Journal of the Korea Industrial Information Systems Research Vol. 23 No. 5, Oct. 2018 : 61-76
http://dx.doi.org./10.9723/jksiis.2018.23.5.061 ISSN:1229-3741

Python을 이용한 SNS 크롤링 시스템 구축

(Building an SNS Crawling System Using Python)

이 종 화[1)*]
(Jong-Hwa Lee)

요 약 현대인이 살고 있는 네트워크 세상으로 모든 사물들이 들어오고 있다. 사물에 센서를 부착하는 사물인터넷의 영향으로 인해 네트워크로 실시간 데이터를 주고받는 것이 가능해졌다. 현대인들의 필수품인 모바일 디바이스는 일상생활의 모든 자취를 실시간으로 남기는 역할을 하고 있다. 바로 소셜 네트워크 서비스를 통하여 정보획득 활동과 커뮤니케이션 활동을 실시간으로 거대한 네트워크에 남기고 있는 것이다. 비즈니스 관점에서 고객의 니즈 분석은 바로 SNS 자료에서부터 시작된다는 등가가 성립된다. 본 연구는 웹 환경의 SNS 콘텐츠를 파이썬을 이용하여 실시간으로 자동 수집 시스템을 구축하고자 한다. 세계적으로 많은 이용자수를 확보하고 있는 인스타그램, 트위터, 유튜브의 비정형적 데이터 수집 시스템을 통하여 고객의 니즈 분석에 도움이 되고자 한다. 파이썬의 웹드라이버 환경에서 가상 웹브라우저를 이용하여 마이닝 처리와 NLP 과정을 거쳐 DB에 저장된다. 본 연구의 결과 웹페이지를 통하여 서비스를 진행하고자하며 검색 기능만으로 원하는 데이터가 자동 수집되며 데이터의 시계열 분석을 통하여 네티즌의 이슈 반응을 실시간으로 확인할 수 있었다. 또한 검색부터 실행결과가 나오기까지 5초 이내 이루어지므로 제시된 알고리즘의 우수성을 확인하였다.

핵심주제어 : 빅데이터, 소셜네트워크서비스, 웹마이닝, 웹마이닝, 웹 크롤러, 파이썬

Abstract Everything is coming into the world of network where modern people are living. The Internet of Things that attach sensors to objects allows real-time data transfer to and from the network. Mobile devices, essential for modern humans, play an important role in keeping all traces of everyday life in real time. Through the social network services, information acquisition activities and communication activities are left in a huge network in real time. From the business point of view, customer needs analysis begins with SNS data. In this research, we want to build an automatic collection system of SNS contents of web environment in real time using Python. We want to help customers' needs analysis through the typical data collection system of Instagram, Twitter, and YouTube, which has a large number of users worldwide. It is stored in database through the exploitation process and NLP process by using the virtual web browser in the Python web server environment. According to the results of this study, we want to conduct service through the site, the desired data is automatically collected by the search function and the netizen's response can be confirmed in real time. Through time series data

* Corresponding Author : newjwcom@daum.net
Manuscript received October 3, 2018 / revised October 16, 2018 / accepted October 23, 2018
1) 부경대학교 경영학부, 제1저자 및 교신저자

analysis. Also, since the search was performed within 5 seconds of the execution result, the advantage of the proposed algorithm is confirmed.

