# 대량 문서의 사후 불용어 처리와 형태소 분류기의 차이에 따른 TF-IDF 행렬 비교

20182800 나요셉

## 연구배경

텍스트 마이닝은 대량의 텍스트 자료에서 유용한 정보를 찾아내는 작업을 일컫는다. (Kim, 2013) 이러한 텍스트 마이닝 기법은 GPU의 성능이 늘어나고 오픈 된 데이터의 양이 많아 짐에 따라 텍스트 마이닝의 연구가 활발히 진행되고 있다. 또한 대용량의 텍스트를 학습한 후 단어를 컴퓨터에 맞게 적절히 Embedding 해주는 모델들도 많이 오픈 돼있어 텍스트 마이닝이 활발해지는 추세는 점점 늘어나고 있다.

텍스트 마이닝 기법을 사용하여 텍스트 안의 유용한 정보를 찾아내는 것은 데이터 마이닝의 일종이라고 할 수 있지만, 직접적으로 분석이 가능한 숫자로 이루어진 일반적인 정형 데이터와는 다르게 텍스트 데이터는 다양한 변이형 작업을 포함하는 비정형 데이터이다.

따라서 텍스트 마이닝은 우선 이러한 텍스트 데이터를 분석이 가능한 형태로 가공을 우선적으로 진행해야 한다. 이러한 과정을 데이터 전 처리라고 하는데, 이러한 전 처리 과정은 오탈자 제거, 누락된 표현 복원, 대/소문자 통일 등의 작업을 거친다. 또한 이 단계에서 불용어 처리의 단계가 있다. 대부분의 텍스트 데이터 분석을 하는 이유는 앞서 말했듯 텍스트의 의미를 분석하고 유의미한 결과를 도출하기 위함이다. 하지만 텍스트 데이터에 불용어가 많다면, 의미를 분석하는 데에 필요하지 않은 단어들이 많아지므로 유의미한 결과를 도출하는 것이 어려울 것이다.

일반적으로 영어에서 불용어란 일반 관사 (a, an, the) 전치사 (for, of, in, through) 등이 있으며 이러한 단어들은 자주 사용되긴 하지만 어휘나 문장에서 큰 역할을 한다고 보기 어렵기 때문에 불용어로써 제거된다. 또한 이러한 불용어를 제거하는 것이 텍스트 마이닝을 할 때 혼잡도를 일반적으로 더 낮춘다고 알려져 있다. (Gil, 2018)

일반적으로 영어 텍스트 분석의 경우 불용어 처리 목록이 존재하는 반면, 한국어로 만들어진 불용어 목록은 아직까지 연구가 활발히 진행되고 있지 않다. (Gam & Song, 2012) 한국어의 경우 영어보다 더 유동적인 언어에 속하면 한국어로 나타낼 수 있는 조합이 매우 많고 표현도 다양하기 때문에 정형화된 불용어 목록을 구축하기에 힘든 것이 있다. 하지만 연구가 활발하지 않음에도 불구하고 한국어 불용어 목록의 시안을 제시하는 Prototype의 논문은 찾아볼 수 있다. (Gil, 2018, Ju, 2008, Lee, 2021)

불용어는 목록을 구성하는 방법은 일반적으로 두 가지가 있는데, 언어학적인 분석을 통하여 목록을 구성하는 방법과 어휘 출현을 빈도로 하여 구성하는 방법이 있다. 본 연구는 후자의 방법으로 불용어 처리를 진행하고자 하였다. 앞선 사전 연구들은 공통적으로 말뭉치를 통한 고 빈도 어휘 목록을 추출하고 문법적 기능을 수행하는 품사를 추출하여 불용어를 선정하는 방법론을 취하고 있기 때문이다.

따라서 본 연구는 사전 논문에서 제시한 불용어 처리 방법론을 차용하여 불용어를 처리하는 방법에 따른 TF-IDF를 비교하고 단어들의 중요도 양상을 비교할 것이다. 불용어 처리에 따른 TF-IDF Matrix의 양상을 비교하고자 하는 이유는 TF-IDF 행렬 안에서 고빈도 어휘에 대한 가중치를 낮게 주어 상대적인 중요도를 낮게 추출하긴 하지만, 해당 Matrix의 Column Space인 단어 집합 공간의 차원은 불용어 처리 이후 더 감소할 것이기 때문에 열공간의 차원이 감소한 Matrix는 불용어를 처리하기 전보다 더 Dense한 Matrix가 될 것이고 Matrix의 양상도 바뀔 것이라 가정했기 때문이다.

## 관련 선행연구

Rajaraman, A. & Ullman, J. D.(2011)은 어느 언어에나 통용되는 불용어는 없다고 주장한다. 이러한 주장은 근거가 있는 주장이기는 하지만, 근본적인 언어의 구조, 문법적 활용은 문장 구조의 위치만 다를 뿐 단어 간의 상호작용은 비슷하기 때문에 불용어 처리의 방법론 자체는 언어별로 크게 다르지 않을 것이다.

영어에서 일반적으로 가장 많이 사용되는 불용어 목록은 1990년 Fox의 ‘a stop list’ 이다. Fox는 Brown corpus 라는 말뭉치를 기반으로 빈도가 높은 어휘를 추출한 후 불용어 목록의 표준적인 모형을 구축했다. 또한 영어 문서에서 총 1백만 14000개의 단어를 추출하여 각 단어의 빈도를 확인하였다. 또한 Fox에서 불용어를 처리한 흥미로운 방법은 빈도가 높은 어휘를 추출하여 의미를 분석한 후 불용어 목록에 추가한 것 외에 상대적으로 적게 나온 단어에 대해서 글을 쓴 사람의 ‘어투’ 라고 가정하여 해당 단어들도 불용어 목록에 넣어 보편적인 의미를 가진 단어들만 추출하고자 하였다. (Chirstopher Fox, 1990)

또한 Sifa-tullah Siddiqi & Aditi Sharan(2018)은 영어의 불용어 리스트를 힌디어로 번역한 후 힌디어에서 사용되지 않은 언어를 제거한 후 필요한 어휘를 추가하는 형식으로 불용어 목록을 구축하였다.

