데, 실질 형태소는 실질적인 의미를 가진 형태소다 (예, '철수', '밥', '먹'). 실질 형태소에는 자립 형태소뿐 아니라 의존 형태소(예, '먹')도 포함될 수 있다. 형식 형태소 (문법 형태소라고도 함)는말과 말 사이의 형식적인 (문법적인) 관계를 표시하는데 사용되는 형태소다 (예, '는', '와', '다').

하나의 단어를 형성하는 형태소는 그 관점에 따라서 불리는 이름이 다르다. 단어 형성론 관점에서는 단어를 구상하는 형태소를 어근과 접사로 구분하는 반면, 활용론 관점에서는 어간과 어미로 구분한다 (유현경 외, 2015).

2.2 미등록 단어 문제

앞에서 언급된 것 처럼 미등록 단어 문제는 형태소 분석기를 이용하여 단어를 추출하는 경우, 단어들이 분석기가 사용하는 사전에 등록되어 있지 않아 분석이 잘 되지 않는 것을 의미한다. 보통 고유명사나 신조어 등이 미등록 단어에 속하게 된다. 예를 들어, '최순실'이라는 고유명사를 Komoran 이라는 형태소 분석기를 사용해서 분석하면, '최순'+'실'로 분리가 된다. 이는 최순실이라는 명사 단어가 Komoran 형태소 분석기가 사용하는 형태소 사전에 등록이 되어 있지 않기 때문이다. 또 다른 예로, 파이썬 컨퍼런스를 의미하는 '파이콘' 이라는 단어도 등록되어 있지 않아, 형태소 분석을 하게 되면 '파' + '이콘'으로 분리된다. '브렉시트'와 같은 외래어도 제대로 처리되지 않는다.

2.3 한글 텍스트 분석의 주요 순서와 원리

[3] 하나의 명사 단어는 하나의 형태소로 간주된다. 따라서 형태소 분석을 통해서 명사를 구분하는 것이 가능하다.

텍스트 분석을 하기 위해서는 먼저 분석에 사용되는 텍스트가 있어야 한다. 이러한 텍스트는 보통 여러 개의 문서로 구성되어 있다.[4] 특정 주제에 대한 여러 개의 신문기사들을 분석하고자하는 경우에는 하나의 신문기사가 하나의 문서가 된다.

분석하고자 하는 텍스트 데이터를 준비한 다음에 먼저 전처리 (preprocessing) 과정을 거친다. 전처리 과정에서는 불필요한 기호 제거, 불용어 제거, 최종 분석에 사용하고자 하는 품사의 단어 추출 등의 작업이 이뤄진다. 특정한 품사들의 단어를 추출하기 위해서는 텍스트를 구성하고 있는 문장들을 형태소 단위로 분리하고 각 형태소의 품사를 찾는 과정이 필요한데, 이를 형태소 분석이라고 한다.

전처리 과정을 거쳐 원하는 품사의 단어들만을 선택하고 난 후에는 분석의 목적에 따라 원하는 분석 방법을 적용한다. 보통 사용되는 분석 방법으로는 키워드 빈도 분석 (Keyword frequency analysis), 의미 연결망 분석 (Semantic network analysis), 군집 분석 (Clustering analysis), 토픽 모델링 (Topic modeling), 감성 분석 (Sentiment analysis) 등이 있다.

많은 분석 방법에서 가장 중요하게 간주되는 품사가 명사다. 왜냐하면, 명사는 실질적인 의미를 담고 있어 문서의 특성을 잘 나타내기 때문이다. 따라서 명사를 제대로 추출하는 것이 중요하다.

한글의 형태소 분석을 위해서는 보통 특정 컴퓨터 프로그래밍 언어 (예, Java, Python, R 등)에서 제공되는 형태소 분석을 위한 특정한 모듈을 사용하게 된다 (예, Python 은 KoNLPy (KoNLPy: Korean NLP in Python, 2018), R 은 KoNLP (Jeon, 2016)). 형태소 분석은 그러한 모듈에서 제공되는 형태소 분석기를 사용해서 수행하게 된다.

