JDCS
디지털콘텐츠학회논문지
Journal of Digital Contents Society
Vol. 24, No. 1, pp. 79-89, Jan. 2023

Check for updates

공간정보 추출을 위한 Storm 기반 실시간 분석 시스템 설계 및 개발

전 원 표[1*] · 윤 효 근[2] · 안 창 원[3]
[1]*바이브컴퍼니 스마트시티연구소 책임연구원
[2]바이브컴퍼니 스마트시티연구소 수석연구원
[3]바이브컴퍼니 스마트시티연구소 연구소장

Storm-based Real-Time Analysis System Design and Development for Spatial Information Extraction

Won-Pyo Jeon[1*] · Hyo-Gun Yoon[2] · Chang-Won Ahn[3]

[1]*Senior Researcher, Smart City Institute, VAIV company, Seoul 04419, Korea
[2]Chief Researcher, Smart City Institute, VAIV company, Seoul 04419, Korea
[3]Director, Smart City Institute, VAIV company, Seoul 04419, Korea

[요 약]

최근 성숙기에 접어든 소셜 미디어 시장 성장률이 코로나 19를 기점으로 다시금 상승 추세로 돌아섰다. 그 수치는 2022년 1월 기준 45억 2천만 명이나 되고, 이에 따라 생산되는 데이터의 양도 급속도로 증가하고 이를 분석하기 위한 기술도 발전하고 있다. 이러한 소셜 미디어 데이터를 다양한 관점에서 분석하고 활용하기 위해서는 기존의 공간 정형데이터와 융합 분석이 필요하다. 소셜 미디어의 특성상 데이터를 실시간으로 분석할 수 있어야하고, 융합 분석을 위해 공간 정보 추출 할 수 있어야 한다. 이에 따라, 본 논문에서는 실시간 분석 프레임워크인 스톰을 활용하여 공간 정보를 실시간으로 추출하는 시스템을 제안한다. 실험 결과, 15개의 익스큐터를 사용할 경우 377.59ms, 25.03ms의 지연시간이 발생하였으나, 34개의 익스큐터를 사용한 결과 39.54ms, 0.8ms로 338.05ms, 24.23ms의 지연시간이 감소됨을 확인하였다.

[Abstract]

The growth rate of the social media market, which has recently entered a mature stage, has turned to an upward trend again with the outbreak of COVID-19. The number is 4.52 billion as of January 2022, and the amount of data produced is rapidly increasing accordingly, and the technology for analyzing it is also developing. In order to analyze and utilize these social media data from various perspectives, existing spatial structured data and convergence analysis are required. Due to the nature of social media, data must be analyzed in real time and spatial information must be extracted for convergence analysis. Therefore, in this paper, we propose a system that extracts spatial information in realtime using Storm, a real-time analysis framework. As a result of the experiment, when using 15 executors, delay times of 377.59ms and 25.03ms occurred, but as a result of using 34 executors, it was confirmed that the delay times of 338.05ms and 24.23ms were reduced to 39.54ms and 0.8ms.

색인어 : 실시간 분석, 공간 정보 추출, 텍스트 마이닝, 개체명 인식, 아파치 스톰

Keyword : Real-time analysis, Spacial information extraction, Text mining, Named entity recognition, Apache storm

http://dx.doi.org/10.9728/dcs.2023.24.1.79

Received 28 November 2022; **Revised** 14 December 2022
Accepted 29 December 2022

***Corresponding Author; Won-Pyo Jeon**

Tel: +82-2-565-0531
E-mail: jwp@vaiv.kr

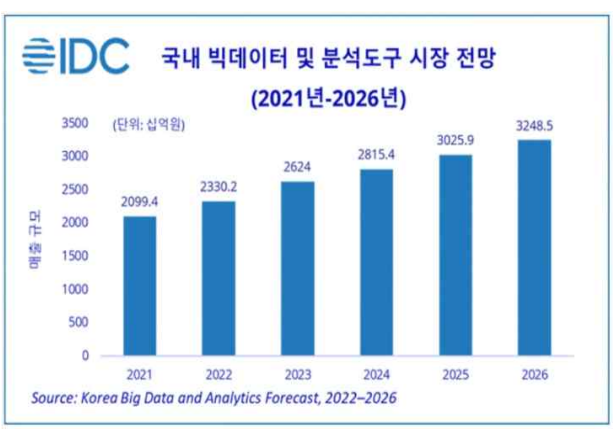
Ⅰ. 서 론

2010년 이후 전 세계적으로 소셜 미디어 이용자들은 매년 10% 이상의 성장세를 지속해왔다[1]. 소셜 미디어 관리 플랫폼 Hootsuite가 발표한 '디지털 2022년 1월 글로벌 현황' 보고서에 따르면 최근 성숙기에 접어든 소셜 미디어 시장 성장률이 코로나 19를 기점으로 다시금 상승 추세로 돌아섰고, 그 수치는 2022년 1월 기준, 전 세계 인구의 절반이 넘는 58.4%(45억 2천만 명)의 소셜 미디어 사용자가 있으며, 전년 동월 대비 10%이상(4억 2,400만 명) 성장했다고 발표했다[2]. 이에 따른 데이터의 양도 급속도로 증가하고 있으며, 데이터 처리 기술 발전에 따라 빅데이터 및 분석도구 시장 전망도 <그림 1>과 같이 함께 증가하고 있는 추세이다[3].

이와 같이 소셜 미디어에서 생산되는 대용량의 빅데이터를 분석하기 위해 하둡(Hadoop) 기반의 텍스트 분석 시스템에 대한 연구가 활발히 진행되었고, 산업계에도 다양한 서비스에 적용되어 운영되고 있다[4]-[8]. 하둡은 2003년 구글의 분산 파일 시스템 아키텍처에 관한 연구와 2004년 분산 처리 방식에 대한 연구를 기반으로 HDFS(Hadoop Distribute File System)와 MapReduce가 개발되며 시작되었다[9],[10]. 초기의 하둡 기반 플랫폼은 배치 처리의 강점을 가지고 있었으나 실시간 처리 등의 한계로 인하여 실시간으로 데이터와 사용자가 변하는 소셜 미디어의 특성을 반영하지 못한다는 단점이 있다. 이와 더불어 데이터의 단순한 분석만을 수행하기에 활용도가 떨어지는 문제점도 가지고 있다[11]. 이러한 문제점을 해결하기 위해 추가적인 에코 시스템들이 포함되고 지속적인 기능 개선이 일어났으며, 인메모리(In-Memory) 속도와 고성능 컴퓨팅을 위한 분산 데이터베이스 IMDB(In-Memory DataBase)와 스트리밍 데이터(Streaming Data) 처리를 위한 실시간 분산 처리 프레임워크 아파치 스톰(Apache Storm), 아파치 스파크(Apache Spark) 등이 등장 하였다. 그 중 스톰은 하둡의 맵리듀스 프레임워크를 사용하지 않고 분산 처리할 수 있는 환경을 제공한다. 클러스터링 기능을 이용하여 수평적 확장(Scale-Out)이 가능하며 장애 노드가 발생하더라도 Fault Tolerance가 가능하기 때문에 데이터 유실 없이 분석을 할 수 있는 환경을 제공한다. 또한 Java 기반의 JVM(Java Virtual Machine) 위에서 동작하지만 Thrift 프로토콜을 기반으로 하기 때문에 python, java, ruby, perl, C/C++ 등의 다양한 프로그래밍 언어를 지원하며, <그림 2>와 같이 데이터 입출력을 위한 다양한 플러그인을 지원하기 때문에 Kafka, RabbitMQ, Kinesis, Cassandra, Elasticsearch 등의 다양한 DB 및 시스템과 연계가 용이하다.