한국어의 경우도 불용어 연구가 과거 시행이 되었는데, 검색 엔진에서 무의미한 색인어를 제거하여 검색의 효율성을 증대하는 목적으로 연구가 진행되었다. 색인어란 정보 검색 시스템에서 어떤 문서에 대해 그 문서의 전체적 내용을 나타내거나, 그 문서를 다른 문서 들로부터 구별할 수 있도록 그 문서의 선택단서가 되는 단어 또는 단어 구 등을 의미하며 문서를 유사한 것들끼리 묶을 수 있는 능력을 가지고 있다. 김판구, 조유근(1993)의 연구에서는 자동으로 무의미한 색인어를 제거하기 위해 불용어 목록을 만들어 제시하였지만 해당 불용어 목록은 뚜렷한 과정이나 방대한 데이터로부터 고 빈도 어휘를 추출하여 불용어 목록을 제시한 것이 아니라 언어학적 분석을 통해 제시한 것이기 때문에 신뢰를 하기에 한계가 있다.

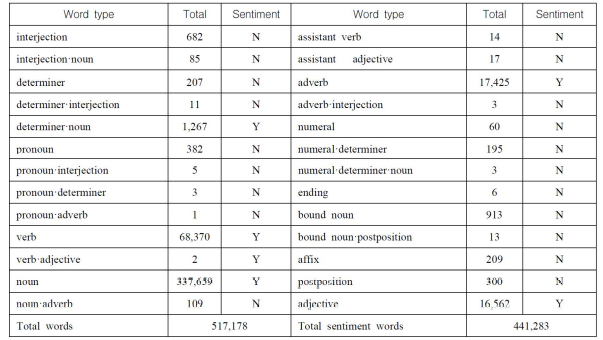


Fig1. 감정어 사전 구축 표

해당 표의 경우 Gam & Song(2012)의 연구에서 감정어 사전을 구축하는 목적을 갖고 한국어 어휘를 26개의 품사로 나누었으며 품사별 빈도 산출 후 이에 대해 감정 표현 여부를 판단하여 분류를 한 것이다. 이는 불용어 목록이 아닌 감정어 사전 구축을 위해 해당 감정에 대한 품사별 표현을 표로 나타낸 것이기 때문에 불용어 목록으로 사용할 수 없지만 해당 표에서 제시한 26 개의 품사 별 감정 표현 여부를 판단하는 목적을 바꿔 품사 별 불용어 여부를 판단하는 기준으로 사용하여 불용어 목록 구축의 틀을 만들 때 참조할 수 있다.

또한 불용어 선정에 앞서 어떤 품사의 어휘를 불용어로 처리할지에 대한 논의가 필요하다. 선행 연구의 경우 대부분 감탄사, 조사, 부사와 접속사를 불용어 품사를 선정하였는데, 권호경(1996)의 연구에서 불용어 품사 선정 방식을 A-B Test 로 진행한 결과 선행 연구에서 불용어 품사로 가정한 감탄사, 조사, 부사, 접속사를 불용어 품사로 지정하여 처리하는 것이 명사와 동사 등도 불용어 품사로 포함하여 진행한 것 보다 더 낮은 혼잡도를 보인다고 주장헸다.

또한 TF-IDF를 활용하여 문서 행렬을 만든 뒤 해당 문서에서 TF-IDF값이 낮은 단어를 불용어로 처리하여 텍스트 데이터 전처리를 한 장민승(2020) 연구에서는 우선 대표적인 불용어를 사전에 리스트 업 시킨 후 TF-IDF를 구해 값이 낮은 단어들을 확인하여 불용어 사전에 추가하였으며, 불용어 처리가 완료된 텍스트를 토대로 MLP 모델을 통해 각 텍스트 별 주제를 분류하는 Task를 수행한 결과 유의미한 성능 지표를 확인할 수 있었다.

여기서 TF-IDF란 자연어 처리에서 텍스트를 표현하는 방법 중 하나라고 할 수 있는데, 카운트 기반의 텍스트 표현 방법인 문서 단어 행렬(DTM)과 TF-IDF(Term Frequency-Inverse Document Fr-equency)가 주로 사용된다. 텍스트가 위와 같은 방법으로 수치화가 된다면, 여러 문서로 이루어진 텍스트 단어에 대해서 특정 문서 내에 얼마나 영향력을 갖는지 알 수 있다.

우선 문서 단어 행렬이란 다수의 문서에 등장하는 각 단어들의 빈도를 행렬로 표현한 것인데, 예를 들어 4개의 문장이 있다고 가정하자.

1. 조선의 왕은 왕비를 사랑한다
2. 조선의 서민은 왕비를 사랑한다
3. 왕은 부지런한 국민을 국민을 좋아한다.
4. 부지런한 서민은 부지런한 왕비를 사랑한다.

해당 4가지 텍스트에 대해서 단어의 빈도를 고려하여 행렬을 표현한다면

|  | **조선의** | **왕은** | **왕비를** | **사랑한다** | **서민은** | **부지런한** | **국민을** | **좋아한다** | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| d1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | | 0 |
| d2 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | | 0 |
| d3 | 0 | 1 | 0 | 0 | 0 | 1 | 2 | | 1 |
| d4 | 0 | 0 | 1 | 1 | 1 | 2 | 0 | | 0 |

표 1.DTM의 예시

해당 표와 같이 각 행은 각 문서이며 각 열은 모든 문서에서 나온 단어의 종류가 되고 각 값은 해당 문서에서 얼마나 그 단어가 나왔는지에 대한 것으로 표현할 수 있다.