3. 명사 추출 방법 제안

본 연구에서는 문서에서의 명사 사용 특성과 Komoran 형태소 분석기 (SHINEWARE, 2018)를 기반으로 명사 단어들을 추출한다. 여러가지 형태소 분석기 중에서 Komoran 형태소 분석기를 사용한 이유는 현재 가장 지속적으로 업데이트가 이뤄지고 있는 형태소 분석기로 미등록 단어의 문제가 다른 형태소 분석기 보다 적기 때문이다.

3.1 명사의 사용 특성

[4] 이러한 문서의 집합을 말뭉치(corpus)라고 표현한다.

1) 문서에서의 명사 출현 특성

본 연구에서 고려한 문서에서의 명사 출현 특성은 아래와 같다. 명사 출현 특성은 이도길 외 (2003), 유현경 외 (2015), 강승식 (2002)를 참조하였다.

① 명사는 일반적으로 조사와 함께 사용된다 (유현경 외, 2015; 이도길 외, 2003).

명사는 많은 경우에 '은/는'의 주격 조사 또는 '을/를'의 목적격 조사와 함께 사용된다. 동일한 명사라고 할지라도 하나의 문서 혹은 텍스트 데이터에서 여러 개의 다른 조사들과 사용된다.

예) 철수는, 철수가, 철수를 등

위의 예에서 보이는 것 처럼 명사 부분('철수')은 변하지 않고 그와 사용되는 조사들은 변경될 수 있다. 즉, 조사 앞에 사용되며, 그 출현 빈도가 높은 단어일수록 명사일 확률이 높은 것이다. 조사를 구분하기 위해, 본 연구에서는 연세 한국어 사전 (언어정보연구원, 2018)을 사용하였다.

② 명사는 단독으로 사용되기도 한다.

명사는 보통 조사와 함께 사용되지만, 조사 없이 단독으로 사용되기도 한다.

예) '문재인 대통령은 북한을 방문하였다.'

위의 예에서 '문재인' 이라는 (고유) 명사는 조사의 도움없이 홀로 사용되었다.

③ 문서의 특성을 나타내는 명사는 반복적으로 사용된다.

보통 하나의 문서 (예, 신문기사)의 주제 혹은 특성을 나타내는 명사는 해당 문서에서 자주 등장한다. 이러한 특성은 문서에 사용된 명사들 중에서 상대적으로 자주 사용되는 명사만을 추출하고자 하는 경우에 유용하게 사용할 수 있다.

2) 명사가 포함되지 않는 어절의 특성

어절에서 명사를 추출하는데 있어, 우선적으로 명사가 포함되지 않는 어절을 배제하게 되면 명사 추출 작업을 보다 효율적으로 수행할 수 있다 (이도길 외, 2003). 특히, 어절을 구성하는 단어 중에

형용사, 부사[5], 동사 등이 포함되어 있는 어절은 명사 단어를 포함하지 않을 확률이 높다. 그리고, 이러한 어절들은 Komoran 형태소 분석기를 통해서 품사 태깅이 비교적 정확하게 된다는 특징이 있다. 아래는 그러한 어절들의 예를 나타낸다.

'어쩌면', '빨리', '천천히', '느리게', '그리고', '그런데', '그러므로', '가는', '돌아오는'

위의 어절들에 대해서 Komoran 을 통해 품사 태깅을 해보면 결과는 아래와 같다.

('어쩌면', 'MAG'), ('빨리', 'MAG'), ('천천히', 'MAG'), ('느리', 'VA'), ('게', 'EC'), ('그리고', 'MAJ'), ('그런데', 'MAJ'), ('그러므로', 'MAJ'), ('가', 'VV'), ('는', 'ETM'), ('돌아오', 'VV'), ('는', 'ETM')

MAG 는 일반 부사, VA 는 형용사, EC 는 연결 어미, MAJ 는 접속 부사, VV 는 동사를 의미한다. 즉, 명사로 구분되는 어절은 없는 것이다.

3) 명사 배제 정보 사용하기

추가적으로 명사가 아닌 단어들을 배제하기 위해서 본 논문에서는 다음과 같은 명사 배제 정보를 사용하였다.

① 이도길 외 (2003)에 따르면, 단어의 받침에 다음과 같은 낱말들이 사용되는 경우, 해당 단어는 명사일 가능성이 매우 낮다고 한다.