Key Words : Bigdata, SNS, Web Mining, Web Crawler, Python

1. 서 론

18세기 후반, 영국의 제임스 와트(James Watt)가 발명한 화석연료를 이용한 증기기관을 바탕으로 한 1차 산업혁명과 19세기 말, 전기가 이용되면서 공장에 전력 공급이 가능하게 되었고 기계를 통해 대량생산이 가능하게 되면서 2차 산업혁명이 시작되었다. 1970년대 IT 기술이 빠르게 확산되고 컴퓨터 제어를 통한 생산 자동화 및 정보 시스템이 가능해지면서 3차 산업혁명의 발판이 만들어졌다. 증기기관, 전기, 컴퓨터 IT 기술이 사회를 혁신하고 일하는 방식을 바꿨다. 첨단 기술의 발전이 기존 일자리를 없앤다는 불안이 가득한 4차 산업혁명 시대는 스마트폰과 인공지능(AI), 로봇 등 신기술이 단순 업무를 대체하게 되므로 단순 일자리는 사라질 수밖에 없다[1,2]. 인공지능은 단순히 사람이 입력하는 지식을 출력하는 것이 아니다. 다양한 경로로 입력된 지식을 바탕으로 학습하고 결론을 도출하는 것이다. 4차 산업혁명을 주도하는 인공지능은 시맨틱 웹(Semantic Web) 기술이 뒷받침되어 발전하고 있다. 시맨틱 웹은 컴퓨터 스스로가 인터넷상의 정보를 탐색하고 수집하여 논리적으로 추리하는 정보처리 기술을 말한다[3,4]. 이러한 기술은 단순 업무적인 일자리를 대신하는 것을 넘어 상황 인식을 통한 이용자 맞춤형 서비스까지 제공하고 있다. 인간만이 가질 수 있는 상황 판단력과 적응력을 컴퓨터가 알아서 찾아주는 진보된 기술이 시작되고 있다[5,6].

테라바이트(TeraByte)에서 페타바이트(Peata Byte)까지 이르는 데이터베이스를 관리하는 도구로서, 데이터를 수집, 저장, 관리, 분석하는 기술을 빅데이터라고 한다. 빅데이터는 단순 데이터 수집을 넘어서 가치를 추출하고 결과를 분석하는 기술이다. 빅데이터 기술은 복잡하고 어렵지만, 시시각각으로 변하는 현대 사회를 더욱 정확하게 예측하여 개인에게 맞춤형 정보를 제공할 수 있고, 빠른 분석 속도로 사용자 요구에 대응할 수 있다.

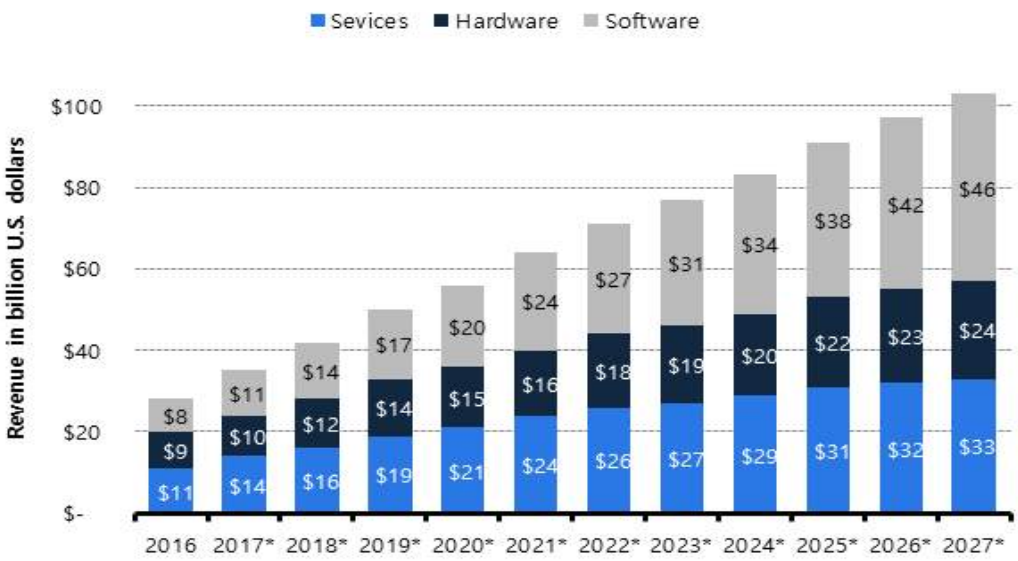

Fig. 1 The World Big Data Market Size(Wikibon, 2018)