이러한 DTM 구조는 매우 단순하게 표현될 수 있다는 장점이 있으며 직관적이지만 명확한 단점이 존재한다. 예를 들어 각 문서가 방대한 양의 단어들을 갖고 있다면 열 공간에 들어가는 단어의 수가 많아지며 행렬이 매우 Sparse해질 수 있으며 자주 등장하는 단어가 DTM에서 가장 값이 큰 즉, 영향을 가장 많이 주는 단어라고 해석할 수 있지만 이러한 해석이 옳은 경우는 일반적으로 드물기 때문이다.

예를 들어 영어에서 ‘The’라는 단어는 매우 많이 등장하지만 각 문서의 특징을 알아내는 것에 ‘The’라는 단어는 별다른 영향을 끼치지 않기 때문이다.

따라서 이러한 단어 빈도수 표현으로만 문서에서 단어의 영향력을 측정하는 것은 올바르지 못한 해석이라고 할 수 있으며 이러한 단점을 일부 보완한 것이 TF-IDF라고 할 수 있다.

TF-IDF란 단어 빈도와 역빈도를 모두 고려한 것인데, DTM내의 각 단어들마다 중요한 정도를 가중치를 부여하여 계산하는 작업이다. 보통의 경우 TF-IDF는 DTM보다 좋은 성능을 얻기 때문에 주로 TF-IDF를 문서에 관해 각 단어의 영향력을 비교할 때 주로 사용한다. 이러한 과정에서 각 가중치는 학습 가능한 파라미터가 아닌 수식에 의해 계산이 되는데, 전체 문서 수 대비 특정 단어가 얼마나 많은 문서에 출현했는지에 대해 계산한 후 로그를 씌우는데 이것이 가중치가 되어 단어의 등장 횟수에 곱해주게 된다.

만약 구하고자 하는 특정 단어가 극단적으로 모든 문서에 등장하게 되면 가중치 값이 0이 되므로 문서 단어 행렬에서 해당 단어는 영향력이 없는 단어가 된다. 이러한 해석은 타당할 수 있는데, 문서와 단어의 관계를 나타내는 행렬에서 모든 문서에 나타난 단어는 문서를 특정 지음에 있어 영향을 주지 않기 때문이다.

이 연구에서 확인할 수 있는 것은 분석을 통해 불용어 사전을 정의한 후 TF-IDF를 수행하여 값에 따른 불용어를 추가한 결과값을 통한 텍스트 전처리를 진행하는 것이 분류 모델 성능을 높이는 것에 일정부분 영향을 준다는 것이다.

## 제안 방법론

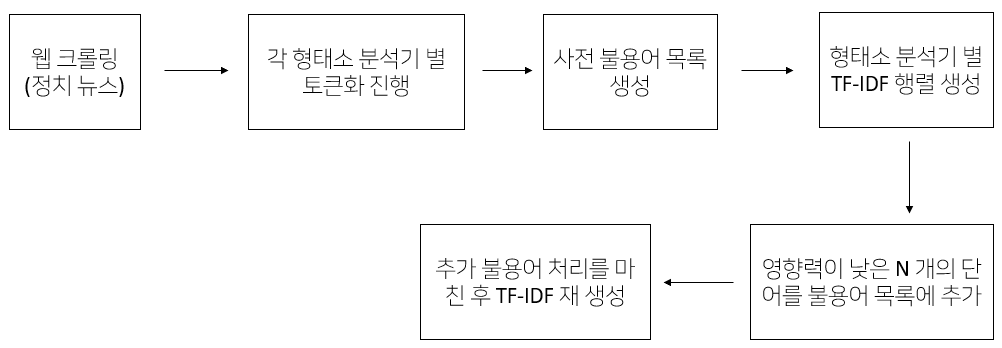


Fig 2. 방법론 모형

우선 본 연구는 웹 크롤링을 통해 네이버 뉴스기사 중 대선 기간 요일 별 탑 랭킹 뉴스들을 추출하였다. 이 기간의 텍스트 데이터를 추출한 이유는 대선 기간 중 빈도가 높은 어휘들이라 많을 것이라고 가정하였으며 특정 사건이 발생했을 때 지엽적인 단어들이 많이 나와 불용어가 많을 수 있다고 판단하여 해당 단어들을 불용어 처리할 때와 하지 않았을 때의 차이가 명확히 드러날 것이라고 생각했기 때문이다.

이후 각 형태소 분석기 별 텍스트 데이터 토큰화를 진행하였다. 여기서 형태소 분석기 별 토큰화를 진행한 이유는 각 형태소 분석기에 따라 품사 처리 방식과 형태소 분석 방식이 다르며 하나의 품사를 추출한다고 하더라도 형태소 분석기의 품사 지정 방식이 다르기 때문에 단어의 종류가 바뀌고 TF-IDF 행렬의 열공간이 달라지기 때문이다. 또한 단어의 표기는 같지만 품사에 따라서 단어의 의미가 달라지기 때문에 형태소 분석기 별 품사 지정 방식이 다른 것은 형태소 분석기에 종류에 따라 들어오는 문서를 다르게 해석하는 것을 내포한다고 판단하였기 때문에 각 형태소 분석기에 대해 이후 과정을 동일하게 진행하였다.

형태소 분석은 우선 Python 환경에서 자연어처리를 제공해 주는 패키지인 KoNLPy 패키지를 활용하였다. 해당 패키지 내부에 있는 Okt(Open Korea Text), 코모란(Komoran), 한나눔(Hannanum), 꼬꼬마(Kkma) 총 네 가지의 형태소 분석기를 사용하여 크롤링 한 모든 문서 각각의 토큰화를 진행하였다.

예를 들어 “텍스트 마이닝 중간고사, 보고서 작성하고 있습니다” 다음과 같은 문장이 있다고 할 때 해당 문자에 대해 품사 추출, 형태소 분석을 진행한 결과에 대해서 OKT, KKma 형태소 분석기로 진행한 결과에 차이가 존재하였다.