'ㄱㅅ','ㅆㅆ', 'ㄹㅎ', 'ㄹㅌ', 'ㄴㅎ','ㄴㅈ', 'ㄹㄱ', 'ㄹㅐ','ㄹㅛ'

② 한 음절의 단어 배제하기

한 음절의 단어는 명사가 아니거나, 의미없는 명사인 경우가 많다. 중요한 명사라고 할지라도 형태소 분석기를 통해서 명사로 간주된 한 음절의 단어는 그 단어 자체만 보고 무엇을 의미하는 단어인지 정확하게 파악할 수 없는 경우가 많다. 따라서 본 연구에서는 한 음절의 단어 혹은 어절은 배제하였다.

3.2 주요 명사 추출 과정

[5] 한국어에서는 접속사도 부사로 분류된다 (유현경 외, 2015). 접속 부사라고 일컫는다.

앞에서 살펴본 명사 출현 특성과 배제 정보, 그리고 Komoran 형태소 분석기를 기반으로 하여, 본 논문에서 제안된 알고리즘은 다음의 과정을 거쳐 명사를 추출한다.

① 불필요한 기호 없애기 등의 기본 전처리 작업
② Komoran 을 이용하여 명사 단어를 포함하지 않는 어절 제외하기
③ 위에서 언급된 명사 출현 특성을 사용해서 명사 단어 후보군 생성하기
④ 위에서 언급된 명사 배제 정보를 이용해 후보 단어들 중에서 명사가 될 수 없는 단어 제거하기
⑤ 후보 단어들 중에서 대명사 제외하기
⑥ 최종 후보 단어들에 대해서 Komoran 을 적용하여 명사가 될수 없는 단어들 제거하기
⑦ 최종 명사 단어 추출

예를 들기 위해, 아래의 텍스트 데이터에서 명사를 추출해 보도록 하겠다.

예) '자카르타 아시안게임에서 손흥민은 빠른 드리블과 슈팅으로 한국의 우승을 도왔다. '

위의 텍스트 데이터에 대해서 본 연구에서 제안한 알고리즘의 결과를 순서대로 기술해 보면 아래와 같다.

① 먼저 마침표(.)를 제거한다.
② 두번째로, '빠른', '도왔다'와 같은 어절이 Komoran 을 이용해서 제거된다.
③ 명사 출현 특성을 이용하여, '자카르타', '아시안게임', '손흥민', '드리블', '슈팅', '한국', '우승' 이 명사로 추출된다.
④ ~ ⑥ 과정에서 제거되는 단어들이 없으므로, 최종 결과물은 아래와 같다.

['손흥민', '한국', '자카르타', '드리블', '우승', '슈팅', '아시안게임']

3.3 Python 에서 사용하기

본 논문이 제안한 명사 추출 알고리즘을 Python 에서 사용하기 위해서는 kornounextractor 모듈을 설치해야 한다. 설치는 윈도우의 경우 명령 프롬프트 창 (Mac 의 경우는 터미널)에서 아래와 같이 pip install 명령어를 사용하면 된다.

```
pip install kornounextractor
```

해당 모듈을 설치한 다음에, 아래와 같이 명사 추출에 사용되는 extract() 함수를 import 하여 해당 명사 추출기를 사용할 수 있다.

```
from kornounextractor.noun_extractor import extract
```

extract() 함수는 다음과 같은 파라미터를 가지고 정의되어 있다.[6]

```
def extract(text, include_number = False, freq=1.0, threshold=0.3)
```

text: 분석하고자 하는 텍스트 데이터. 텍스트 데이터는 하나의 문서가 될 수도 있고, 복수 문서의 집합이 될 수도 있다. 본 함수는 대용량 텍스트에 대해서는 그 처리 속도가 느리므로, 작은 수의 문서들을 대상으로 작업할 것을 권고한다.

include_number: 숫자를 포함하고 있는 단어의 추출 여부. 기본값은 False (즉, 추출하지 않는다)로 지정되어 있다.

freq: 추출하고자 하는 단어들이 text 에서 사용된 최소 빈도. 예를 들어, freq=3 이라고 입력하게 되면 text 에서 최소 3 번 이상 사용된 단어들만 추출하게 된다. 기본값은 1.0 으로 지정되어 있다.

threshold: 명사 추출에 사용되는 기준값. 본 연구에서는 명사 단어를 추출할 때, 다음과 같은 값을 사용한다.