빅데이터를 다양한 관점에서 분석하고 활용하기 위한 다차원 분석 기술은 공학 뿐만 아니라 사회과학 등 다양한 분야에서 연구되고 있다. 주제도, 분포도 등 공간 빅데이터 시각화 관련 기술이 한 축을 이루고 있으며, 공간 빅데이터 등으로 분류되는 비정형 데이터를 기존의 공간 정형데이터와 융합 분석하는 융합 기술도 활발히 연구되고 있다[12]. 특히 사용자 위치 기반 맞춤형 서비스를 위해서는 비정형 데이터에서 공간 정보를 추출하는 연구가 필수적이다. 이는 GPS를 장착한 스마트폰의 이용과 이에 기반을 둔 정보 공유가 증대됨에 따라 개인에 의한 사용자 참여형 공간 정보(VGI, Volunteered Geographic Information)가 축적되면서 대안 자료로써 논의되고 있으나[13], 트위터의 경우 이러한 위치 태그를 가지는 자료는 전체의 1.5~3% 정도에 지나지 않으며[14], 그 외 다수의 데이터도 이와 마찬가지로 관심대상의 공간정보를 직접 확인하기 어렵다.

따라서 본 논문에서는 실시간으로 수집되는 데이터로부터 공간 정보를 자동으로 추출하여 분석할 수 있는 스톰 기반 실시간 분석 시스템을 제안한다. 본 논문의 구성은 다음과 같다. 2장에서 실시간 분석 시스템 및 공간 정보 추출에 대한 관련 연구에 대하여 알아보고, 3장에서 제안하는 실시간 공간 정보 추출 시스템에 대한 각 계층 별 설계 및 개발 방법을 설명한다. 4장에서는 제안 시스템을 기반으로 분석 테스트를 수행한 결과에 대하여 기술하고, 5장의 결론으로 마무리하였다.



*Impossible to change due to image copyright

그림 1. 국내 빅데이터 및 분석도구 시장 전망

Fig. 1. Korea Big Data and Analytics Forecast, 2022-2026

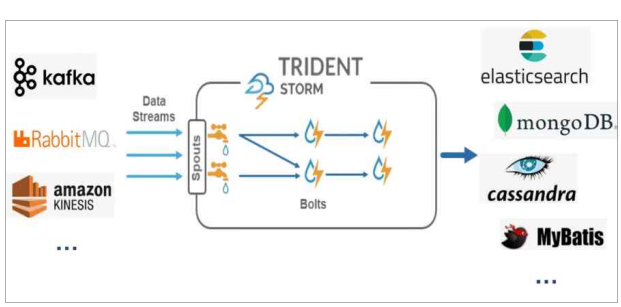


그림 2. 아파치 스톰 지원 시스템

Fig. 2. Apache Storm Support System

Ⅱ. 관련 연구

이번 장에서는 대용량의 빅데이터를 실시간으로 분석하기 위한 실시간 분석 프레임워크와 스톰을 활용한 서비스에 대해서 알아보고, 비정형 텍스트 데이터에서의 공간 정보의 정의와 공간 정보를 추출하기 위한 관련 연구 현황에 대하여 분석한다.

2-1 실시간 분석 프레임워크

하둡은 대용량의 데이터를 분석하기 위해 맵리듀스라는 병렬 처리 프레임워크를 이용한다. 하둡 프레임워크는 요구하는 사양이 상당히 낮으면서 데이터 신뢰성이 높아 대용량 데이터 처리에 최적화된 파일 시스템으로 평가 받아왔다. 하지만 실시간으로 축적되는 데이터의 양이 급증하고 있는 상황에서는 몇 가지 문제점을 가지고 있다. 많은 네트워크 트래픽과 디스크 I/O를 야기할 위험이 있고, 데이터 연산 또는 처리 작업이 산발적이고 비정기적으로 일어날 경우 비효율적이다[15]. 이러한 문제를 해결하기 위해 스트림 데이터를 일정 단위로 쪼개어 작은 배치 처리를 무한히 수행하는 방식의 프레임워크인 스파크가 등장하였다. 스파크는 실시간으로 전달되는 스트림 데이터를 배치 간격마다 나누고, 나누어진 데이터를 처리하는 마이크로 배치(Micro Batch) 형태로 기존 스파크에서 사용하는 RDD와 유사한 데이터 형태인 DStream으로 기존의 스파크의 명령어인 Transform Operations, Join Operations 등을 이용하여 데이터를 분석한다.

아파치 이그나이트(Apache Ignite)는 인메모리 컴퓨팅 프레임워크로 인메모리의 빠른 속도와 고성능 컴퓨팅을 위한 분산 데이터베이스이다. IMDB이지만 SQL과 Key-Value, 트랜잭션 인터페이스를 지원하기 때문에 DDL(Data Definition Language), DML(Data Manipulation Language)를 이용하여 데이터 핸들링이 가능하다. 하지만 제약 조건이나 인덱스를 처리하는 방식에 차이가 있으며, 외래키는 지원하지 않는다. 스톰과 마찬가지로 클러스터를 추가함에 따라 수평 확장이 가능한 장점이 있다. 또한 <그림 3>과 같이 사용자의 필요에 따라 persistence 기능을 사용함으로써 영구 저장소(Persistence Storage)나 인메모리 저장소(In-Memory Storage)로 사용할 수 있다. persistence 기능을 사용할 때는 디스크 중심 RDBMS와 같이 프로세싱은 메모리에서 수행하고 인덱스나 데이터 저장은 디스크 상에서 이루어지지만 일관되고 수평적인 확장이 가능하며 SQL, Key-Value 프로세싱 API를 지원한다.

아파치 스톰은 스트리밍 데이터 처리를 위한 실시간 분산 처리 프레임워크로 BackType에서 개발 되어 2011년 트위터에 인수되며 이름을 알리게 되었고, 2013년부터 아파치의 인큐베이터 프로젝트로 전환되어 오픈소스로 지속적으로 개발되고 있다[16]. 하둡이 배치 일괄처리를 위해 개발된 것과 달리, 스톰은 범용 분산 환경 기반 실시간 데이터 처리 시스템으로 개발되어 데이터의 실시간 처리에 중점을 두었다[17]. 이로 인해 스톰은 빠른 데이터 처리 속도, 높은 신뢰성과 확장성, 비교적 낮은 운용 난이도 등과 같은 장점을 가지고 있다. 스톰은 <그림 4>와 같이 하둡의 마스터 노드와 데이터 노드에 해당하는 님버스(Nimbus)와 수퍼바이저(Supervisor)를 통해 분산 처리를 수행하며, 수퍼바이저에 대한 관리는 주키퍼(Zookeeper)를 통해 이루어진다. 스톰은 토폴로지(Topology)를 기반으로 분석을 수행한다. 토폴로지는 Spout와 Bolt를 통해 각 노드 간의 관계를 정의하고, 데이터 입력부터 출력까지 일련의 프로세스를 의미한다. 하둡 클러스터와 유사한 구조를 가지고 있으며 맵리듀스 처럼 일회성 배치가 아닌 스트리밍 데이터 처리를 위한 영구적인 실행을 지원한다. 또한 스트림 내에서 튜플이 리플레이(Replay) 될 경우 분석 결과를 오버카운트하게 되는데 트라이던트(Trident)를 통해 확장할 경우 fault-tolerant 기능을 정확하게 한번(Exactly-once) 분석하여 메시지 중복 처리를 해결할 수 있다.

이번 장에서 설명한 실시간 분석 프레임워크에 대한 유사점과 차이점에 대한 내용은 <그림 5>와 같이 요약할 수 있다. 이는 모두 오픈소스 프레임워크이고, 실시간 및 빅데이터 처리를 지원한다. 또한 JVM 환경에서 동작하며 내결함성과 확장성을 제공한다는 유사점을 가지고 있다. 차이점으로는 스트림과 배치에 대한 처리 방식과 이에 따른 지연 시간 등과 같은 부분에 차이가 있음을 확인할 수 있다.

