學 士 學 位 論 文

분산 처리 환경에서의 기계학습 기반의 뉴스 기사 빅 데이터 분석

충남대학교

공과대학 컴퓨터공학과

오 희 빈
이 정 철

지도교수 김 경 섭

2018 年 2 月

분산 처리 환경에서의 기계학습 기반의 뉴스 기사 빅 데이터 분석

지도교수 김 경 섭

이 논문을 공학사학위
청구논문으로 제출함

2017 年 12 月

충 남 대 학 교

공과대학 컴퓨터공학과

201202156 오 희 빈
201202168 이 정 철

목 차

그림 목차

수식 목차

1. 서 론

1.1 연구의 목적 및 필요성

정보통신 기술의 발전으로 인해 인터넷에는 매우 많은 양의 데이터들이 쏟아져 나온다. 하루에 인터넷에 게재되는 인터넷 기사의 수는 셀 수가 없을 정도이다. 정보 과도로 인하여 구독자가 얻고자 하는 정보를 정확히 찾는 것은 큰 노력이 필요하고 세간의 전체적인 동향을 살피는 것 또한 쉽지 않은 일이다. 본 논문에서는 이러한 구독자의 불편한 사항들을 해결하기 위해서 '분산 처리 환경에서의 기계학습 기반의 뉴스 기사 빅 데이터 분석' 연구를 통해서 사용자가 검색하고자 하는 키워드와 연관된 다른 키워드를 찾아 시각화하여 인터넷 뉴스의 전체적인 동향을 한눈에 파악하기 쉽게 하는 방안을 모색한다.

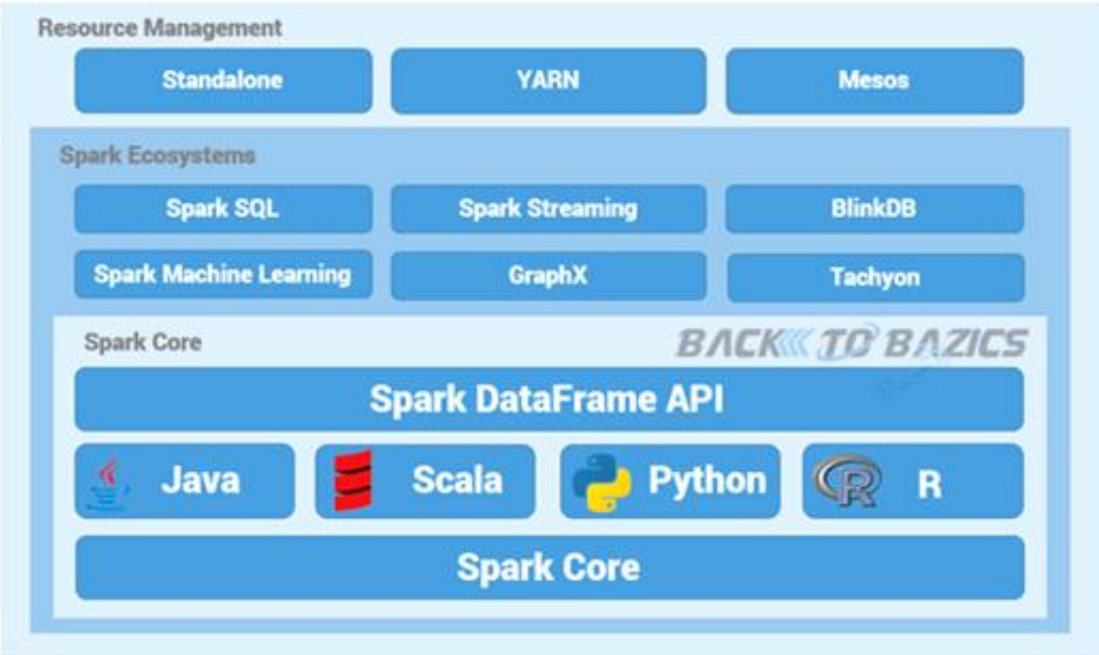
1.2 연구의 내용 및 범위

많은 양의 인터넷 뉴스 기사들을 기계학습 시키기 위해서 Hadoop과 Spark를 이용한 분산 처리 시스템을 구축하고 뉴스 기사 빅 데이터를 분산 처리 시스템을 이용하여 분석을 진행한다. 분석을 통해 얻어낸 데이터를 사용자가 검색한 키워드와 연관 키워드들로 시각화하여 한눈에 보기 쉽게 하고 인터넷 뉴스 기사들의 전체적인 동향을 파악할 수 있도록 하는 것이 논문에서 다루는 시스템의 목적이다. 논문의 다음과 같은 시나리오로 진행된다. 뉴스 기사를 크롤링하여 기사의 제목, 내용을 시스템에 저장하고 기사의 제목, URL, 날짜를 데이터베이스에 저장한다. 기사의 내용은 Konlpy를 이용하여 명사화하고 명사화한 기사 내용을 TF-IDF를 이용하여 불필요하지만, 많이 쓰이는 단어를 걸러낸다. 가공을 거친 데이터를 Word2Vec에서 사용할 수 있는 txt 파일 형태로 만들어준다. 마지막으로 분산시스템인 Spark의 Mllib인 Word2Vec을 통해 키워드와 연관 키워드의 관계를 분석하고 생성된 Model을 이용하여 시각화하여 사용자에게 보여준다. 자세한 내용과 과정은 논문의 뒷부분에서 다룬다.