OKT의 경우 ['텍스트', '마', '이닝', '중간고사', ',', '보고서', '작성', '하고', '있습니다']의 형태로 형태소 분석이 진행된 것을 확인하였고 [('텍스트', 'Noun'), ('마', 'Noun'), ('이닝', 'Noun'), ('중간고사', 'Noun'), (',', 'Punctuation'), ('보고서', 'Noun'), ('작성', 'Noun'), ('하고', 'Josa'), ('있습니다', 'Adjective')]의 형태로 품사 추출이 된 것을 확인할 수 있다.

KKma의 경우 ['텍스트', '마이닝', '중간고사', ',', '보고서', '작성', '하', '고', '있', '습니다']의 형태로 형태소 분석이 진행되었으며 품사 추출의 경우 [('텍스트', 'NNG'), ('마이닝', 'NNG'), ('중간고사', 'NNG'), (',', 'SP'), ('보고서', 'NNG'), ('작성', 'NNG'), ('하', 'XSV'), ('고', 'ECE'), ('있', 'VXV'), ('습니다', 'EFN')]로 진행된 것을 확인할 수 있었다.

따라서 모든 문서에 대해서 각 형태소 분석기 별 TF-IDF 행렬을 생성하여 불용어 처리 시 변화하는 양상을 확인하기 위해 해당 단계를 진행하였다.

이후 정치뉴스 텍스트 데이터를 크롤링하는 작업이기 때문에 언론사, 지엽적 단어들에 대해 사전 불용어 목록을 만들었으며 정규표현식을 사용하여 텍스트 이외의 불용어를 미리 제거하였다.

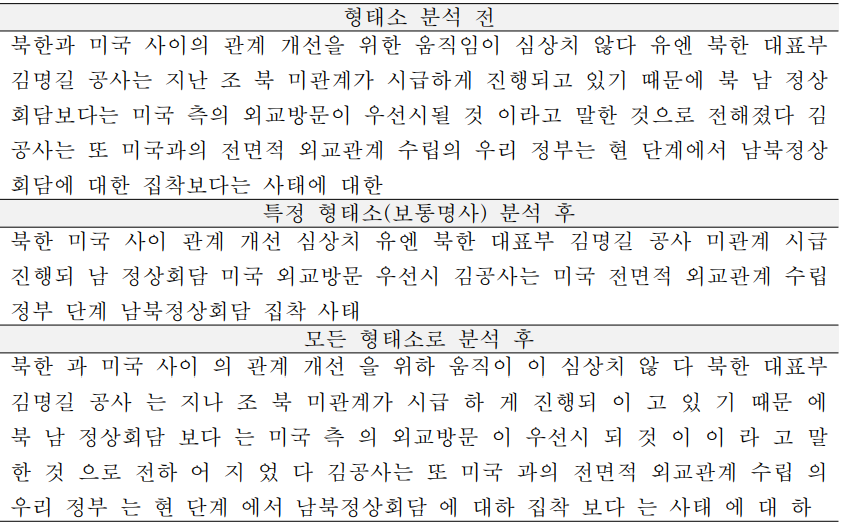
이후 앞서 지정한 네 개의 형태소 분석기 각각 TF-IDF행렬을 생성하였다. 또한 TF-IDF를 통해 문서 별 단어의 영향력을 분석하기 전 선행 연구에서 불용어 품사로 지정한 부사, 감탄사, 조사, 접속사에 대해서는 추출을 하지 않았다. 모든 품사를 고려하여 불용어를 추출하게 된다면 TF-IDF 행렬의 열 공간 차원이 매우 커지게 되어 행렬이 매우 Sparse하게 되기 때문이다. 

Fig3.형태소 개수 별 분석 결과

Fig3의 경우 앞서 이야기한 예시이다. 형태소 분석 전 많은 단어들이 존재하는데, 보통명사라는 특정 형태소로 분석을 진행한 결과가 모든 형태소로 분석을 진행할 때보다 적은 단어들이 추출된 것을 확인할 수 있으며 문서 전체의 정보에 대한 손실은 있지만 해당 문서가 어떤 특징을 가진 문서인지에 대해 파악할 수 있다. 해당 예시를 통해 문서의 길이가 늘어나고 문서의 개수가 많다면 모든 품사로 형태소 분석을 진행하여 TF-IDF를 구하는 것은 메모리적 낭비를 매우 심하게 할 수 있으며 Sparse 한 TF-IDF 행렬은 문서의 특징을 파악하는 것을 더욱 어렵게 한다고 판단하였다.

따라서 부사, 감탄사, 조사, 접속사 품사를 제외한 나머지 품사에 대해서 TF-IDF를 생성하여 수치 값이 낮은 단어들을 불용어 목록에 추가한 뒤 나머지 단어들에 대해서 다시한번 TF-IDF 행렬을 생성하여 나머지 단어들의 영향력 변화 추이에 대해 알아보았다.

이후 TF-IDF행렬 결과값에 따라 영향력이 낮은 N개의 단어를 불용어 사전에 넣은 후 최종적인 TF-IDF 행렬을 생성하여 불용어 처리가 되지 않은 단어들의 영향력이 추가적인 불용어 처리를 하지 않을 때에 비해 얼마만큼 영향력이 향상되었는지에 대해 살펴보았다. 또한 이러한 과정에서 각 단어의 분포 형태를 파악하기 위해 시각화 및 워드 클라우드를 활용하여 보다 다양한 관점에서 변화 추이를 살펴보았다.

## 실험내용

네이버 뉴스 크롤링은 대선 기간 한달 전인 2022-02-09부터 2022-03-09까지 약 한달간 진행하였으며, 각 요일 별 15개의 뉴스 본문의 크롤링을 진행하였다. 이 과정에서 네이버의 뉴스 중 Css 구조가 맞지 않는 경우가 있었기 때문에 이러한 경우 Try Except 문을 통해 예외처리를 진행하여 결과적으로 총 350개의 뉴스 본문 기사를 크롤링하였다.