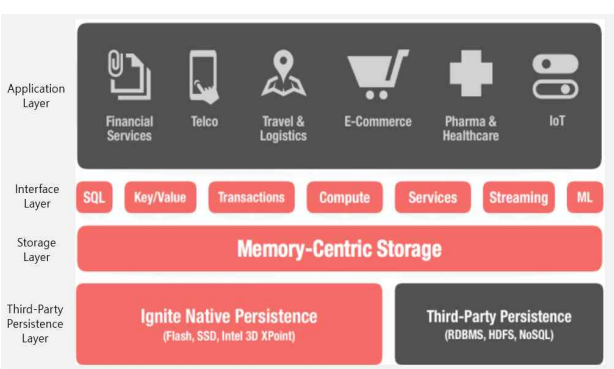


그림 3. 이그나이트 프레임워크 아키텍처
Fig. 3. Ignite Framework Architecture

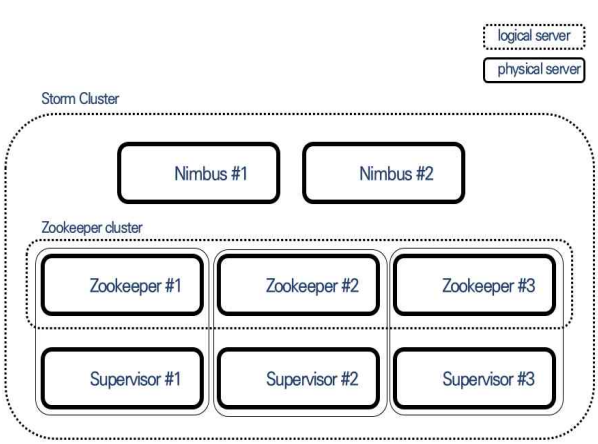


그림 4. 스톰 클러스터 구조
Fig. 4. Storm Cluster Architecture

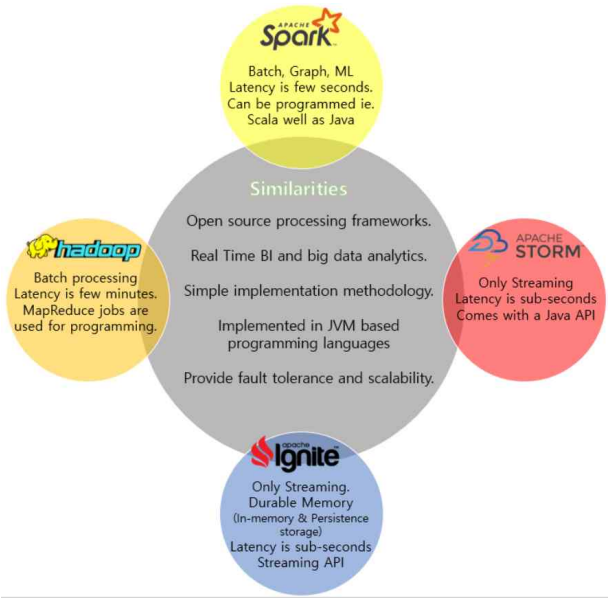


그림 5. 실시간 분석 프레임워크 비교
Fig. 5. Real-time analytics framework comparison

스톰을 활용한 시스템으로는 네이버의 '실시간 콘텐츠 소비 지수 정보 분석 시스템'이 있다[18]. 네이버 TV 연예 카테고리를 대상으로 특정 시간 동안 사용자가 소비한 기사를 분석 대상으로 실시간 분석하여 통계 정보를 제공하는 서비스이다. 사용자가 컨텐츠를 소비하면서 생성되는 데이터를 Kafka에서 수집하고, 이를 스톰 클러스터에서 분석하여 인메모리 DB에 적재한 후 서비스에서 활용한다. 또한 네이버의 마이크로 서비스(Micro Service) 로그를 스톰을 이용하여 분석하고, Elasticsearch Percolator를 이용하여 정의된 규칙에 매칭될 경우 관리자에게 알람을 해주는 '로깅 플랫폼 실시간 알람서비스'도 있다[19]. 기존 알람 모듈은 주기적으로 동작하는 배치 작업을 실행해야 하기 때문에 알람 발송의 실시간성이 보장되지 않는 문제점이 있었다. 또한 마이크로서비스가 추가될 경우 확장성에 제약이 생기고, 수집서버와 알람서버가 이원화되어 알람 처리를 하기 때문에 구조적인 문제점도 있었다. 이러한 문제를 해결하기 위해 스톰을 이용하여 실시간 알람이 가능한 시스템으로 개선하였다. 이에 따라 유연한 확장성과 알람 처리의 일원화에 대한 구조적 개선도 이루어 졌다.

2-2 공간 정보 추출

텍스트에서 추출하는 공간정보는 국제 표준인 ISO Space에서 정의한 공간 개체와 공간 관계로 정의할 수 있다[20]. 이는 텍스트 공간 정보 자동 추출 연구인 SpaceEval에서 말뭉치 구축 및 자동 추출 연구에 활용되기 시작하였으며, 참고 [21]의 연구에서는 이를 한국어 데이터에 적합한 형태로 변형한 한국어 공간 정보 주석 가이드 라인을 발표했다. 공간 정보 주석의 대상 태그를 SpaceEval에서 정의한 공간 정보 주석 태그와 동일하다. 단 주석의 난이도가 매우 높고, 한국어 공간 정보 자동 추출 연구에서 직접적인 추출 대상이 아닌 일부 공간 개체 태그와 공간 관계 태그는 주석을 수행하지 않았다. 공간 정보 태그의 종류는 <표 1>과 같고, 추출 대상에서 제외된 정보는 '*'를 사용하여 따로 표기하였다. 공간 정보에 대한 추출 예시는 <그림 6>과 같다.

텍스트에서 공간 개체와 같은 의미 있는 개체 정보를 추출하는 것을 개체명 인식(Named Entity Recognition)이라 한다. 개체명 인식은 인명, 지명 ,기관 등과 같은 범용적인 정보를 가지는 일반적인 개체명(Generic named entity)과 의학용어, 법률 용어 등과 같은 특정분야 개체명(Domiain-specific named entity)으로 나눌 수 있다. 개체명 인식은 전통적인 통계 기반의 기계학습 방법론으로 부터 심층 신경망(Deep Neural Network) 기반의 방법론으로 발전해왔다. 통계 기반의 기계학습의 대표적인 방법으로는 Hidden Markov Model(HMM), Support Vector Machines(SVMs), Conditional Random Fields(CRFs) 등이 있으며, 심층 신경망 기반의 방법론 중에서는 순차적 레이블링에 좋은 성능을 보이는 양방향 RNN(Bidirectional Recurrent Neural Network)과 CRFs를 결합한 모델인 BiLSTM(Long Short-Term Memory)-CRFs 모델에 대한 연구가 있었다[22]-[25]. 최근에는 대용량의 말뭉치로부터 언어의 문법 및 의미를 학습한 사전학습 언어모델(Pre-trained Language Model)과 미세조정(Fine-Turninng)한 모델을 활용한 연구가 좋은 성능을 보인다. 대표적인 사전학습 언어모델에는 구글(Google)의 BERT(Bidirectional Encoder Representation from Transformers)가 있다. BERT는 발표 당시 자연어처리 관련 11개 태스크에서 최고 성능을 기록했다[26]. BERT의 후속 연구로 참고 [27]은 MLM의 문제점을 개선한 RTD(Replaced Token Detection) 방식으로 학습을 수행한 ELECTRA(Efficiently Learning an Encoder that Classifies Token Replacements Accurately) 모델을 발표하였다. 이는 말뭉치에서 15%만 마스킹하여 학습에 활용하는 MLM과 달리 입력 토큰 전체를 활용하여 보다 효율적으로 학습할 수 있는 방식을 제안하였다.