2. 관련 연구

2.1 Apache Spark

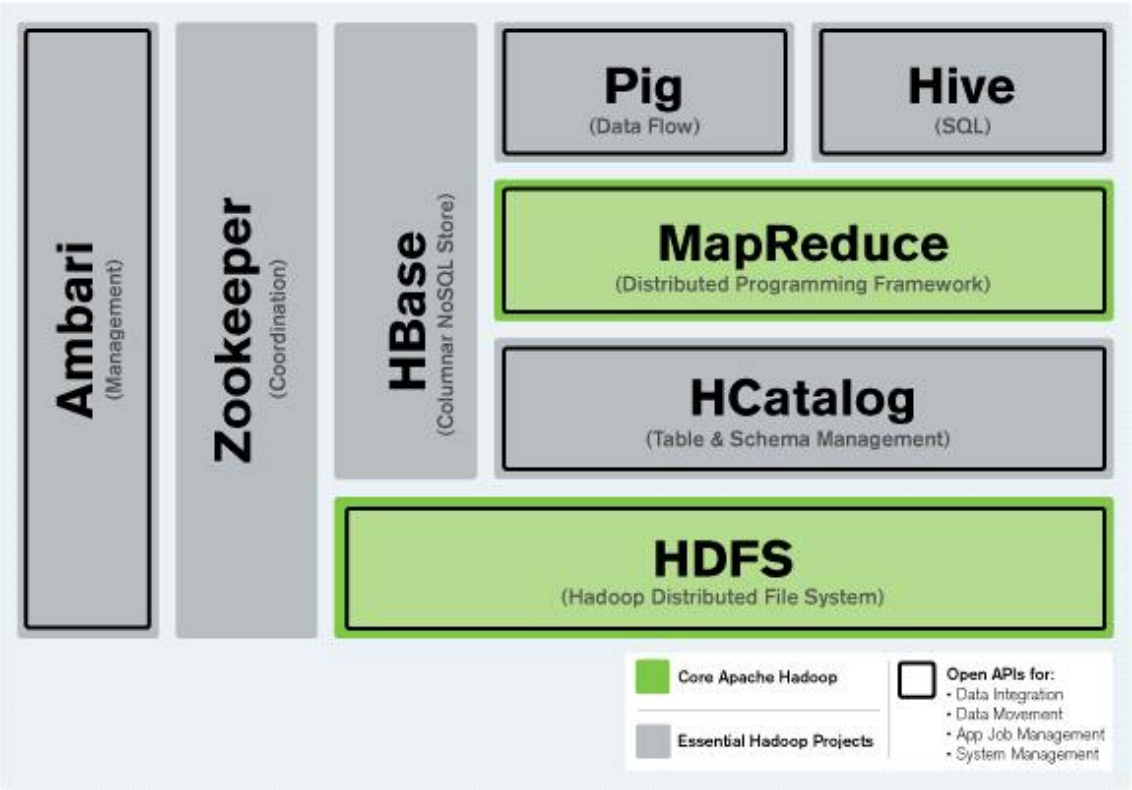
Apache Spark[1]는 그림 1과 같은 구조로 메모리상에서 동작하는 확장성이 뛰어난 분산 시스템으로, Java, Scala, Python, R과 같은 여러 언어를 사용하여 Application을 개발할 수 있는 기능을 제공한다. Mahout과 같은 Apache 시스템은 이제 MapReduce를 대신하여 프로세싱 엔진으로 자리매김하고 있으며, Spark Application은 Hive를 사용할 수 있으므로 Hive와 직접 데이터를 주고받을 수 있다.



[그림 1: spark 구조]

2.2 Apache Hadoop

아파치 하둡(Apache Hadoop)은 대량의 자료를 처리할 수 있는 큰 컴퓨터 클러스터에서 동작하는 분산 응용 프로그램을 지원하는 프리웨어 자바 소프트웨어 프레임워크이다. 원래 너치의 분산 처리를 지원하기 위해 개발된 것으로, 아파치 루씬의 하부 프로젝트이다. 분산처리 시스템인 구글 파일 시스템을 대체할 수 있는 하둡 분산 파일 시스템(HDFS: Hadoop Distributed File System)과 맵리듀스를 구현한 것이다.