Fig4. 뉴스 본문 크롤링 코드

크롤링 작업을 마친 후 350개의 문서에 대해 정규표현식에 맞게 각 본문을 문장으로 나눠주었다. 이후 각 본문에 내용을 살펴본 결과 본문에서 ‘▲, ⓒ’와 같은 특수 기호들이 사용된 데이터가 많기 때문에 각 문서에서 저러한 특수기호가 포함된 문장이 있다면 제거하여 이후 형태소 분석을 진행하는 데에 영향을 끼칠 요인을 최대한 배재하였다. 토큰화 진행 전 해당 단어에 대해 먼저 사전 제거를 시행한 이유는, 해당 단어가 각 형태소 분석기에 따라 어떠한 품사로 분류되는지 모르기 때문에 실험의 효율성을 위해 우선적으로 제거한 것이다. 따라서 최종적으로 나온 텍스트 데이터 배열은 [문서 수, 해당 문서의 문장 수, 문장의 단어 수]로 구성된 3차원 Tensor 형태가 추출되었다.

이후 각 문서를 앞서 정의한 Okt, 코모란, 한나눔, 꼬꼬마 총 네 가지의 형태소 분석기를 통해 형태소 분석을 진행하였으며 각 형태소 분석기 별로 각 품사 별 크기가 얼마나 되는지, 품사 별 최대 빈도어는 무엇인지 확인했다. 우선적으로 Okt를 통해 진행을 해 보았을 때 [것: 1538, 등: 860, 수: 831] 등 ‘것’, ‘등’과 같은 문서의 의미를 특정하는 것에 영향을 주지 않는 단어의 빈도가 매우 높게 나온 것을 확인할 수 있었다. 또한 길이가 1인 명사들을 추출함으로써 알 수 있었던 사실은 꼬꼬마의 경우 Okt, 한나눔에서 가장 빈도가 높은 명사형 단어로 분류한 ‘것’이라는 단어의 빈도가 낮게 나온다는 사실이다.

단어의 길이가 1인 명사들도 포함한 결과에서 명사 품사로 가장 많이 분류된 것은 [Okt: (것, 1538), 꼬꼬마: (후보,1336), 한나눔: (것,1730)]으로 확인되었다.

따라서 4가지 형태소 분석기로 명사를 추출하고 단어의 빈도를 비교한 결과 단어의 길이가 1인 명사들이 많이 포함되어 있다는 것을 확인할 수 있었고 이러한 단어들 중 ‘것’, ‘등’, ‘그’, ‘고’와 같은 단어는 문서의 의미를 특정-하는 것에 도움을 주지 않는다고 판단하여 사전 불용어 목록에 넣었으며, ‘말’과 같은 길이가 1임에도 뜻을 내포하는 단어와 대선 기간 후보의 성인 [‘심’,’안’,’이’,’윤’] 은 한 글자임에도 불구하고 텍스트 데이터의 특징 상 특정 뜻을 내포하는 단어라고 판단하여 해당 글자는 추출하는 방향으로 진행하였다.

또한 동사, 형용사에 대해서도 품사 추출을 진행한 결과 Okt의 경우 [했다: 2126, 하는:972, 할: 714]회가 추출되었으며 꼬꼬마의 경우 [하다: 2029, 있다: 1309, 크다: 892]회가 나온 것을 확인할 수 있었다. 한나눔의 경우와 코모란의 경우도 하다, 있다가 가장 많이 등장한 동사 품사로 나타났으며 위 결과에 따라 동사, 형용사의 품사 추출 방식도 차이가 있으나 단어의 형태 혹은 시제가 다를 뿐 실제 의미의 관점으로 바라보았을 때 추출되는 단어는 비슷하다고 판단할 수 있다.

이후 위 결과를 바탕으로 각 문서 별로 명사를 우선추출 한 뒤 명사에 대해서 TF-IDF 행렬을 만들었다. 앞서 각 형태소 분석기의 명사 추출 방식과 결과가 다르다는 사실을 확인한 후 해당 추출 방식으로부터 TF-IDF 값을 비교하는 작업도 유의미한 결과를 도출할 것이라는 가정을 통해 해당 실험을 진행하였다. 또한 동사, 형용사에 대해서 추가적으로 행렬을 만들어 불용어 처리에 따른 과정을 확인하고자 하였으나 동사 형용사의 개수가 매우 많아지고 형태소 분류기마다의 기준이 모호하여 명사로 추출한 것을 비교하고 불용어 처리를 사후에 진행할 때의 행렬의 변화만 확인할 수 있다면 사후 불용어 처리의 영향을 확인할 수 있으므로 명사에 대해서만 TF-IDF 행렬 생성을 진행하였다.

우선적으로 뉴스 본문의 개수는 총 350개 이기에 3차원 Tensor 형태로 존재하는 텍스트 데이터에 대해서 [문서, 본문 내용]의 이차원 행렬로 바꿔주는 작업을 진행하였다. 해당 작업을 진행하기 위해 우선적으로 각 뉴스 본문에 대해 형태소 분석기 별로 명사를 추출하고 명사의 종류와 수를 나열하는 행렬을 만들었다.

각 뉴스 본문 내용에 대해 전체 명사를 추출하고 해당 명사들을 새로 정의한 dictionary에 저장하는 방식을 총 네가지 형태소 분류기 모두 진행을 하였으며 이 과정에서 앞서 사전 불용어 목록에 추가한 ‘것’, ‘등’, ‘수’, ‘고’ 와 특수 문자들과 더불어 언론사 정보, 기자 정보 등의 내용도 사전 불용어 목록에 추가하여 진행하였다.