특정 키워드나 개체 간의 관계에 대한 정보를 추출하는 것을 관계 추출이라 한다. 관계 추출에 대한 연구는 규칙을 사용한 연구와 의존 구문 경로 상 개체 간의 확률을 사용한 연구가 있다[28][29]. 또한 MLP(Multi Layers Perceptron)을 사용하여 ISO Space에 트리거에 대응되는 spatial indicators를 추출하고, 추출된 spatial indicators를 기준으로 CNN(Convolution Neural Network)을 사용해 관계를 형성하는 개체 및 속성을 추출하는 모델을 제안하였다[30]. 참고 [28]과 같이 한국어 공간 관계 추출은 의존 구문 정보를 사용하여 공간 개체들 간의 확률을 이용한 모델이 규칙을 사용한 모델보다 공간 관계 추출 성능이 좋았다. 하지만 의존 구문 정보를 사용한 모델은 재현율은 높지만 정확률이 낮은 추출 성능을 보였다.

표 1. SpaceEval 공간 정보 태그 종류
Table 1. Types of SpaceEval spatial information tags

Spatial Elements Information	PLACE
	PATH
	SPATIAL ENTITY
	NON-CONSUMING
	MOTION
	NONMOTION EVENT(*)
	MOTION SIGNAL
	SPATIAL SIGNAL
	MEASURE
Spatial Relations Information	QSLINK
	QLINK
	MOVELINK
	MLINK
	METALINK(*)

* Translation can misrepresent meaning
그림 6. 공간 정보 추출 예시
Fig. 6. Example of Spatial Information Extraction

Ⅲ. 실시간 공간 정보 추출 시스템 설계 및 구현

이번 장에서는 본 논문에서 제안하는 실시간 공간 정보 추출 시스템에 대한 전체적인 구조 설계와 시스템 세부 계층에 대하여 설명한다.

3-1 제안 시스템

제안 시스템은 데이터 공급 계층과 분석 계층, 검색 및 관리 계층으로 구성되어 있다. 데이터 공급 계층은 실 서비스에 따라 RDBMS, HDFS, Kafka 등으로 커스터마이징이 가능하며, 본 연구에서는 HDFS 및 Kafka를 이용한 데이터 공급 계층을 구성하였다. 통합 테스트를 위하여 기 구축되어 있는 HDFS 클러스터로부터 Kafka로 데이터를 전송하는 Document Producer를 개발하였다. 분석 계층은 공간 정보를 추출하기 위한 형태소 분석(Morpheme Analysis) 및 개체명 추출 엔진과 관련 지식베이스(Knowledge Base)로 구성되어 있다. 데이터 공급 계층으로부터 실시간으로 유입되는 데이터를 스톰 클러스터에서 모니터링하고, 데이터가 입수되면 주키퍼를 통해 각각의 수퍼바이저로 분산하여 처리된다. 검색 및 관리 계층은 분석 계층에서 Elasticsearch 클러스터에 저장한 결과를 검색 API를 통해 서비스에 제공된다. 또한 정교한 분석을 위한 지식베이스 관리 및 스톰 클러스터 관리를 위한 관리도구로 구성되어 있다. 제안 시스템의 구조도는 <그림 7>과 같다.

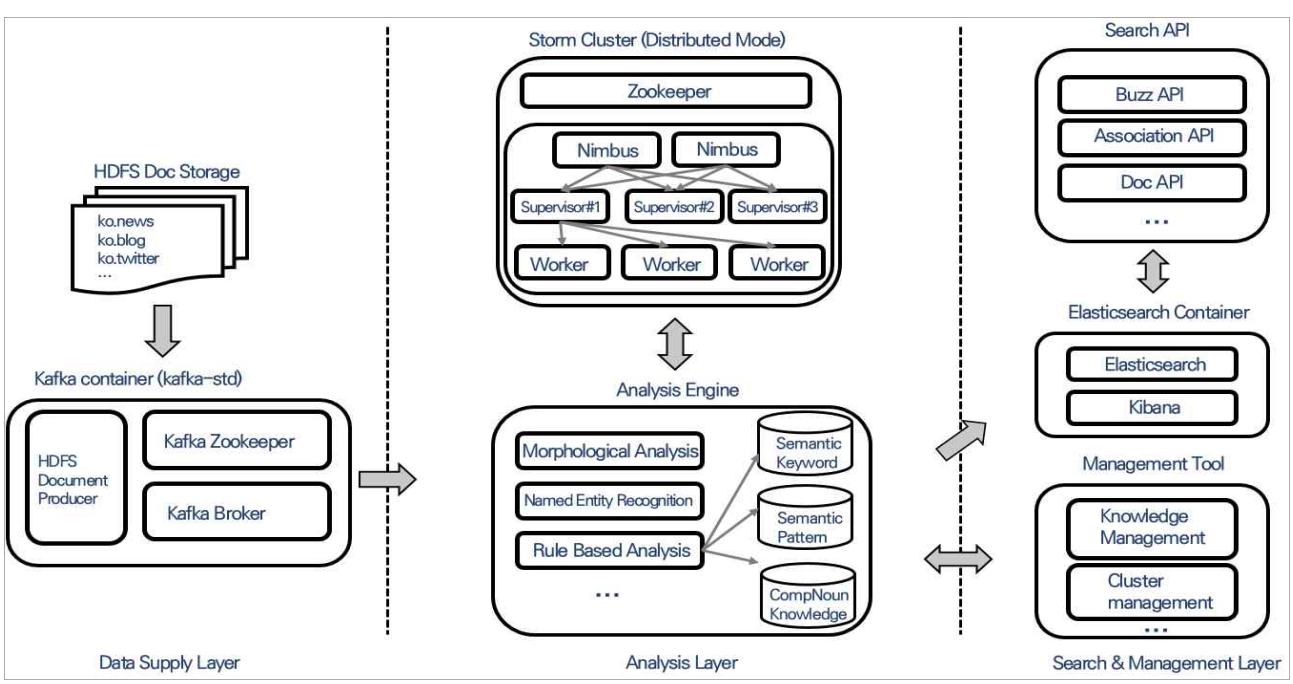


그림 7. 실시간 공간정보 추출 시스템 전체 구조도
Fig. 7. Storm-based Real-Time Analysis System Architecture for Spatial Information Extraction

3-2 데이터 공급 계층

데이터 공급 계층은 향후 연구와의 연계와 확장성을 위해 데이터 소스 별로 Spout를 구현해야 한다. 본 연구에서는 Kafka에서 데이터를 입수하기 때문에 Kafka Spout를 구현했다. Spout는 외부의 데이터 스트림으로 부터 스톰 클러스터 내부로 데이터를 가져오는 역할을 한다.

스톰은 기본적으로 Spout를 제공하지만 트라이던트 API를 사용하기 위해서는 Opaque Spout를 구현해야 한다. 트라이던트는 스톰을 확장하여 상태유지 스트림 프로세싱(Stateful Stream Processing)및 고성능 분산 쿼리를 지원하기 위해 개발된 것으로 정확한 분석을 위해서는 트라이던트를 이용하여 구현해야 한다. 트라이던트는 Fault Tolerance를 위해 3가지 상태를 지원한다. 각 상태와 Spout 타입의 조합에 대한 내결함성 정보는 <그림 8>과 같다. 본 연구에서는 신뢰성 있는 분석을 위해 Opaque Spout와 Opaque State를 사용한다. Kafka Spout Opaque를 생성하면 Kafka의 오프셋(offset) 관리는 지원하지만, 첫 번째 폴링에서 사용할 오프셋은 지정해주어야 한다. 해당 오프셋에 대한 설정 정보는 <표 2>와 같다. 본 연구에서는 UNCOMMITED_EARLIEST을 사용하여 커밋된 오프셋이 없을 경우, Kafka 파티션의 첫 번째 오프셋에서 부터 레코드를 폴링하여 사용했다. 또한 Kafka 브로커의 url과 토픽명, fetch byte 등과 같은 설정을 해주어야 한다.