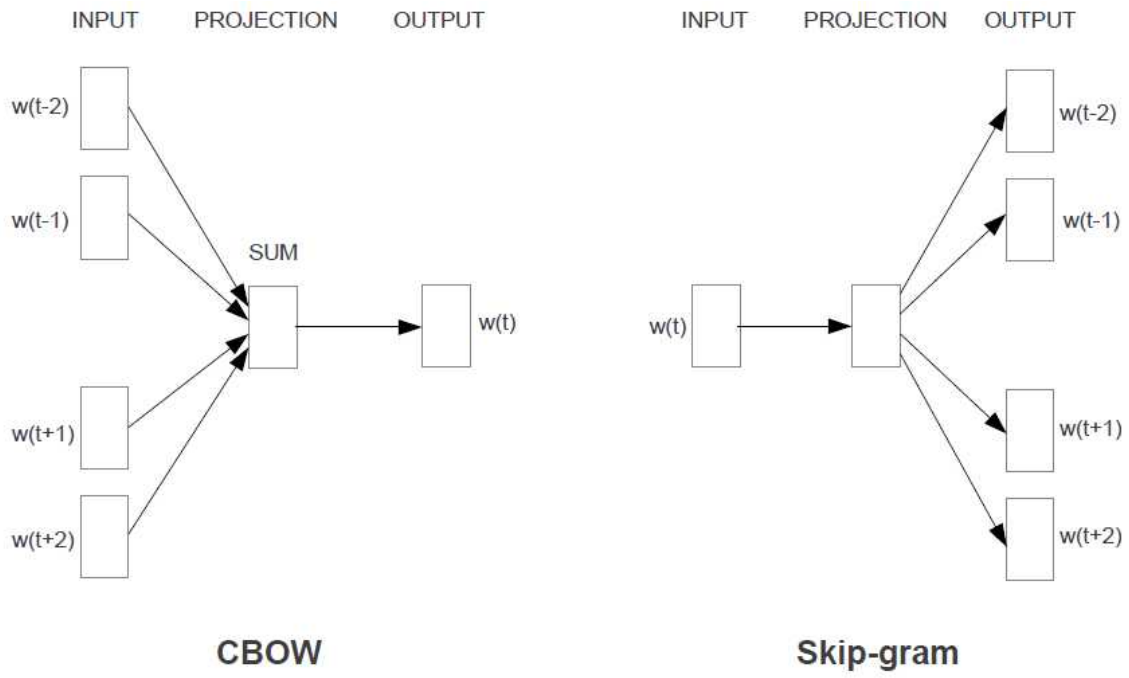


[그림 2: 하둡 프레임워크]

2.3 Word2Vec

구글에서 만들어진 Word2Vec[2]은 word embedding과 관련된 학습 모델이며, 그림 3와 같이 Word2Vec 모델은 신경망 구조를 가지며, Deep Learning이 아닌 Shallow, two layer 구조로 되어 있다. 이는 언어적인 단어 문단을 재구성하고 학습하는데 용이하다. input으로 큰 텍스트를 받아 수백 차원의 벡터 공간을 만들어 낸다. 만들어진 벡터 공간에는 단어에 따른 기준 단어와 연관된 단어들이 거리에 따라 분류된다.

Word2Vec은 CBOW(Continuous Bag-of-Words) 모델과 Skip-gram 모델 두 가지가 있다. CBOW는 주어진 단어에 대해 앞 뒤로 C/2개씩 총 C개의 단어를 Input으로 사용하여, 주어진 단어를 맞추기 위한 네트워크를 만드는 방식이다. Skip-gram 모델은 현재 주어진 단어 하나를 가지고 주위에 등장하는 나머지 몇 가지의 단어들의 등장 여부를 유추하는 것이다.
본 논문에서 사용한 PySpark MLlib(Machine Learning Library)의 Word2Vec은 Skip-gram 모델을 사용하였다.



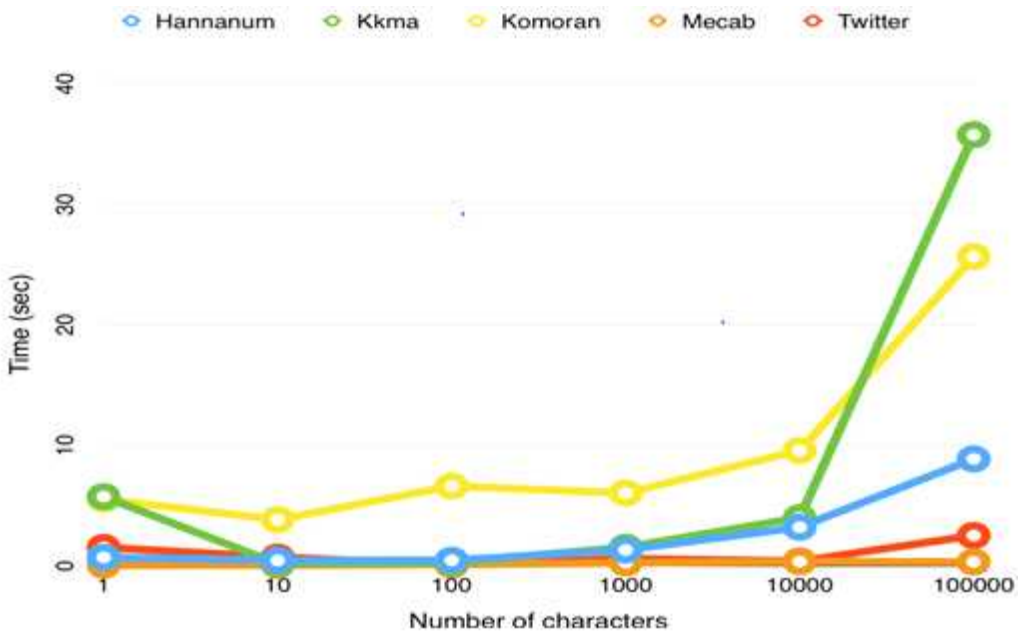
[그림 3: Word2Vec 구조]

2.4 Web Crawler

웹 크롤러(web crawler)는 조직적, 자동화된 방법으로 월드 와이드 웹을 탐색하는 컴퓨터 프로그램이다. 검색 엔진과 같은 여러 사이트에서는 데이터의 최신 상태 유지를 위해 웹 크롤링한다. 웹 크롤러는 대체로 방문한 사이트의 모든 페이지의 복사본을 생성하는 데 사용되며, 검색 엔진은 이렇게 생성된 페이지를 보다 빠른 검색을 위해 인덱싱한다. 또한 크롤러는 링크 체크나 HTML 코드 검증과 같은 웹 사이트의 자동 유지 관리 작업을 위해 사용되기도 하며, 자동 이메일 수집과 같은 웹 페이지의 특정 형태의 정보를 수집하는 데도 사용된다.