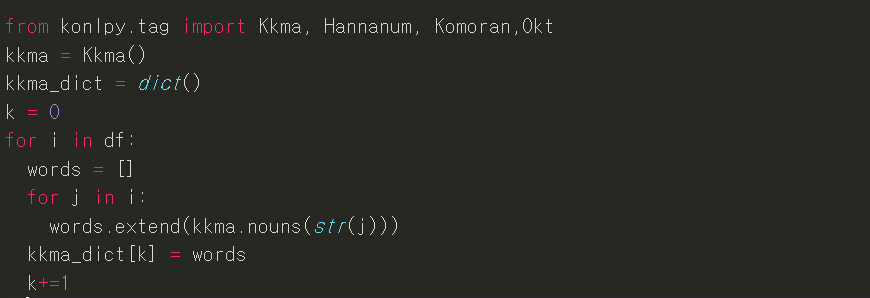


Fig5. 각 형태소 분류기 별 문서 별 명사 추출

Fig5와 같이 꼬꼬마로 명사를 추출한 경우 각 문서의 명사 단어가 index number로 표현된 Dictionary에 모두 저장된다. 이 과정을 네가지 형태소 분류기에 대해 모두 진행하였으며, 각 형태소 분류기가 추출한 모든 명사에 대한 기본 문서/단어 행렬을 생성한 결과 350개의 행에 대해 꼬꼬마는9391개의 명사, Okt의 경우 6293, 한나눔의 경우 8971, 코모란의 경우 6077개의 명사가 추출되었으며 꼬꼬마의 경우 코모란과 Okt보다 상당히 많은 단어가 추출되었음을 확-인할 수 있다.

이후 SckitLearn 패키지 내부의 Tfidfvectorizer 모듈을 통해 기본 문서/단어 행렬을 생성한 결과 행렬의 결과값이 매우 Sparse 함을 확인할 수 있었다. 각 행의 모든 원소의 TF-IDF값 평균을 비교한 결과 약 0.019 정도의 값을 지녔다. 이 값이 의미하는 것은 전체 행렬에서 유효한 숫자가 2%정도가 된다는 것이므로 사전 불용어 작업만 거친 명사를 통해 대형 문서에서 추출한 기본 문서/단어 행렬은 텍스트 데이터의 크기가 커짐에 따라 Spars해지는 것을 알 수 있었다.

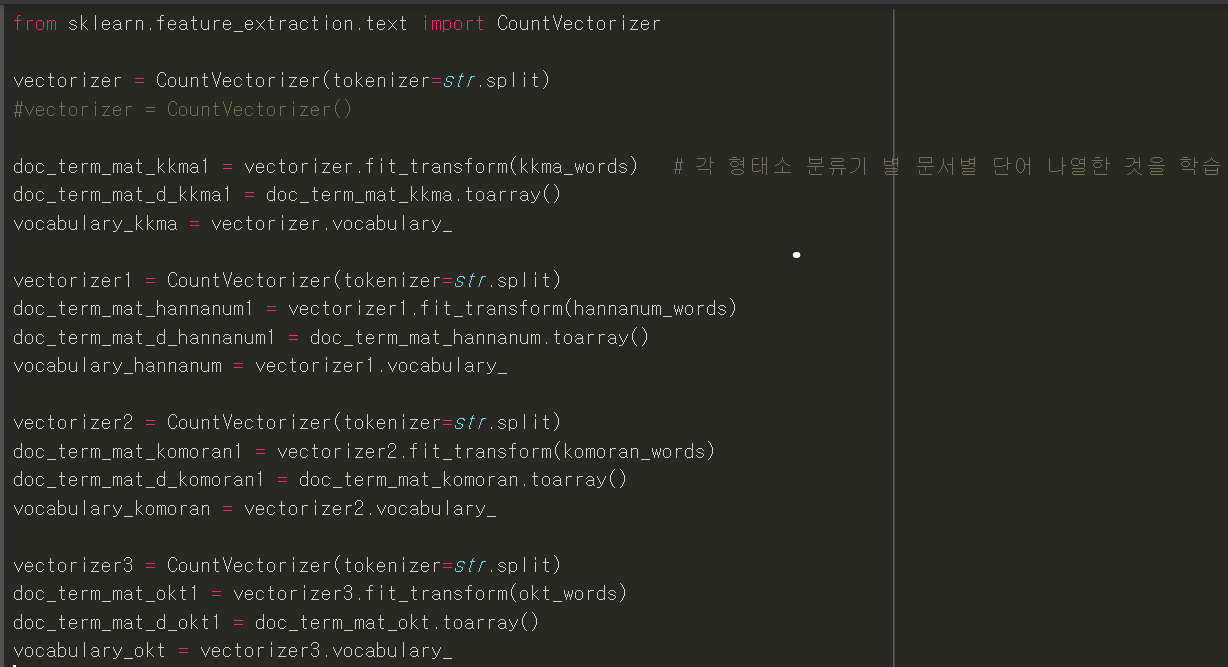


Fig6. 형태소 분류기 별 기본 문서/단어 행렬 생성 코드

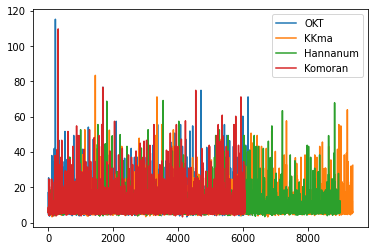
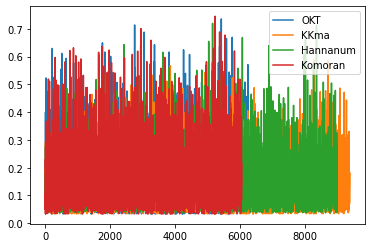
이후 정규화를 진행하지 않은 단순 TF-IDF 행렬을 생성하여 각 형태소 분류기 별 불용어 목록 추가 이전 TF-IDF를 생성한 결과에서도 기본 문서/단어 행렬과 비슷한 양상을 띈 것을 확인할 수 있는데, 여기서 TF-IDF 값이 매우 각 단어들이 특정 문서에서 높은 값을 나타-내고 있는 것을 확인할 수 있었다. 이러한 이유는 문서/단어 행렬을 구성하는 단어는, 문서의 개수가 커짐에 따라 늘어나게 되며 희귀하게 나오는 단어의 경우 단어의 종류가 매우 많기 때문에 가중치 값이 높아져 실제 계산 시 높은 값을 지니게 된 것이라고 판단할 수 있다. 