$$\omega_i = \text{Word}_i \text{ 가 조사와 함께 사용된 어절의 수} \Big/ \text{Word}_i \text{ 가 사용된 전체 어절의 수}$$

이는 명사의 사용 특성 중, '조사와 주로 사용된다'라는 특성을 고려한 것이다. 즉 특정 단어 (Word_i)가 조사와 함께 많이 사용될 수록 Word_i가 명사일 확률 높다라고 본 논문은 가정한다. 본 연구에서는$\omega_i > \text{threshold}$ 인 단어들만을 명사 단어 후보로 추출한다.

extract() 함수 사용 예

extract() 함수의 구체적인 사용 예를 설명하기 위해 아래와 같은 신문기사가 있다라고 가정하겠다.

[6] 함수의 Python code 는 게재 확정시 공개할 예정

토트넘이 긴장해야 할 지도 모른다. 손흥민의 바이에른 뮌헨 이적설이 끊이질 않는다.

영국 유력지 '가디언'은 22 일(한국시간) "마우리시오 포체티노 감독에게 좋지 않은 소식이다. 내년 1 월 뮌헨이 손흥민 영입 추진하고 있다. 손흥민은 분데스리가 챔피언 뮌헨으로 떠날 수도 있다"라고 설명했다.

손흥민의 이적설은 이탈리아 현지에서 흘러 나왔다. 이탈리아 일간지 '칼치오메르카토'의 최초 보도 이후 영국과 일부 독일 언론에서 재해석해 이적설을 전했다. 당시 언론들은 손흥민이 최근 재계약을 체결했다는 점을 근거로 가능성을 낮게 점쳤다.

'가디언'도 "손흥민이 이적설을 민감하게 반응하진 않을 것이다. 뮌헨 이적설을 토트넘과 연봉 협상에 이용할 수도 있다"라고 전했다. 그러나 '가디언'의 보도로 뮌헨 이적설이 재점화 된 점을 고려하면, 토트넘 입장에서 흘려 들을 만한 이야기는 아니다.

손흥민은 2015 년 토트넘 입단 전까지 독일 분데스리가에서 활약했다. 언어와 리그 적응에 문제가 없단 점과, 뮌헨이 유럽축구연맹(UEFA) 챔피언스리그 우승에 도전하는 점을 돌아보면 매력적인 팀이다. 2018 자카르타-팔렘방 아시안게임으로 더 이상 병역 문제도 없다. 축구공은 둥글기에 어떤 일도 일어날 수 있다.

그러나 내년 1 월 이적에는 회의적이다 . 손흥민은 잉글랜드 프리미어리그 적응 이후 2 시즌 동안 두 자리 득점에 성공했다. 리그와 챔피언스리그가 본격적으로 접어드는 시점에 토트넘이 손흥민이 내줄 가능성은 낮다.

<그림 1. 신문기사의 예>

이 신문기사의 내용이 example.txt 라는 텍스트 파일에 저장되어 있다고 하는 경우, 우리는 아래와 같은 코드를 통해서 해당 텍스트 내용을 Python 으로 불러 올 수 있다.

```
with open('example.txt', 'r', encoding='utf8') as f:
    news_article = f.read()
```

news_article 변수에 저장되어 있는 해당 기사의 내용을 가지고 명사 추출을 위한 extract()함수를 아래와 같이 호출 할 수 있다.

extract(text)

위 함수 호출은 아래와 같이 61 개의 명사 후보 단어들을 추출한다.

['UEFA', '가능', '가디언', '감독', '고려', '근거', '긴장', '내년', '도전', '독일', '득점', '리그', '마우리시오', '매력', '문제', '뮌헨', '바이에른', '반응', '병역', '보도', '분데스리가', '소식', '손흥민', '시점', '아시안게임', '언론', '언어', '연봉', '영국', '영입', '우승', '유럽축구연맹', '유력지', '이야기', '이용', '이적설', '이탈리아', '일간지', '일부', '입단', '입장', '잉글랜드', '자리', '자카르타팔렘방', '재계약', '재점화', '재해석', '적응', '챔피언스리그', '최근', '최초', '추진', '축구공', '칼치오메르카토', '토트넘', '포체티노', '프리미어리그', '한국시간', '현지', '협상', '회의']

만약, 더 자주 출현하는 명사 단어들만을 추출하고자 한다면, freq 의 값을 증가시키면 된다. 예를 들어,

extract(text, freq=2.0)

라고 입력하게 되면, 아래와 같이 17 개의 명사 후보 단어들이 추출된다.