		State		
		Non-transactional	Transactional	Opaque transactional
Spout	Non-transactional	No	No	No
	Transactional	No	Yes	Yes
	Opaque transactional	No	No	Yes

그림 8. 트라이던트 상태 타입과 Spout 타입의 내결함성 정보
Fig. 8. Combinations of Spouts / States for Exactly-Once Messaging Semantics

표 2. Kafka Spout Opaque 오프셋 설정 옵션
Table 2. Kafka Spout Opaque offset settings options

parameter	description
EARLIEST	The kafka spout polls records starting in the first offset of the partition, regardless of previous commits.
LATEST	The kafka spout polls records starting at the end of the partition, regardless of previous commits.
TIMESTAMP	The kafka spout polls records starting at the earliest offset whose timestamp is greater than or equal to the given startTimestamp.
UNCOMMITTED_EARLIEST, LATEST, TIMESTAMP	The kafka spout polls records from the last committed offset, if any.

3-3 분석 계층

분석 계층은 데이터 공급 계층에서 전달받은 데이터를 필터링하고 전처리 및 분석을 수행한다. Kafka Spout에서 튜플(Tuple) 단위로 전달 받은 데이터를 각각의 트라이던트 Filter와 Function을 이용하여 순차적으로 분석을 수행한다. 토폴로지 순서도는 <그림 9>와 같다. Spout를 제외한 1~3단계 Function은 필터링 및 형식 변환과 같은 전처리를 수행하는 Function이다.

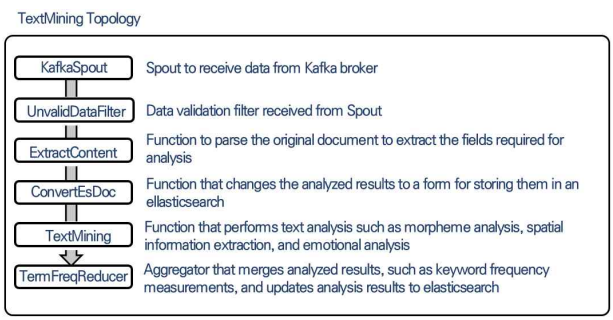


그림 9. 공간정보 추출 토폴로지 흐름도
Fig. 9. Spatial Information Extraction Topology Flow Chart

첫 번째 단계인 UnvalidDataFilter는 전달 받은 튜플이 분석에 필요한 데이터인지 확인하는 Filter이다. 전체 데이터 중에서 특정 데이터만 한정지어 분석이 필요할 경우 사용한다. 사용자 정의 카테고리 별로 문서를 분류하여 분석에 사용할 수 있으며, 본 연구에서는 5,000 글자를 넘는 문서와 한국어 문서를 제외한 나머지 데이터는 필터링하고, 필터링 되지 않은 문서는 다음 단계인 ExtractContent Function으로 전달된다.

두 번째 단계인 ExtractContent Function은 분석에 필요한 필드만 추출하는 기능을 수행한다. 전달 받은 튜플은 JSON 형태이며, 수집 채널 별로 문서의 형식이 다를 수 있기 때문에 분석을 위해 필요한 제목이나 본문과 같은 내용을 추출하는 기능을 한다. 분석해야 하는 필드가 존재하지 않거나, 형식이 잘못된 경우 다음 단계의 분석을 수행하지 않는다.

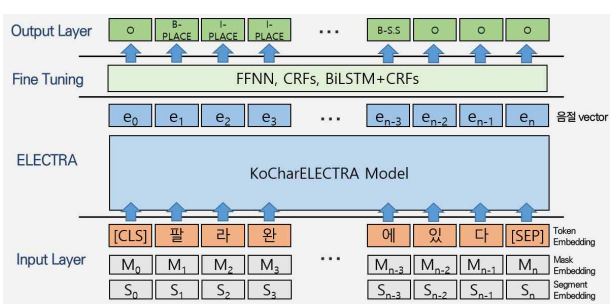
세 번째 단계인 ConvertEsDoc Function은 텍스트마이닝 작업 완료 후에 Elasticsearch 클러스터에 저장하기 위한 형태로 변환하는 기능을 수행한다. Elasticsearch의 인덱스와 타입을 지정하여 클러스터 정보를 세팅하고 저장하기 위한 JSON 형태로 변환한다.

전처리 단계가 끝나면 TextMining Function에서 키워드 및 공간 정보를 추출하기 위한 분석을 수행한다. 본 연구에서는 공간 관계는 제외하고 공간 개체를 대상으로 분석을 수행하며, 심층 신경망을 이용하여 공간 개체를 추출한다. 사용 모델은 자연어처리 관련 11개 태스크에서 최고 성능을 기록했던 BERT의 후속 연구로 발표된 ELECTRA(Efficiently Learning an Encoder that Classifies Token Replacements Accurately) 사전 학습 모델을 미세 조정

(Fine-Tuning)한 모델을 사용한다. 공간 개체를 추출하기 위한 딥러닝 모델의 구조는 <그림 10>과 같다.

키워드 추출은 패턴 기반 키워드 추출기를 이용하여 추출한다. 패턴은 어휘 의미 패턴(Lexico-Semantic Pattern)을 사용하며, 의미 키워드 사전과 의미 태그 및 의미 패턴을 기반으로 구성되어 있다. 의미 키워드는 키워드 후보군이 될 수 있는 키워드 사전이고, 의미 태그는 특정 키워드 집합의 의미를 뜻하는 레이블이다. 의미 패턴은 의미 태그와 형태소 태그, 키워드 등으로 구축된 패턴이며, <그림 11>은 구축된 의미 패턴의 예시이다. 최종적으로 해당 패턴으로부터 FSA(Finite State Automata) 기법을 이용하여 키워드를 추출하게 된다.

마지막 단계인 TermFreqReducer Function은 추출한 키워드와 공간 개체를 Elasticsearch 클러스터에 저장하는 기능을 수행한다. 스톰은 각 각의 수퍼바이저로 분산하여 처리한 각 파티션을 병합하기 위해 Aggregator를 제공한다. 또한 통신을 위해서 HTTP에 비해 상대적으로 빠른 Netty 모듈을 이용하여 네이티브 클라이언트(Native Client)를 통해 통신하는 TransPort 방식을 사용하고, 정확한 분석을 위해 트라이던트의 트랜잭션 ID를 같이 저장한다.



* Example of a syllable-based model for Korean

그림 10. ELECTRA 기반 공간 개체 추출 딥러닝 모델

Fig. 10. Deep Learning Model for Spatial Element Extraction based on ELECTRA

PATTERN MARK : [$] NER TAG, [%] Lexical, [#] Sem tag, [@] Morp tag

입력	형태소 분석 결과 / 개체명 패턴	키워드 추출 결과
Galaxy S20 Black 128GB	Galaxy/#phone + S20/@num_eng + Black@eng + 128 GB/@unit	Galaxy S20 128GB
	#phone @num_eng [⟨3⟩ @unit	
iPhone SE6 Gray 256GB	iPhone/#phone + SE6/@num_eng + Gray/@eng + 256GB/@unit	iPhone SE6 Gray 256GB
	#phone @num_eng [⟨3⟩ @unit	
iPhone 11 pro 128GB	iPhone/#phone + 11/@num + pro/#edition + 128GB/@unit	iPhone 11 pro 128GB
	#phone @num [⟨3⟩ #edition [⟨3⟩ @unit	
실종 10명 등으로 인명 피해 집계가 늘고 있다.	실종/N + 10/NU + 명/ND + … 인명/N + 피해/N + 집계/N + 가/J 늘/V + 고/J +있/AX …	인명 피해 늘다
	@NNG @NNG [⟨3⟩ @VV [=0] !@ETM	
6일 오후 3시 강타할 것으로…	6/NU + 일/ND + 오후/N + 3/NU + 시/ND + 강타하/V + ㄹ/E …	6일 오후 3시
	@NU @ND @NU @ND	