Fig7. 각 단어 별 TF-IDF 최댓값 추이 (형태소 분류기에 따라)

이후 정규화를 거쳐 각 문서 별 TF-IDF의 최댓값을 확인한 결과 앞선 결과와 같이 불 완전한 분포를 띄는 것이 아닌 대부분의 값이 비슷하게 나옴을 확인할 수 있다. 여기서 알 수 있는 사실은 TF-IDF 행렬로 문서를 표현할 때 문서와 단어의 개수가 많아져 행렬의 크기가 커지게 된다면 행렬의 분포가 커지게 되며 이 결과 각 문서의 벡터 표현 식의 분포가 커지게 된다는 것이다. 또한 이러한 분산을 낮추기 위해 정규화를 거치는 것이 행렬 표현을 더욱 응집시켜 할 수 있다고 판단할 수 있다.

Fig8. 각 문서 별 TF-IDF 최댓값 추이 (형태소 분류기에 따라, L2 정규화 적용)

이후 사후 불용어 처리의 영향을 확인하기 위해 모든 분류기에 대해 각 단어가 나온 빈도에 대해 살펴보았다.

TF-IDF 계산 시 희귀한 단어에 가중치를 크게 부여하는 것이 희소 행렬에서 각 문서들의 특징을 부여하는 것에 유의미한 영향을 주기도 하지만 가장 큰 문제는 행렬이 너무 Sparse 해진다는 것이다. 이러한 행렬에서 문제점은 자칫 어떠한 문서가 매우 희귀한 단어를 가지고 있다고 하더라도 그 단어가 의미가 없는 단어였다면 해당 문서는 그 단어에 부여된 가중치 값이 매우 커져 그 단어의 TF-IDF 값에 매몰되어 문서를 특정 지을 때 잘못된 방향으로 특정 지을 수 있다.

따라서 본 연구는 각 단어에 대해서 빈도가 문서에 나온 단어 중 얼마만큼의 문서에서 나왔는지를 확인하고 만일 그 단어가 N 개의 문서에서 출현했다면 N개의 문서에서 평균적으로 몇 번 출현했는지에 대해 확인하여 N개의 문서에서 평균적인 출현 횟수가 낮은 단어는 불용어로 처리하여 제거하였다.

이러한 방식으로 진행한 이유는 아무리 희귀한 단어라도 하더라도 해당 단어가 들어간 문서에서 출현한 횟수가 적다면 그 단어가 해당 문서의 특징을 잘 표현한다고 이야기-하기에 무리가 있다고 판단하였으며, 이러한 단어만 줄인다고 하더라도 행렬의 희소함 정도가 줄어들 것이라고 판단했기 때문이다.

각 형태소 분류기 별 단어의 출현빈도의 평균을 구한 결과 약 1.9% 정도의 값이 나왔다. 해당 결과로 알 수 있는 것은 대부분의 단어가 출현 빈도가 그리 높지 않다는 것이며, 가장 낮은 출현 빈도를 보인 단어는 0.2%의 출현 빈도를 보였으며 해당 출현 빈도를 갖는 단어는 약 5000개에서 6000개 정도가 됐다.

해당 결과로 알 수 있는 사실은 문서의 개수가 많을 때 전체 문서/단어 행렬을 통틀어 1번 나온 단어의 수가 많다는 것이다. 꼬꼬마 형태소 분류기의 경우 약 50%의 단어가 1번 나온 단어임을 확인할 수 있다. 이러한 단어는 각 문서의 특징을 특정 짓는 것에 대한 영향력 보다 단어/문서 행렬을 희소하게 표현하는 것-에 영향을 주는 단어들로 판단하여 이러한 단어를 예외처리 하였으며, 또한 각 형태소 분류기 별로 단어가 몇 개의 문서에서 나왔는지 확인 한 결과 약 90%의 문서에서 나온 단어들의 수가 평균적으로 7371개가 나온 것을 확인할 수 있었다.

따라서 문서를 통틀어 1번 나온 단어와 더불어 90%문서에서 공통적으로 나온 단어는 문서의 특징을 확인하는 것에 도움을 주는 것 보다 행렬의 차원을 크게 하여 정보를 분산시키는 효과가 더 크다고 판단하여 해당 단어들을 제거하여 해당 단어를 사후 불용어 처리를 하였으며, 각 형태소 분류기 별 TF-IDF 값을 확인하였다. 해당 문장을 모두 삭제한 결과 약 70% 단어가 삭제됨을 확인할 수 있었는데, 이러한 결과는 Sparse한 행렬의 밀집 정도를 높여 TF-IDF의 분산 정도가 낮아질 수 있음을 기대할 수 있었다.

이 과정에서 불용어 처리에 대해 해당 단어가 무엇인지 판단하고 정의된 불용어 목록 안에 추가하는 과정이 아닌, 각 형태소 분석기에서 나온 TF-IDF 행렬안에서 단어들의 빈도를 분석하여 그 단어를 제거하는 형식으로 진행하였다. 이렇게 진행한 이유는 따로 불용어 목록을 추가적으로 정의하고 다시 행렬을 새로 생성하는 작업은 생략해도 되는 작업이기 때문이다.