['가능', '가디언', '내년', '독일', '리그', '문제', '뮌헨', '보도', '분데스리가', '손흥민', '언론', '영국', '이적설', '이탈리아', '적응', '챔피언스리그', '토트넘']

3.4 Python 의 기본 형태소 분석기 성능과의 비교

이를 Python 의 KoNLPy 에서 제공이 되는 다른 형태소 분석기의 결과와 비교를 해보도록 하겠다.

1) Komoran 을 사용하는 경우

Komoran 을 통해 명사로 추출된 단어들 중에서 1 음절의 단어들은 제외한 경우, 아래와 같이 82 개의 명사가 추출된다.

['가능성', '가디언', '감독', '게임', '고려', '근거', '긴장', '내년', '당시', '도전', '독일', '동안', '득점', '리그', '매력', '메르', '문제', '뮌헨', '바이에른', '반응', '병역', '보도', '보도로', '본격', '분데스리가', '설명', '소식', '손흥민', '수도', '시간', '시오', '시점', '아시안', '언론', '언어', '연맹', '연봉', '영국', '우리', '우승', '유럽', '유력', '이상', '이야기', '이용', '이적', '이적설', '이탈리아', '이후', '일간', '일도', '일부', '입단', '입장', '잉글랜드', '자리', '자카르타', '재계약', '재해', '적응', '점화', '지도', '챔피언',

'챔피언스리그', '체결', '최근', '최초', '추진', '축구', '축구공', '카토', '칼치', '토트넘', '티노', '팔렘방', '포체', '프리미어리그', '한국', '현지', '협상', '활약', '회의']

2) Twitter 분석기를 사용하는 경우

Twitter 분석기를 이용하는 경우, 1 음절의 명사 단어들을 제외하면, 아래와 같이 82 개의 명사가 추출된다. 이 예제 기사에 대해서는 Komoran 의 결과와 동일하게 나왔는데 보통은 Komoran 의 결과와 다르다.

['가능성', '가디언', '감독', '게임', '고려', '근거', '긴장', '내년', '당시', '도전', '독일', '동안', '득점', '리그', '매력', '메르', '문제', '뮌헨', '바이에른', '반응', '병역', '보도', '보도로', '본격', '분데스리가', '설명', '소식', '손흥민', '수도', '시간', '시오', '시점', '아시안', '언론', '언어', '연맹', '연봉', '영국', '우리', '우승', '유럽', '유력', '이상', '이야기', '이용', '이적', '이적설', '이탈리아', '이후', '일간', '일도', '일부', '입단', '입장', '잉글랜드', '자리', '자카르타', '재계약', '재해', '적응', '점화', '지도', '챔피언', '챔피언스리그', '체결', '최근', '최초', '추진', '축구', '축구공', '카토', '칼치', '토트넘', '티노', '팔렘방', '포체', '프리미어리그', '한국', '현지', '협상', '활약', '회의']

위의 결과에서 보이는 것 처럼 본 논문에서 제시된 명사 추출기의 경우 고유명사, 외래어와 같은 미등록 단어를 더 잘 찾는 것으로 나왔다 (예, 마우리시오, 포체티노, UEFA 등). 하지만, '가능성' ' 당시' 등과의 일반 명사는 기존의 형태소 분석기가 더 잘 찾는 것으로 나왔다. 많은 실험 결과, 이러한 특성은 다른 문서에 대해서 비슷하게 나타나는 것으로 밝혀졌다.

3.5 Python 의 기본 형태소 분석기와의 혼용

본 논문에서 제시한 명사 추출 방법을 단독으로 사용할 수도 있지만, 실제 텍스트 분석에서 분석 결과의 정확도를 높이기 위해서는 본 연구에서 제안하는 명사 추출기와 Python 의 KoNLPy 에서 제공하는 기본 형태소 분석기 (예, Komoran 등)를 혼합하여 사용하는 것이 더 적합한 경우도 있다.