2.5 Konlpy

Konlpy[3]는 한국어 정보처리를 위한 Python 패키지이며, 한국어 문장을 입력받아 이를 원하는 형태로 가공할 수 있다. 분석 방법에 따라 Kkma, Komoran, Hannanum, Twitter, Mecab 으로 클래스가 나누어져 있다. 명사 추출, 형태소 분리 등 여러 가지 기능들을 이용하여 한국어 문장의 형태를 가공할 수 있다. 문장의 개수에 따른 클래스의 시간은 그림 4와 같다.



[그림 4 : Konlpy 실행 시간]

2.6 TF-IDF

TF-IDF[4]는 정보 검색과 텍스트 마이닝에서 이용하는 가중치로, TF는 특정한 단어가 문서 내에 얼마나 자주 등장하는지를 나타내는 값이며, IDF는 특정한 단어가 몇 개의 문서 안에서 쓰였는지에 대한 DF의 역수이다. TF-IDF는 TF와 IDF를 곱한 값이다. 여러 문서로 이루어진 문서 군이 있을 때 어떤 단어가 특정 문서 내에서 얼마나 중요한 것인지를 나타내는 통계적 수치이다. 문서의 핵심어를 추출하거나, 검색 엔진에서 검색 결과의 순위를 결정하거나, 문서들 사이의 비슷한 정도를 구하는 등의 용도로 사용할 수 있다.

$$\mathrm{tf}(t,d) = 0.5 + \frac{0.5 \times \mathrm{f}(t,d)}{\max\{\mathrm{f}(w,d) : w \in d\}}$$

[수식 1: TF]

$$\mathrm{idf}(t,D) = \log \frac{|D|}{|\{d \in D : t \in d\}|}$$

[수식 2: IDF]

$$\mathrm{tfidf}(t,d,D) = \mathrm{tf}(t,d) \times \mathrm{idf}(t,D)$$

[수식 3: TF-IDF]

2.7 Machine Learning

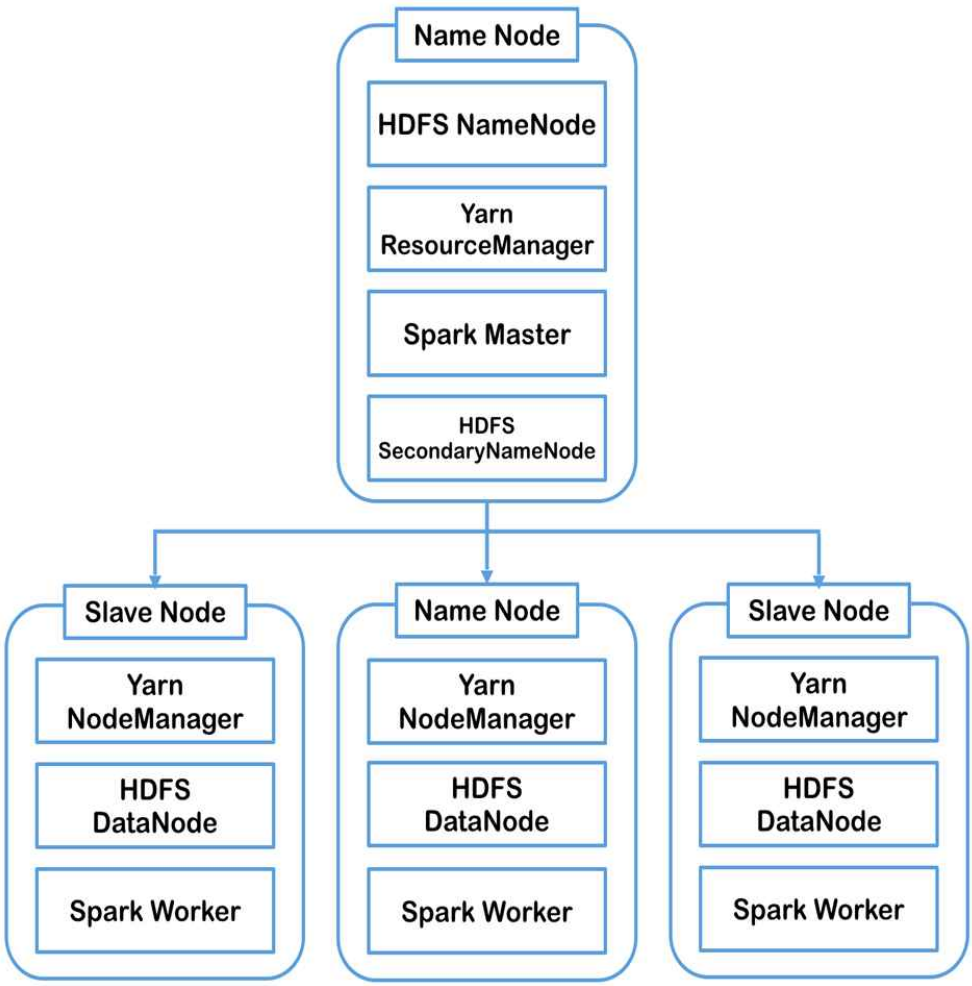
인공 지능의 한 분야로, 컴퓨터가 학습할 수 있도록 하는 알고리즘과 기술을 개발하는 분야를 말한다. 가령, 기계 학습을 통해서 수신한 이메일이 스팸인지 아닌지를 구분할 수 있도록 훈련할 수 있다.

기계 학습의 핵심은 표현(representation)과 일반화(generalization)에 있다. 표현이란 데이터의 평가이며, 일반화란 아직 알 수 없는 데이터에 대한 처리이다. 이는 전산 학습 이론 분야이기도 하다. 다양한 기계 학습의 응용이 존재한다. 문자 인식은 이를 이용한 가장 잘 알려진 사례이다.