* Example of Keyword Extract Patterns for Korean

그림 11. 키워드 추출을 위해 구축된 어휘 의미 패턴 예시

Fig. 11. Example of Lexico-Semantic Pattern for Keyword Extraction

3-4 검색 및 관리 계층

검색 및 관리 계층은 분석 계층에서 저장한 데이터를 서비스에 활용하기 위한 키워드 조회, 공간 개체 조회, 원본 문서 조회 등과 같은 기능을 API 형태로 제공하고, 스톰 클러스터의 상태와 분석 통계 정보를 확인할 수 있는 대시보드를 제공한다. 또한 분석 결과의 신뢰성 확보를 위한 지식 관리도구도 제공한다. 지식 관리는 시스템 관리자가 관리하는 분석 엔진 관련 공통 지식과 사용자가 직접 관리하는 사용자 지식으로 구분되어 있다. <그림 12>는 분석 시스템 통계정보를 제공하는 대시보드 인터페이스 예시이고, <그림 13>은 지식베이스 관리 인터페이스 예시이다.

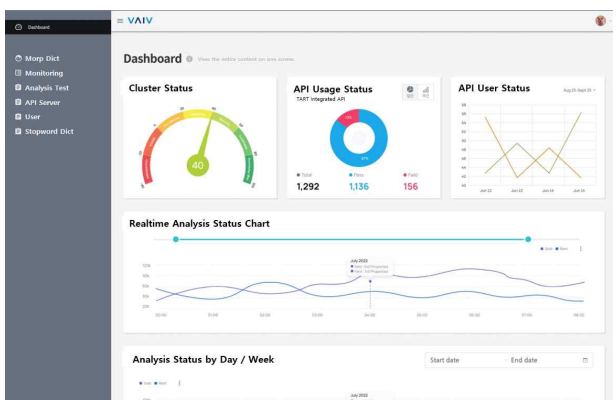


그림 12. 대시보드 인터페이스 예시

Fig. 12. Example of Dashboard Interface

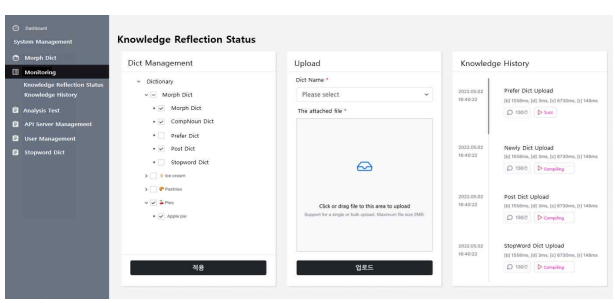


그림 13. 지식베이스 관리 인터페이스 예시

Fig. 13. Example of Knowledge Management Interface

검색 API와 대시보드의 사용 현황 조회 등을 위하여 Django 프레임워크를 기반으로 API 서버를 구축하였다. 모든 API는 사용자 통계를 위해 사용자 ID를 입력 받아 Django Model에 저장한다. 분석 결과는 모두 Elasticsearch에 저장되어 있기 때문에 검색 API는 Elasticsearch API를 Wrapping한 형태로 구축이 되어 있고, 필요 시 Django View와 Service를 추가하여 확장할 수 있다. 분석 API의 경우는 데몬(Daemon) 형태로 실행 중인 분석 엔진과 HTTP 형태로 통신하도록 설계하였다.

지식 관리도구의 경우 고품질의 분석 결과를 얻기 위한 지식베이스를 관리하기 위해 설계하였다. 분석에 필요한 형태소 사전, 의미 사전, 패턴 사전 등은 관리자에게만 접근이 허용되고, 사용자별 맞춤형 분석 결과 제공을 위한 사용자 사전은 사용자나 특정 그룹에 대해서만 접근이 허용된다. 분석 결과 또한 검색 API를 호출한 사용자에 따라 맞춤형 결과를 제공하도록 설

계하였다. 지식 관리도구를 통해 웹상에서 사전의 엔트리를 추가, 삭제 등이 가능하며, 사전 전체에 대한 다운로드 및 업로드 기능을 통해 벌크 로드(Bulk load) 기능을 지원한다.

Ⅳ. 분석 테스트

본 연구에서 설계 및 구축한 실시간 분석 시스템을 테스트하기 위한 환경으로 스톰 클러스터는 님버스 노드 1대, 수퍼바이저 노드 3대로 구축하였고, Elasticsearch와 Kafka는 1대의 서버에 도커 컨테이너(Docker Container)를 이용하여 구축하였다. 서버의 사양은 <표 2>와 같다.

스톰 클러스터의 수퍼바이저는 분산병렬 처리를 위해 각 노드당 워커(Worker)의 수와 익스큐터(Executor)의 수를 설정할 수 있다. 수퍼바이저는 N개의 워커를 실행시킬 수 있고, 워커는 N개의 익스큐터를 실행시킬 수 있다. 각각의 익스큐터에서 토폴로지에 정의된 Spout와 Function, Filter 등을 실행 시키는 역할을 한다. 스톰의 토폴로지 실행 구조는 <그림 14>와 같다.

표 2. 테스트를 위한 실시간 분석 시스템 서버 사양
Table 2. Server Specifications for Testing

Server Name	CPU	RAM
Storm Cluster (Nimbus 1EA, Supervisor 3EA)	Intel(R) Xeon(R) E5606, 2.13 GHz, 8 core	48Gb (4Gb * 12)
TextMining server	Intel(R) Xeon(R) E5-2630, 2.4GHz 32 core	128Gb (16Gb * 8)

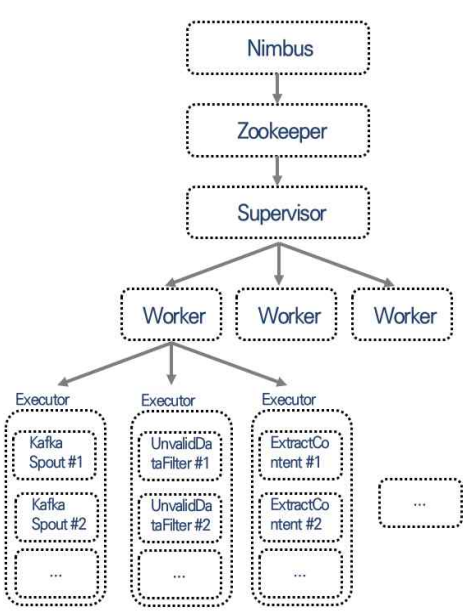


그림 14. 스톰 토폴로지 실행 구조
Fig. 14. Storm Topology Execution Architecture

본 연구에서는 수퍼바이저 노드 당 1개의 워커 프로세스를 실행하도록 설정하고 님버스를 통해 토폴로지를 배포하였다. 토폴로지의 설계에 따라 익스큐터 수는 결정되며, <그림 9>의 토폴로지 프로세스에서 상대적으로 지연시간이 더 발생하는 Kafka Spout와 TextMining 단계에서는 익스큐터를 추가 할당하여 분석을 수행하였다.