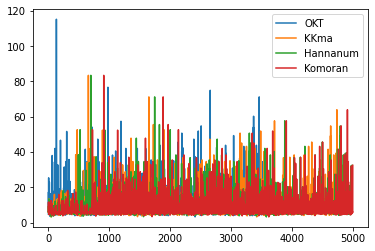
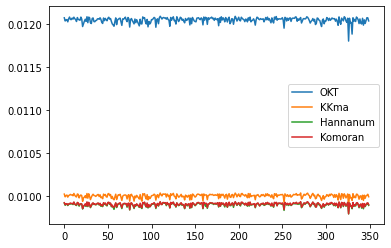
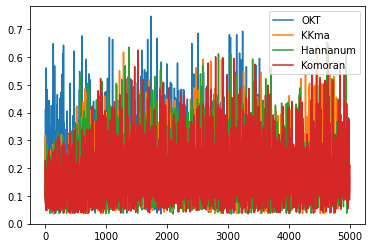
이러한 과정을 거쳐 사후 불용어 처리가 된 TF-IDF에서 각 단어 별 가장 높은 TF-IDF값을 가진 문서에서의 값에 대해 비교를 해 본 결과 아래의 그림처럼 불용어 처리 전 단계에서 진행했던 것 보다 Scale이 작아지게 되는 것을 확인할 수 있었으며 특히 Komo-ran의 경우 최댓값의 분포에는 큰 영향을 받는 것으로 나타났다. 이 결과에서 알 수 있는 것은 사후 불용어 처리를 거친 후 특정 문서에서 이상치 값을 갖는 단어들이 제거되었음을 알 수 있고 이러한 결과는 특정 TF-IDF 값에 특정 문서가 매몰되는 일을 방지하여 행렬 표현의 다양성을 부여할 수 있을 것이라 기대된다. 

Fig9. 사후 불용어 처리를 거친 후 각 단어 별 TF-IDF의 최댓값

추가적으로 각 문서에서 구성하는 열의 차원이 대폭 감소하였으므로 정규화의 영향이 더 클 것으로 판단하였고 열 삭제 후 정규화를 거쳐 각 문서에 대한 TF-IDF 값의 추이에 대해 살펴본 결과

Fig10. 사후 불용어 처리를 거친 후 L2 정규화를 진행하여 추출한 각 문서 별 표준 편차Fig11. 사후 불용어 처리를 거친 후 L2 정규화를 진행하여 추출한 각 단어 별 최댓값

해당 그림에서 확인한 결과 OKT의 경우 사후 불용어 추출 후 분산이 가장 높았으며 각 단어에 대해서 최댓값을 나타낸 결과 모든 형태소 분류기에 대해 Scale이 맞춰져 고르게 분포하는 것을 확인할 수 있다.

## 결론

각 분류기에 대해 전체 문서 대비 90% 이상에서 출현한 단어들과 단 하나의 문서해서 한번 출현한 단어들을 사후 불용어 처리한 후 기본적인 처리만 거친 TF-IDF 행렬과 비교한 결과 각 형태소 분류기에 대해 Scale이 맞춰줘 분포가 비슷하게 나타났다는 결과가 나왔으며, 특히 각 단어에 대한 최댓값 분포에서는 OKT 형태소 분류기를 제외한 나머지 분류기에서 형태의 변화가 있음을 알 수 있다.

선행 연구조사에서 알 수 있듯, 일반적으로 불용어 처리를 적절하게 수행한다면, TF-IDF를 통한 행렬의 혼잡도는 감소하는 것으로 알려져 있다. (Gil, 2018) 따라서 불용어 처리를 적절히 수행하는 것이 중요한데, 사전에 텍스트를 확인하는 정제 과정에서 불용어 처리를 하는 것도 방법이지만, 우선 TF-IDF 행렬을 만든 후 해당 행렬을 분석한 다음 이후 악영향을 주는 단어집합을 제거하는 식의 불용어 처리 방법도 한 가지 방안이 될 수 있다는 것을 여러가지 실험을 통해 살펴보았다.

이러한 결과와 더불어 TF-IDF안에서의 다중 공선성의 탐색이나, 공분산성 탐색, Convergence 요소 분석을 통해 TF-IDF 행렬 안에서도 따로 정제할 수 있는 방안들을 고안하여 대량의 문서에서 문서/단어 행렬을 생성할 때 Sparse해지는 행렬을 보다 밀집된 행렬로 바꾸는 실험을 할 수 있으며, 이러한 행렬을 통한 모델을 직접 실험하여 해당 모델이 TF-IDF의 변화에 따라 혼잡도의 차이는 어떻게 변하는지 알 수 있을 것이다.

’

[1] 김지숙. "빅데이터 활용과 분석기법 고찰." 국내석사학위논문 高麗大學校 大學院, 2013. 서울

[2] 길호현. (2018). 텍스트마이닝을 위한 한국어 불용어 목록 연구. 우리말글, 78, 1-25.

[3] 감미아 송민(2012), 텍스트마이닝을 활용한 신문사에 따른 내용 및 논조 차이점 분 석, 지능 정보 연구 18권 3호, 한국지능정보시스템학회, 53~77쪽.

[4] 주길홍 and 이원석. (2008). 효율적인 문서검색을 위한 레벨별 불용어 제거에 기반한 문서 클러스터링. 컴퓨터교육학회 논문지, 11(3), 67-80.

[5] 이원조. (2021). 텍스트 데이터 워드클라우드 분석을 위한 데이터 정제기법에 관한 연구. 문화기술의 융합, 7(4), 745-750.

[6] 김판구 조유근(1993), 한국어 정보 검색을 위한 불용어의 구성 및 적용, [한국 정보 과학회], 20권 1호, 한국정보과학회

[7] 장민승. "TF-IDF를 利用한 MLP 基盤의 文書 分類에 關한 硏究." 국내석사학위논문 성균관대학교 정보통신대학원, 2020. 서울

[8] 권호경. "자동색인을 위한 가중치 부여 불용어 사전의 구축." 국내석사학위논문 한양대학교 산업대학원, 1996. 서울