3. 시스템 설계 및 구현

3.1 분산 시스템 구축

대량의 인터넷 뉴스 기사를 분석하고 학습하기 위해서 분산처리 환경을 Hadoop과 Spark를 이용하여 구축했다. 대량의 인터넷 뉴스 기사들을 처리하기 위해서 HDFS를 기반으로 YARN을 통해 자원 관리와 스케줄링을 실행했다. 이를 위해 그림 5와 같이 분산 시스템을 1대의 NameNode와 그 NameNode를 포함한 3대의 DataNode를 설정하고 Master Node에는 추가로 Yarn ResourceManager, SecondaryNameNode, Spark Master 그리고, Yarn NodeManager를 설정해 준다. 그리고 2대의 Slave Node에는 추가로 Spark Slave, Yarn NodeManager를 설정해서 분산 시스템 환경을 구축했다.



[그림 5: 분산 시스템 구성]

3.2 시나리오

3.2.1 전체 시나리오

본 논문에서 구현된 시스템의 전체 시나리오는 그림 5와 같이 이루어져 있다.

(1) bs4 라이브러리를 이용하여 네이버 뉴스 기사를 크롤링한다.

(2) 데이터 중 뉴스 제목, URL, 날짜를 뉴스 기사 데이터베이스에 저장한다.

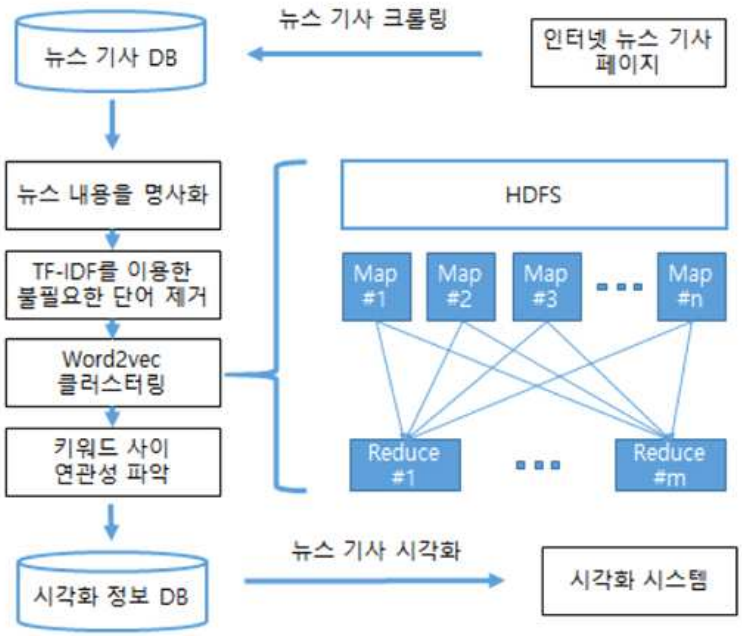
(3) 크롤링한 데이터 중 뉴스 기사의 내용을 Konlpy 라이브러리 중 twitter를 이용하여 명사화하며 날짜별로 기사를 저장한다.

(4) 저장한 뉴스 기사 내용을 TF-IDF 알고리즘을 이용해 기사에 중요도가 낮고 불필요한 명사들을 제거하고 Word2Vec에 이용할 수 있게 하나의 txt 파일로 저장한다.

(5) 생성된 txt 파일을 Spark 분산 시스템을 이용해서 Pyspark의 Mllib인 Word2Vec 알고리즘을 통해 각각의 키워드와 연관된 키워드의 Model을 만들어 낸다.

(6) 생성된 Model에 키워드 간의 연관도를 시각화 데이터베이스에 저장한다.

(7) 시각화 정보 데이터베이스에 저장된 내용을 시각화 시스템을 통해 사용자가 이용할 수 있게 만들어준다.



[그림 6: 시나리오 구성도]

3.2.2 Crawling

Python의 bs4 라이브러리를 이용하여 코드를 구현했으며, 네이버 뉴스 웹페이지에서 뉴스 기사를 수집하였다.

뉴스기사를 Crawling하는 과정은 네이버 뉴스 웹페이지의 속보:정치 부분의 URL을 확인하여 URL의 날짜를 입력하는 부분을 파라미터로 받게 한다. 그 후에 파라미터로 날짜를 입력하면 네이버 뉴스 웹페이지의 첫 번째 페이지부터 마지막 페이지까지 순서대로 URL을 HTTP request를 통해 기사 원문을 서버로부터 제공 받으며 그림 7과 같이 인터넷 뉴스 기사의 URL, 뉴스 기사 제목, 기사 기재 날짜를 News 데이터베이스에 저장한다.

저장된 데이터를 이용하여 시각화 중 키워드에 존재하는 뉴스 기사를 뉴스 기사 제목을 이용하여 보여주게 한다.

Num	URL	뉴스 기사 제목	기사 기재 날짜
1	http://news.naver.com/main/read.nhn?...	트럼프, DMZ 방문 안 한다 ...	20171101
2	http://news.naver.com/main/read.nhn?...	다음 주 한중 정상회담 "모든..	20171101
3	http://news.naver.com/main/read.nhn?...	제주시 국수문화거리 일부 구..	20171101
4	http://news.naver.com/main/read.nhn?...	국민신문고, "국민이 선택한...	20171101

[그림 7: DB에 저장된 데이터]

3.2.3 Konlpy

본 논문에서는 한글로 되어있는 인터넷 뉴스 기사의 명사를 추출하여 하나의 명사와 연관된 명사들을 Word2Vec을 이용하여 데이터를 얻어내기 위해 Konlpy 라이브러리를 이용하였다.