시장 조사 기관인 위키본(Wikibon)에 따르면 빅데이터 시장 중 소프트웨어 부분이 IT내 산업 중에서 가장 빠른 속도로 증가하는 것으로 나타났다. Fig. 1을 살펴보면 서비스, 하드웨어, 소프트웨어를 모두 포함한 세계 빅데이터 시장은 2018년 420억 달러에서 2027년까지 1,030억달러 규모로 성장할 것으로 전망하고 있다. 이러한 시장 변화는 연간 약 10.5%의 성장률을 보일 것으로 예측되며 전 세계 빅데이터 시장의 소프트웨어 부문은 모든 범주 중 가장 빠른 속도로 평창하고 있으며 2018년 140억 달러, 2027년 460억 달러로 12.6% 이상의 연 평균 성장률을 달성할 것으로 보고 있다. 한편, 국내 빅데이터 시장 규모를 살펴보면 한국과학기술정보연구원에 따르면 2016년 3억 3천 2백만 달러, 2018년에 5억 3천 9백만 달러, 2020년엔 8억 9천 3백만 달러로 연평균 24% 이상의 데이터 시장의 성장을 전망하고 있다[7].

본 연구는 데이터 산업의 성장과 함께 SNS 사용도 증가하고 있으며 고객 니즈 분석에 주요한 재료인 소셜 데이터 수집에 관한 연구를 진행하고자 한다. 인스타그램, 트위터, 유튜브 등 인터넷 사용자들이 즐겨 사용하는 서비스에서 연구에 필요한 자료만을 자동 수집 가능한 시스템을 설계, 구현, 실험을 하고자 한다.

2. 이론적 배경

2.1 소셜네트워크서비스(Social Network Services)

엔터테인먼트를 이끌고 있는 배우들은 일상생활에서 커피 한 잔의 여유를 즐기는 모습을 공개한다. 수많은 경쟁 연예인들 속에서 나의 존재감을 확인하며 네티즌들의 반응을 보는 것 또한 우리의 일상적인 사회 패턴이 되고 있다. 우리는 페이스북, 트위터, 블로그, 유튜브 등 다양한 소셜 미디어(SNS)의 등장으로 개개인과의 관계를 맺고, 소통하며 관심사를 공유한다. 1인 미디어, 실시간 미디어로 불리는 SNS를 통해 개인 간의 실시간 소통이 이루어지고 다양한 정보를 원활하게 얻을 수 있게 된다. 하지만 개인에게 제공되는 정보가 지나치게 많은 '정보의 홍수 시대'가 되었다. 페이스북은 매월 활동하는 가입자만 약 21억명 이상이며 매일 50억회 이상의 글이 게재되며, 인스타그램은 매일 약 9,500만 개의 사진이 업로드 된다. 트위터는 매일 5억 개의 트윗이 전송되며 매초당 6,000개의 트윗이 생산된다. 하루 이용자만 3,000만 명 이상인 유튜브는 월 단위 사용자만 15억 명이며 분당 400분 분량의 동영상 콘텐츠가 업로드 된다고 한다[8]. 이렇게 많은 정보의 바다 속에서 자신이 원하는 자료를 찾는 것이 점점 어려워지고 있다. 큐레이션(Curation)은 온라인상에서 질 높은 정보를 수집, 공유하고, 이러한 정보에 목적에 따른 가치를 부여해 사람들이 소비할 수 있도록 도와주는 작업을 한다. 다양한 자료를 조합해내는 파워블로거나, 인터넷 백과사전인 위키피디아, SNS 글 등록시 해시태그(#)또한 그런 맥락이라 본다. 객관적 사실을 신속히 전하는 뉴스 미디어의 정보 전달의 속도보다 개인의 일상을 소개하는 SNS에서 전달되는 사건들의 정보 속도가 더 빠르게 전달되는 것 또한 큐레이션의 기술이라 본다. 이런 SNS에서의 이슈가 뉴스 미디어의 재료로 사용되는 것이 일상이 되어 가고 있다[9,10,11].