Komoran 은 다음과 같이 새로운 단어들을 일시적으로 Komoran 이 사용하는 형태소 사전에 추가시킬 수 있다. 여기서 '일시적'이라는 말은 Python 프로그램이 실행되는 동안에만 추가가 된다라는 것을 의미한다. 즉, Python 프로그램을 새롭게 실행하는 경우에는, 다시 Komoran 사전에 추가하는 작업을 해주어야 한다.

```
from konlpy.tag import Komoran

komoran = Komoran(userdic='new_words.txt') #새로운 단어를 사전에 포함시키기
```

new_words.txt 파일은 명사 추출기를 통해 추출된 명사들의 정보를 아래와 같은 형태로 저장하고 있다. 단어와 단어의 품사의 내용이 각 줄에 저장되어 있는데, 단어와 품사 정보 사이에 탭 (\t) 구분자가 존재한다.

```
손흥민 NNG
한국 NNG
자카르타 NNG
드리블 NNG
슈팅 NNG
아시안게임 NNG
```

<그림 2. new_words.txt 파일의 예>

새로운 명사 단어들이 일반 명사 (NNG)인지 고유 명사(NNP)인지 알기 어려우므로 보통 모두 일반 명사로 저장을 한다. 이렇게 해도 Komoran 에 의해서 명사로 인식되기 때문에 명사를 추출하여 사용하는데 문제가 없다. 그 다음에 komoran.pos(text) 를 통해서 모든 단어의 품사태깅을 할 수도 있고, komoran.nouns(text)을 이용해서 명사 단어들만 추출할 수 있다.

4. 논의

본 논문은 온라인에 있는 한글 텍스트 데이터를 형태소 분석기를 이용해 분석하는데 있어 발생할 수 있는 명사 단어의 미등록 문제를 해결하기 위한 알고리즘을 제안하였다. 많은 텍스트 분석에서 정확한 명사 단어 추출이 중요함에도 불구하고 인문사회 분야에서 이뤄지는 관련 연구들은 명사 단어들에 대한 미등록 단어 문제를 제대로 처리하지 않는 경향 있다. 이는 인문사회 분야 연구자들이 상대적으로 코딩 기술이 부족하기 때문일 수 있다. 이에 따라, 본 연구에서는 명사 추출 알고리즘 제안하고, 실제로 Python 을 통해서 독자들이 쉽게 설치·구현할 수 있게 하였다.

본 연구에서 제안한 명사 추출 알고리즘은 명사의 사용 정보와 Komoran 라는 기존 형태소 분석기를 혼용하여 명사 추출의 정확성을 높이는 방법을 기반으로 하였다. 기존 형태소 분석기를

단독으로 사용할 때와 비교하여, 본 명사 추출 알고리즘은 다음과 같은 장점이 있는 것으로 나타났다.

① 고유 명사, 외래어, 신조어 등과 같은 미등록 명사를 잘 찾는다.
② 추출 기준으로 단어의 사용 빈도를 사용할 수 있다.
③ 숫자 포함 여부를 지정할 수 있다.
④ 조사가 포함된 어절의 수를 조절하여 명사를 추출할 수 있다.

한계점

대부분의 알고리즘이 그러하 듯 본 연구에서 제안한 알고리즘에도 한계점이 존재한다. 주요한 한계점으로는 다음과 같은 것들이 있다.

① 대용량 텍스트 입력시 처리 속도가 느리다.

따라서 본 논문에서 제안한 알고리즘은 분석 텍스트의 양을 많지 않은 경우에 적합하다. 예를 들어, 분석하고자 하는 텍스트 데이터가 많은 수의 신문기사의 집합이라면, 모든 신문기사에 대해서 한번에 명사를 추출하기 보다는 각 신문기사의 명사를 별도로 추출할 것을 권한다.

② 추출이 잘 안되는 단어들이 존재한다.

본 연구에서 제안한 명사 추출 알고리즘은 조사 정보를 사용하여 명사를 추출하기 때문에 끝 음절이 조사와 소리가 같은 명사는 끝 음절이 제외되고 추출되는 문제 발생한다. 이를 해결하기 위해 그러한 명사들을 저장하는 별도의 사전 사용 하거나, 코딩을 통해 후처리 작업을 해주어야 한다.