Python의 Konlpy 라이브러리의 Kkma, Komoran, Twitter, Mecab, Hannanum 클래스 중 많은 양의 문장의 명사를 추출하기 때문에 문장의 개수가 많아져도 시간이 오래 걸리지 않는 twitter를 이용하여 한글 문장 명사를 그림 8과 같이 추출하여 txt 파일로 저장한다.

기존 뉴스 기사 내용	명사 추출 후 뉴스 기사 내용
홍준표 자유한국당 대표가 1일 박근혜 전 대통령에 대한 재명안 처리와 관련해 "최고위원회의 연가는 없다"고 밝혔다. 홍 대표는 이날 서울 여의도의 한 식당에서 당내 초선 의원들과 만찬 회동을 한 뒤 기자들과 만난 자리에서....	홍준표 자유 한국 대표 일 박근혜 전 대통령 대한 제명 안 처리 최고 위원회 연기 고 홍 대표 날 서울 여의도 식당 당내 초선 위원 만찬 회동 뒤 기사 만난 자리 일 최고 위원 회의 예정 개최 기자 질문 홍 대표 당내 시간 생각 순리 처리 박 전 대통령 .....

[그림 8: Konlpy를 이용한 명사 추출]

3.2.4 TF-IDF

HTML 형식의 뉴스 기사들을 Crawling하여 저장하였기 때문에 기사의 내용 이외의 불필요한 명사들이 파일에 포함되어 있다. 불필요한 명사가 포함된 상태로 Word2Vec을 수행하면 제대로 된 결과값을 얻어낼 수 없다. 이와 같은 문제를 해결하기 위해 TF-IDF를 구해내는 알고리즘을 Python 코드로 구현하였다. 1글자인 내용은 TF-IDF와 Word2Vec의 정확도를 떨어트리기 때문에 2글자 이상인 명사만을 사용했다.

전체 문서에서 명사가 등장하는 횟수와 각각의 문서에 명사가 등장하는 횟수를 이용하여 TF와 IDF를 얻어내어 TF-IDF 값을 계산해낸다. 중요도가 떨어지는 명사들은 TF-IDF가 낮은 값을 갖는다. 기준값을 1.9로 설정하여서 그림 9과 같이 TF-IDF값이 기준값보다 낮은 명사들을 제거해 주었다.

불필요한 명사들을 제거한 후 문서들을 Word2Vec에 이용하기 위해 하나의 txt 파일에 합쳐서 저장해 주었다.

명사	모든 뉴스에서 사용된 키워드 수	키워드가 쓰인 뉴스 수	TF	IDF	TF-IDF
배포	4442	4410	3.6476763	0.304218842	1.10969186
본문	17753	8877	4.2492962	0.000391212	0.00166237
제보	5418	2821	3.7339191	0.498254345	1.86044144
클릭	3101	3078	3.4916417	0.460388881	1.60751283
…	……	…….	…….	……	…..

[그림 9: TF-IDF를 이용해 제거할 단어 추출]

3.2.5 Word2Vec

대량의 기사 내용을 이용해서 뉴스 기사들의 키워드와 연관된 키워드를 찾기 위해 분산 시스템인 Spark의 Mllib인 Word2Vec 라이브러리를 이용하며 언어로는 Pyspark를 선택하여 구현하였다.

하나로 저장된 txt 파일을 Hadoop의 HDFS에 저장시킨 후 분산 시스템에 이용하게 만든다. Pyspark로 구현한 Word2Vec을 수행시켜 하나의 명사와 그 명사와 연관된 명사들의 데이터가 저장된 Model을 생성한다. 생성된 Model을 이용하여 명사 하나당 5개의 연관된 명사들을 연관된 순서에 따라 그림 10와 같이 데이터베이스에 저장한다.

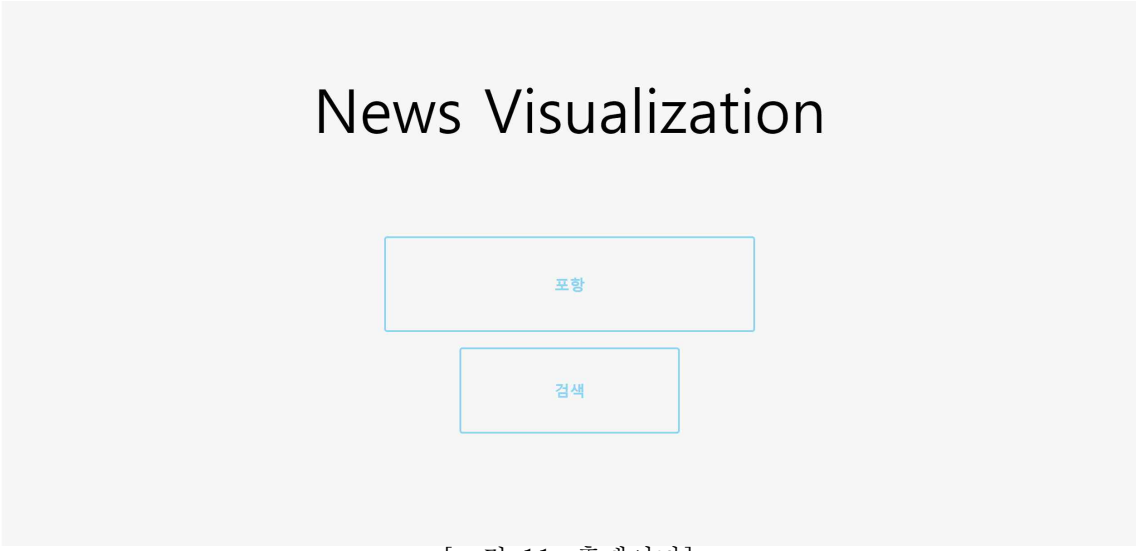
데이터베이스에 저장된 데이터는 시각화하는데 사용한다.