사진 한 장을 통하여 전달되는 일상 공유가 이젠 크리에이티브 아티스트(Creative Artist)를 통하여 확장되고 있다. 마음에 드는 음악과 동영상을 감상하고, 직접 만든 콘텐츠를 웹 환경에 공유하여 친구, 가족뿐 아니라 전 세계 네티즌과 공유할 수 있는 1인 콘텐츠 시대를 이끌고 있는 SNS는 유튜브이다. 남녀노소 할 것 없이 개인이 갖고 있는 끼를 표현하며 스타 크리에이터의 영향력이 점점 우리 일상으로 자리 잡고 있다. 가수 싸이, 방탄소년단, 원더걸스 등 빌보드 싱글차트에 오른 K팝의 열풍을 주도하기 위한 마케팅 전략적 차원의 SNS 역할은 네트워크 시대를 살아가는 "디지털 원주민"들을 위한 기본적인 마케팅 채널로 자리 잡고 있는 것이다. Lee et al.[12]은 SNS 확산에 관한 연구를 위해 페이스북을 콘텐츠로 활용하여 서비스 확산 모델을 연구하였으며, Lee et al.[13]은 인스타그램 콘텐츠를 활용하여 국내 특정 지역 상권의 크기, 움직임 등 공간 분석을 연구하였다[12,13]. Park and Choi[14]

는 소셜 데이터에서 무엇을 이야기하고, 어떤 이가 그 이야기에 주목하며 응답하는지 또한, 생성된 콘텐츠가 얼마나 지속하며 어떤 콘텐츠와 연결되며 누구와 네트워크를 유지하는지 등 6가지 체계로 정리하였다[14]. Kim et al.[2]의 연구에서는 특정 카드사의 정형적 데이터와 2차 데이터인 소셜 미디어 데이터를 활용하여 소비 트랜드 분석을 연구하였다[2]. 이렇듯 많은 사회 과학 연구자들의 SNS 분석을 통한 사회 기여점을 찾고자 노력을 하고 있다[15].

본 연구는 소셜 데이터 분석의 연구 사례의 증가와 IT 비전공자를 위한 다양한 오픈 소스 소프트웨어들의 활용 측면을 고려하여 실제 웹의 데이터를 채굴할 수 있는 크롤링 시스템을 연구하고자 한다. 원하는 소셜 미디어와 검색어, 기간 등을 설정하면 해당 조건에 만족하는 콘텐츠를 쉽게 얻을 수 있다면 빅데이터와 소셜 미디어, 웹마이닝 연구에 많은 도움이 될 것으로 본다.

2.2 크롤링(Crawling)

잡코리아의 데이터 관련 업무를 살펴보면 데이터 분석과 관리, 데이터 분석 서버 관리, 데이터 분석 기획, 데이터 분석 개발, 리뷰 크롤러 개발, 텍스트 분석, 웹 크롤러 플랫폼 개발, 크롤링 데이터 분석, 데이터 수집 시스템 개발 및 크롤링 개발 등의 업무를 요하는 채용 정보를 흔히 볼 수 있다[16].

기업들의 신규 채용 정보 속에서 빅데이터 관련 직종의 인기도를 확인할 수 있다. 데이터 분석을 위한 재료인 데이터 수집 범위의 역량에 따라 그 결과는 크게 차이가 난다. 일반적인 텍스트마이닝(TextMining) 과정은 분석 데이터 수집을 시작으로 불필요한 불용어 및 조사 등을 제거하는 작업인 전처리 과정과 의미 있는 단어들만을 추출하기 위한 명사 처리 과정을 거쳐 누구나 쉽게 분석 결과를 알 수 있도록 시각화하는 결과물 도출까지이다[17,18]. 가령 데이터 수집 가능한 데이터를 수작업으로 진행한다면 연구자의 단순 노동은 디지털 시대에서 아날로그식 연구를 한다고 볼 수 있다.