실험 방법은 Kafka Spout와 TextMining 단계의 익스큐터 수를 변경해가며, 10만 건의 뉴스 기사를 분석 후 단계별로 발생하는 실행 지연시간(Execute Latency), 프로세스 지연시간(Process Latency)과 가동률(Capacity)을 측정한다. 실행 지연시간은 실행 메서드(Execute Method)에서 튜플이 소비하는 평균 시간이고, 프로세스 대기시간은 처음 수신된 튜플을 분석하는데 걸리는 평균 시간이다. 가동률은 해당 단계에서의 할당된 자원 사용률으로 볼 수 있으며, <수식 1>과 같이 나타낼 수 있다. 가동률이 1.0에 근사할 경우 해당 단계에 익스큐터를 추가 할당하여 최적화된 토폴로지를 설계할 수 있다.

$$capacity = \frac{number\ of\ executed * average\ execute\ latency}{measurement\ time} \quad (1)$$

테스트에 사용한 데이터는 2019년 3월에 작성되어 기수집된 평균 용량 2KB의 네이버 뉴스 기사를 활용하였다. 실험 결과 <표 3>과 같은 성능을 확인할 수 있었다. <표 3>의 Num Executor는 토폴로지에 할당된 전체 익스큐터 수와 Kafka Spout 및 TextMining 단계에 할당된 익스큐터 수를 의미하고, Assigned Mem은 토폴로지에 할당된 메모리를 의미한다.

추가 익스큐터를 할당하지 않은 실험 1의 경우 Kafka Spout는 실행 지연시간 377.53ms, 프로세스 지연시간 383.76ms으로 측정되었고, 가동률이 0.951로 익스큐터에 할당된 자원을 대부분 사용하고 있는 것을 확인할 수 있다. TextMining 단계의 경우 25.03ms, 24.51ms, 0.12로 측정되었다. TextMining 단계에 3개의 익스큐터를 할당한 실험 2의 경우 Kafka Spout는 실험 1과 유사하게 나타났다. 하지만 익스큐터를 추가 할당한 TextMining의 경우 실행/프로세스 지연시간이 각각 13.3ms, 10.01ms 감소됨을 확인했다. 이는 추가 할당된 TextMining 익스큐터에서 병렬로 분석을 수행함에 따라 나타난 현상으로 볼 수 있다. 실험 3은 Kafka Spout에 3개의 익스큐터를 할당한 결과 75.2ms, 76.2ms로 실험 1,2 대비, 약 80%의 지연시간 감소가 있었다. 하지만 토폴로지에 인입되는 튜플의 수가 급격하게 증가하면서 TextMining 단계의 프로세스 지연시간은 60% 가량 증가하였다. 실험 4의 경우도 마찬가지로 익스큐터를 추가 할당할 경우 해당 단계의 시간 감소가 있었으나, 실험 5~7의 경우 익스큐터를 추가 할당하더라도 지연시간이 감소되지 않는 현상이 발생하였다. 이는 HTTP API 형태로 통신하는 TextMining 엔진의 성능 문제로 확인되었다.

표 3. Executor 수에 따른 분석 소요 시간
Table 3. Time Required for Analysis According to the Number of Executors

Experiment Number	Num Executor	Assigned Mem(MB)	Kafka Spout exec latency(ms)	Kafka Spout process latency(ms)	Kafka Spout capacity	TextMining exec latency(ms)	TextMining process latency(ms)	TextMining capacity
1	15 (1,1)	1920	377.59	383.76	0.95	25.03	24.51	0.12
2	17 (1,3)	2176	**397.24**	398.83	1.01	11.7	**14.50**	0.11
3	19 (3,3)	2432	**75.28**	76.22	0.47	5.15	**58.40**	0.11
4	22 (6,3)	2816	39.73	39.64	0.49	3.38	**33.42**	0.11
5	25 (6,6)	3200	36.99	37.0	0.49	1.32	39.09	0.051
6	28 (6,9)	3584	41.28	41.31	0.53	1.08	36.72	0.044
7	34 (6,15)	4352	39.54	39.55	0.56	0.81	36.25	0.043

TextMining 단계에 할당된 익스큐터의 수가 증가하더라도, 텍스트 분석을 수행하는 분석 서버의 데몬(Daemon)은 증가하지 않았기 때문에, 익스큐터에서 실행하는 실행 지연시간은 감소하지만 분석 서버와의 통신과 분석 수행 시간과 연관된 프로세스 지연시간은 감소하지 않았다.

V. 결 론

최근 성숙기에 접어든 소셜 미디어 시장 성장률이 코로나19를 기점으로 다시금 상승 추세로 돌아서며, 이에 따라 생산되는 데이터의 양도 급속도로 증가하였다. 이러한 소셜 미디어 데이터를 기존의 공간 정형데이터와 융합 분석하기 위해서는 텍스트 공간 정보를 실시간으로 추출할 수 있어야한다. 이러한 문제를 해결하기 위해 본 논문에서는 소셜 미디어에서 생산되는 대용량의 빅데이터를 다양한 관점에서 분석하고 활용하기 위한 텍스트 공간 정보를 실시간으로 추출하여 분석할 수 있는 스톰 기반 실시간 분석 시스템을 제안하였다.

본 논문에서는 실시간으로 수집되는 데이터로부터 공간 정보를 자동으로 추출하여 분석할 수 있는 스톰 기반 실시간 분석 시스템을 제안하였다. 이를 위해 데이터 공급 계층, 분석 계층, 검색 및 관리 계층으로 나누어 설계하여 시스템 확장성을 확보하였다. 실험에서는 HDFS와 Kafka를 통해 데이터 공급 계층을 개발하였고, 공간 정보 추출을 위한 텍스트 마이닝 엔진을 통해 분석 계층을 개발하였다. 검색 및 관리계층은 Elasticsearch와 Django를 통해 서비스에 활용할 수 있는 API 및 관리도구를 개발하였다.

실험 결과, 실험 1과 같이 15개의 익스큐터를 사용할 경우 Kafka Spout는 실행 지연시간 377.53ms, 프로세스 지연시간 383.76ms으로 측정되었고, 가동률이 0.95로 익스큐터에 할당된 자원을 대부분 사용하고 있는 것을 확인할 수 있었다. TextMining 단계의 경우 25.03ms, 24.51ms, 0.12로 측정되었다. 이를 기반으로 익스큐터를 추가 할당함에 따라 지연시간이 감소되어 전반적인 성능이 향상되었으나, 실험 5~7의 경우 익스큐터를 추가 할당하더라도 지연시간이 감소되지 않는 현상이 발생하였다. 이는 HTTP API 형태로 통신하는 TextMining 단계에서 병목 현상이 생김을 확인할 수 있었다. 이를 통해 하드웨어 자원과 데이터의 규모에 따른 클러스터 구축 방안에 대해서 알 수 있었고, 공간 정보를 추출하여 다양한 사용자 위치 기반 서비스에 활용할 수 있는 토대를 마련하였다.

향후 연구로 다양한 하드웨어 사양과 데이터 크기에 따른 공간 데이터 분석 작업의 빠른 실시간성 확보를 위한 하드웨어 및 계층 별 최적 구성에 대한 연구가 필요하고, 분석 엔진의 병렬화 및 로드 밸런싱(Load Balancing) 기능 개발이 필요하다. 또한 서비스 측면에서 다차원 분석을 위해 추가 기능 개발 및 고도화에 대한 연구도 수행되어야 할 것이다.

감사의 글

본 논문은 2022년도 정부(국토교통부)의 재원으로 국토교통과학기술진흥원의 지원을 받아 수행된 연구임
(RS-2022-00143336, 공간 지식추론 엔진 기술개발 사업)

참고문헌

[1] S. Y. Kim, “Mobile SNS service trends and forecasts”, Journal of Communications and Networks, Vol. 26, No. 4, pp. 19-25, Mar 2009.

[2] S. Kemp, DataReportal, “DIGITAL 2022: GLOBAL OVERVIEW REPORT” [Internet]. Available: https://datareportal.com/reports/digital-2022-global-overview-report

[3] B. S. Kim, M. Kim, M. C. Kim, “Korea Big Data and Analytics Forecast, 2022–2026”, IDC Corporate, Doc # AP48913222, Sep 2022.