Noun	Synonyms_1	Synonyms_2	Synonyms_3	Synonyms_4	Synonyms_5
피해자	일본군	생존자	위안부	장례식	상숙
피해	몰카	방지	규제	유통	재발
미국	방한	중국	로이스	외교	한미정책
미사일	탄도미사일	로케트	북한	도발	탄도
…	……	…….	…….	……	…..

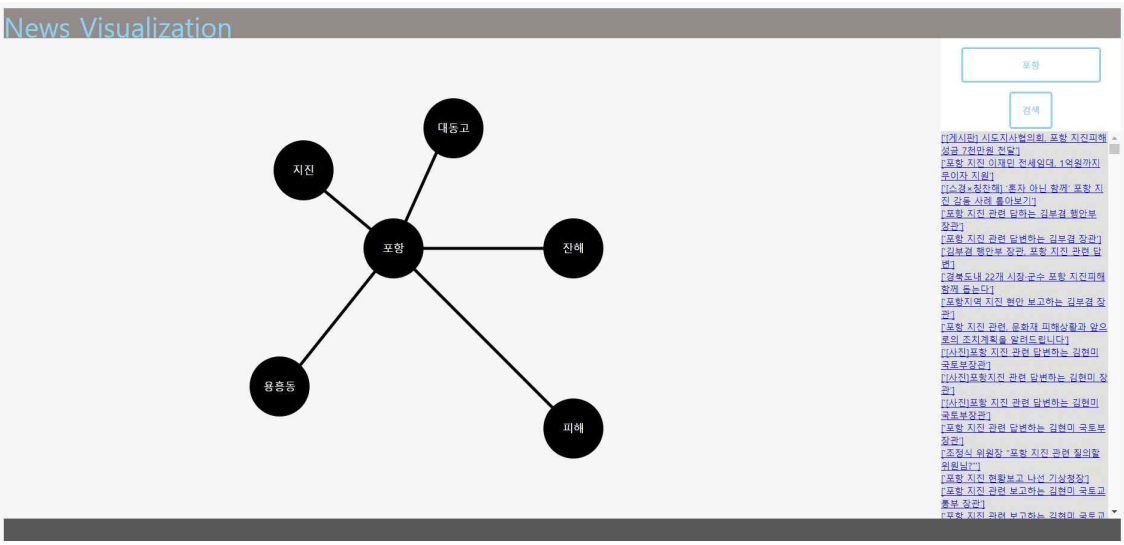
[그림 10: Word2Vec을 연관 키워드 추출]

3.2.6 시각화

사용자가 시각화 시스템을 이용하는 방법은 그림 11과 같은 홈페이지에서 사용자가 검색 키워드를 입력하고 검색 버튼을 클릭한다. 검색 키워드가 데이터베이스에 존재하지 않을 경우에는 다시 그림 11과 같은 페이지로 이동하게 되고 검색 키워드가 데이터베이스에 존재하면 그림12와 같은 페이지에 검색 키워드가 전해진다.



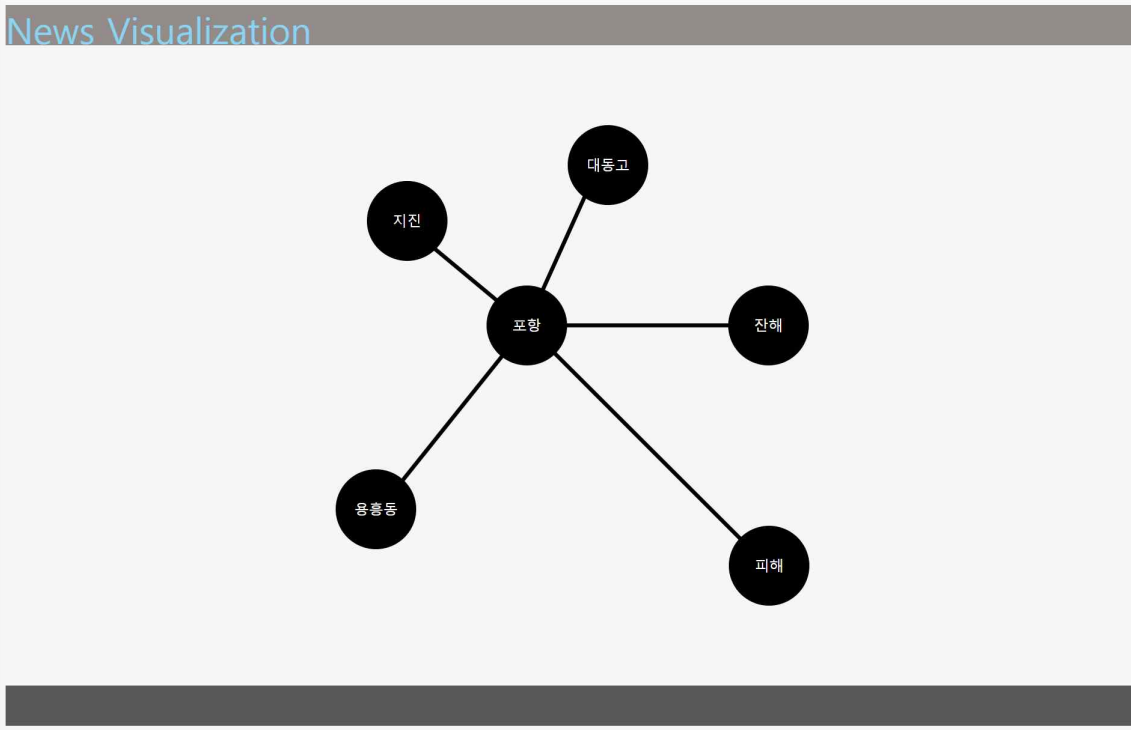
[그림 11: 홈페이지]