③ 띄어쓰기가 잘 안되어 있는 텍스트에는 부적합하다.

본 연구에서 제안한 명사 추출 알고리즘은 문장을 어절로 구분하고 각 어절에서 명사를 추출하는 방법을 사용하고 있기 때문에, 어절 추출이 어려운 텍스트에서는 그 성능이 떨어진다. 문장은 띄어쓰기를 기준으로 어절로 결합되어 있기 때문에, 띄어쓰기가 잘 안되어 있는 텍스트 경우는 어절 추출이 어려워 본 명사 추출기가 제대로 작동하지 않는다.

이러한 한계점은 지속적인 개선 작업을 통해 보완할 예정이다.

5. 결론

앞에서 언급한 것 처럼 본 연구에서 제안한 명사 추출 알고리즘은 몇 가지의 한계점을 가지고 있다. 하지만, 그럼에도 불구하고 기존의 형태소 분석기를 사용함에 있어 발생할 수 있는 명사 단어의 미등록 문제를 해결하는데 큰 도움이 된다. 본 명사 추출 알고리즘을 통해 인문사회 분야의 연구자들이 온라인 텍스트 분석을 하는 과정에서 겪는 어려움이 줄어들기를 바라며, 보다 많은 연구자들이 컴퓨터 기반 한글 텍스트 분석에 관심을 갖기를 기대해 본다.

참고문헌

감미아, 송민. (2012). 텍스트 마이닝을 활용한 신문사에 따른 내용 및 논조 차이점 분석. 지능정보연구, 18(3), 53-77.

강승식. (2002). *한국어 형태소 분석과 정보검색*. 홍릉과학출판.

고창운. (2006). *국어 형태론의 기초*. 푸른사상

박용현, 황재원, 고영중. (2010). 한국어 명사 출현 특성과 후절어를 이용한 명사 추출기. 정보과학회논문지 : 소프트웨어 및 응용, 37(12), 919-927.

심주영. (2017). 용산미군기지 공원화 과정의 도시담론 분석. 한국도시설계학회지 도시설계, 18(5), 37-52.

심준혁, 김준성, 차정원, 이근배. (1999). 통계와 규칙을 이용한 강인한 품사 태거. 한국정보과학회 언어공학연구회 학술발표 논문집, 60-75.

안동언. (1999). 좌우접속정보를 이용한 명사추출기. 한국정보과학회 언어공학연구회 학술발표 논문집, 173-178.

양희수, 현은정. (2018). 문화예술 분야에서 '욜로(YOLO)'의 활용 및 확산. 문화와융합, 40(1), 29-66.

언어정보연구원. (2018). 연세한국어사전. Retrieved 6 월 1 일, 2018, from https://ilis.yonsei.ac.kr/콘텐츠/연세현대한국어사전/

유현경, 서상규, 한영균, 강현화, 고석주, & 조태린. (2015). *우리말 연구의 첫걸음*: 보고사.

윤지운, 한민규. (2016). 텍스트마이닝을 활용한 장애인스포츠관련 신문기사의 중심어 분석. 전국체육대회기념, , 237-237.

이도길, 이상주, 임해창. (2003). 명사 출현 특성을 이용한 효율적인 한국어 명사 추출 방법. 정보과학회논문지 : 소프트웨어 및 응용, 30(1·2), 173-183.

이중영, 신병훈, 이공주, 김지은, 안상규. (1999). COM 기반의 다목적 형태소 분석기를 이용한 명사 추출기. 한국정보과학회 언어공학연구회 학술발표 논문집, 167-172.

좌미라. (2018). 해양쓰레기 관련 신문기사 텍스트 네트워크 분석. 한국해양환경·에너지학회 학술대회논문집, , 85-85.

Jeon, H. (2016). Introduction to KoNLP API. Retrieved 10 월 1 일, 2018, from https://cran.r-project.org/web/packages/KoNLP/vignettes/KoNLP-API.html

KoNLPy: Korean NLP in Python. (2018). Retrieved 10 월 1 일, 2018, from http://konlpy.org/en/latest/

SHINEWARE. (2018). KOMORAN – Java 기반의 한국어 형태소 분석기. Retrieved 10 월 1 일, 2018, from http://www.shineware.co.kr/products/komoran/