[4] N. Paolo, G. Pantaleo, G. Sanesi, “A Hadoop based Platform

for Natural Language Processing of Web Pages and Documents", Journal of Visual Languages & Computing, Vol. 31, pp. 130-138, OCT 2015., https://doi.org/10.1016/j.jvlc.2015.10.017

[5] Hsieh, M. Yen, T. H. Weng, K. C. Li, "A Keyword-Aware Recommender System using Implicit Feedback on Hadoop", Journal of Parallel and Distributed Computing, Vol. 116, pp. 63-73, Jun 2018., https://doi.org/10.1016/j.jpdc.2017.12.008

[6] BIGIDEAN, Bigidean Data Platform(BDP) [Internet], Available: https://sites.google.com/view/bigidean/bdp

[7] EXEM, Exem Bigdata System(EBIGS) [Internet], Available: https://www.ex-em.com/product/ebigs

[8] VAIV Company, Contextual Finder, [Internet], Available: https://blog.naver.com/vaiv_company/221412157841

[9] J. H. Park, S. Y. Lee, D. H. Kang, J. H. Won, "Hadoop and MapReduce". Journal of Korea Data & Information Science Society, Vol. 24, No. 5, pp. 1013-1027, Sep 2013., https://doi.org/10.7465/jkdi.2013.24.5.1013

[10] S. B. Heo, D. C. Kang, J. Y. Choi, "Hadoop based Deep Learning Framework Technology Trend", Communications of the Korean Institute of Information Scientists and Engineers, Vol. 37, No. 10, pp. 25-31, Oct 2019.

[11] Seoul National University Industry-University Cooperation Foundation et al., "Spatial Knowledge Inference Engine Technology Development Planning Final Report", Korea Agency for Infrastructure Technology Advancement, Republic of Korea, OTKCRK220001, pp. 1~15, Nov 2021.

[12] J. Y. Koo, "A Study on the Spatial Distribution of Commercial Space in Seoul Area using the Location-based Social Network Service Data", The Geographical Journal of Korea, Vol. 50, No. 4, pp. 491-502, Dec 2016., G704-001284.2016.50.4.002

[13] S. Wang, R. Sinnott, S. Nepal, "P-GENT: Privacy-Preservi ng Geocoding of Non-Geotagged Tweets", In 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), New York, USA, pp. 972-983. Aug 2018., 10.1109/TrustCom/BigDataSE.2018.00137

[14] H. K. Lee, Y. S. Son, J. B. Kim, "A Study on the SNS analysis model based on the Storm", Asia-pacific Journal of Multimedia Services Convergent with Art, Humanities, and Sociology, Vol. 6, No. 11, pp. 529-536, Jun 2016., 10.35873/ajmahs.2016.6.11.048

[15] J. H. Eom, T. H. Kim, S. W. Lee, C. H. Jeong, H. M. Jeon "The Trend of Next Generation of Real Time Bigdata Distribution System-focus on Spark and Storm", Institute for Information & communication Technology Planning & Evaluation, pp. 1-13, Dec 2014.

[16] Wikipedia, Apache Storm [Internet], Available: https://en.wikipedia.org/wiki/Apache_Storm

[17] V. Khadilkar, M. Kantarcioglu, B. Thuraisingham, "StormRider: Harnessing "Storm" for Social Networks", Proceedings of the 21st International Conference on World Wide Web, New York, USA, pp. 543-544, Apr 2012., 10.1145/2187980.2188118

[18] H. H. Ahn, NAVER, "Real-Time Analysis of Large-Capacity Streaming Data" [Internet], Available: https://d2.naver.com/helloworld/7731491

[19] S. J. Lee, J. I. Lee, E. Sagynov, NAVER, "Improving NELO2 Alarm Function using Storm and Elasticsearch Percolator", Available: https://d2.naver.com/helloworld/1044388

[20] ISO-24617-7:2014, Language Resource Management - Semantic annotation Framework - part 7: Spatial information (ISOspace), ISO/TC 37/SC 4 Language Resource Management, Dec 2014.

[21] Chungbuk National University, Lake Lab., Korean SpaceBank v2.0 Guideline, Oct 2017.

[22] S. H. Na, J. W. Min, "Character-Based LSTM CRFs for Named Entity Recognition", Proc. of the Korean Information Science Society Conference, Republic of Korea, pp. 729-731. Jun 2016.

[23] H. Y. Yu, Y. J. K, "Expansion of Word Representation for Named Entity Recognition based on Bidirectional LSTM CRFs", Journal of KIISE, Vol. 44, No. 3, pp. 306-313. Mar 2017., https://doi.org/10.5626/JOK.2017.44.3.306

[24] K. W. Park et al., "KACTEIL-NER: Named Entity Recognizer using Deep Learning and Ensemble Technique", Proc. of the 29th Annual Conference on Human and Cognitive Language Technology, Republic of Korea, pp. 324-326, Oct 2017.

[25] H. J. Kim, H. S. Kim, "How to Use Effective Dictionary Feature for Deep Learning based Named Entity Recognition", Proc. of the 31th Annual Conference on Human and Cognitive Language Technology, Republic of Korea, pp. 293-296, Oct 2019.

[26] A. Vaswani et al, "Attention is all you need", Advances in Neural Information Processing Systems, State of California, pp. 5998–6008, Dec 2017.

[27] K. Clark, M. T. Luong, Q. V. Le, and C. D. Manning, ELECTRA: pre-training text encoders as discriminators rather than generators, Proc. of 8th International Conference on Learning Representations, pp. 1-14, Mar 2020, https://doi.org/10.48550/arXiv.2003.10555

[28] B. G. Kim, J. S. Lee. "Extracting Spatial Entities and Relations in Korean Text" Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics:

Technical Papers, Japan, pp. 2389-2396, Dec 2016.
[29] H. Salaberri, O. Arregi, B. Zapirain, "IXAGroupEHUSpace Eval:(X-Space) A WordNet-based Approach Towards the Automatic Recognition of Spatial Information Following the ISO-Space Annotation Scheme," In Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), Colorado, pp. 856-861, Jun 2015.
[30] A. Mazalov, B. Martins, D. Matos. “Spatial Role Labeling with Convolutional Neural Networks,” Proceedings of the 9th Workshop on Geographic Information Retrieval. ACM, pp. 1-7, Nov 2015, https://doi.org/10.1145/2837689.2837706

전원표(Won-Pyo Jeon)

2014년 : 강원대학교 일반 대학원 (공학석사)
2014년～현 재: 바이브컴퍼니 스마트시티연구소 책임연구원
※관심분야 : 문서 분류, 개체명 인식, 텍스트 마이닝, 실시간 분석, 스마트시티(Smart City) 등

윤효근(Hyo-Gun Yoon)

2002년 : 공주대학교 일반 대학원 (이학석사)
2006년 : 공주대학교 일반 대학원 (공학박사)
2007년～2007년: 전북대학교 BK21 연구교수
2007년～2010년: (주)시지웨이브 연구소장
2010년～2012년: 건국대학교, 소셜미디어 클라우드 센터 연구교수
2012년～2013년: (주)엠앤엘솔루션, 연구기획 부장
2013년～2015년: (주) 솔트룩스, 전략연구본부 부장
2015년～2016년: 행정안전부 정부통합전산센터 빅데이터과 주무관
2016년～2018년: (주)엑셈, 전략기획본부 부장
2018년～2020년: 한국표준협회 스마트혁신본부 수석연구원
2020년～현 재 : 바이브컴퍼니 스마트시티연구소 연구개발팀장
※관심분야 : 인공지능, 빅데이터, 클라우드, IoT, 디지털트윈 등

안창원(Chang-Won Ahn)

1994년 : 한국과학기술원 일반 대학원 (공학석사)
1998년 : 한국과학기술원 일반 대학원 (공학박사)
1999년～2019년: 한국전자통신연구원(ETRI) 책임연구원
2019년～현 재: 바이브컴퍼니 스마트시티연구소 연구소장
2021년～현 재: 국립부경대학교 기술경영전문대학원 겸임교원
※관심분야 : 확률모형, 소셜 시뮬레이션, 디지털트윈, 스마트시티 등