[그림 12: 메인 페이지]

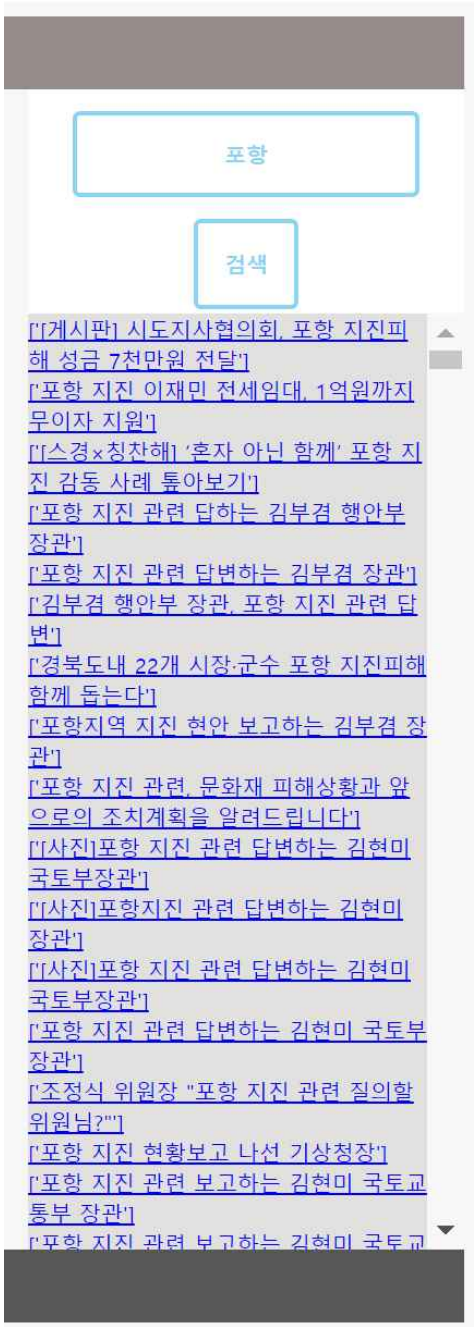
그림 13은 데이터베이스에 저장된 검색 키워드와 연관 키워드 5개를 보여주며, 검색 키워드를 중심으로 5개의 키워드가 배치된다. 5개의 키워드는 검색 키워드가 갖는 연관도가 가장 높은 키워드들이고 각자 키워드들은 자신의 뉴스기사를 가지고 있다. 시각화된 키워드를 클릭하면 클릭한 키워드와 연관된 뉴스 기사의 제목을 URL형식으로 그림 14와 같이 나열한다. 선의 길이는 연관도에 따라서 달라지며 연관도가 높을수록 선의 길이가 짧아진다.

좌측 상단의 배너를 클릭하면 그림 11의 홈페이지 화면으로 이동한다.



[그림 13: 키워드 시각화]

그림 14의 하단은 데이터베이스에 저장된 뉴스 기사의 제목에 키워드가 존재한다면 관련된 뉴스 기사로 정하고 뉴스 기사의 제목을 URL 형식으로 나열한다. 그림 14의 상단은 메인 페이지에서 다시 검색하고 싶은 키워드를 입력함으로써 입력한 키워드를 중심으로 다시 시각화 한다.



[그림 14: 뉴스기사 목록]

4. 결 론

본 논문에서는 매시간 발생하는 대량의 인터넷 뉴스, 즉 뉴스 기사 빅데이터에서 기계학습을 통하여 유의미한 데이터를 추출하는 방법을 다루었다. 또한, 빅데이터의 기계학습을 효율적으로 진행하기 위해 분산처리 환경을 사용하였다. 분산처리 환경에 사용된 PC 수가 그리 많지 않아서 연산에 필요한 시간을 대폭 줄이진 못했지만 많은 성능 향상이 있었다. 분산처리 환경은 확장이 용이하므로 후에 더 많은 PC를 이용한다면 연산처리 시간을 현저히 낮출 수 있을 것으로 보인다.

본 논문에서는 뉴스 기사에 대해서만 분석을 진행하였지만 뉴스 기사뿐만 아니라 텍스트 형태의 모든 자료를 분석할 수 있다. 논문, 특허, 인터넷 게시판 등의 많은 분야에 검색하고자 하는 키워드가 갖는 연관 키워드를 시각화된 형태로 확인할 수 있을 것이다.

참고문헌

[1] Framton, Mike [Mastering Apache spark: 정보문화사] , 2015

[2] Yoav Goldberg. Omer Levy [word2Vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method], 2014

[3] 박은정, 조성준, "KoNLPy: 쉽고 간결한 한국어 정보처리 파이썬 패키지", 제26회 한글 및 한국어 정보처리 학술대회 논문집, 2014

[4] https://ko.wikipedia.org/wiki/TF-IDF