연구에 필요한 데이터를 기업의 DB에 접속하여 쉽게 명령어 한 줄로 원하는 데이터를 가져올 수 있는 계정 정보가 있다면 문제가 없겠지만 대부분이 그렇듯이 기업의 보안과 고객 정보 보호 차원에서 기업의 데이터는 보호 받고 있는 것이다. 이러한 데이터는 다양한 특성을 갖고 있다. 먼저 빅데이터로 불리는 자료는 그 크기(Volume)가 GigaByte, TeraByte의 벽을 넘어서 PeataByte, ExaByte, ZetaByte의 양의 데이터 수집을 의미하고 있다. 실제 데이터 크기와 컴퓨터에서 처리할 수 있는 주기억장치의 로딩 가능한 데이터 범위는 큰 차이를 보이고 있다. 대부분의 연구들은 주기억장치의 가용 용량만큼씩 나누어 처리하고 있다. 두 번째 특성은 데이터 크기의 범위를 넘어 방대한 데이터 처리 속도(Velocity)이다. 실제 데이터를 생성하는 네티즌들이 움직이는 순간 관련 콘텐츠가 수집되고, 양질의 정보를 얻기 위한 전처리 과정을 거쳐 분석 결과가 실시간으로 처리됨으로 속도가 매우 빠름을 뜻한다. 물론 이런 배경은 모바일 디바이스의 보편화와 네트워크 속도, 실시간 분석이 가능한 컴퓨터 파워가 한 층 더 높아졌기 때문이다. 과거 대형 컴퓨터나 슈퍼컴퓨터에서 가능했던 것들이 이제는 PC급 서버에서도 실시간 분석이 가능하다는 연구 결과도 나타나고 있다[19]. 세 번째 특성은 다양(Variety)한 데이터들이 컴퓨터에 쌓여가고 있다. 과거 관계형 데이터베이스의 전통적으로 정형화된 데이터 분석은 물론이고 네티즌의 의견을 받은 후기 데이터, 각종 센서에서 발생하는 센싱 데이터, 사진 및 동영상의 화상 데이터 등 비구조적인 비정형데이터가 다양한 분석 방법을 연구하게 만들고 있다[20,21,22].

빅데이터의 특성이 다양한 성질의 대량 데이터를 빠르게 분석할 수 있는 환경이 만들어진 사회 즉, 4차 산업혁명이 데이터로부터 시작된 것이다. 이러한 데이터 분석의 재료인 데이터 수집은 주요한 과제이며 해결하지 않으면 전수 조사가 아닌 샘플 표본에만 의존해야 할 것이다. 많은 기업들이 크롤러(Crawler)를 채용하는 이유이기도 하다.

크롤링은 컴퓨터내의 수많은 문서를 수집하는 기술이다. 웹 환경에서의 각종 정보를 자동 수집하는 기술을 웹 크롤링(Web Crawling)이라 한

다. SNS, 뉴스 등 다양한 미디어들의 콘텐츠 증가로 웹 크롤러의 중요성이 더욱 부각되고 있다[23]. Choi and Kim[24]의 연구는 국내 포털이 제공하는 뉴스 기사를 중심으로 실시간 검색어와의 영향 관계를 분석하였지만 웹 크롤링에 관한 내용은 제외되어 있었다[24]. Kim et al.[25]의 연구에서는 일반적 크롤러와 분산형 크롤러의 구분과 Jsoup트리 분석의 연구를 진행하였으며 크롤링 시스템에 대한 언급 또한 제외된 상태이며 연구 결과만 제시하였다[25]. Park[26]의 연구는 빅데이터 분석 도구로 많은 연구자들이 사용하는 R를 이용하여 8천여 자료 수집 및 데이터 분석 결과를 제시하였으며 명사 추출 함수만 소개하며 크롤링 과정은 생략된 상태이다[26].

본 연구는 SNS 크롤링 시스템을 구축하고자 한다. 오픈소스 소프트웨어인 파이썬(Python)으로 개발된 개발 환경의 소스 코드들을 공개하며 실제 유튜브, 인스타그램, 트위터에 접속하여 실제 자료를 자동으로 크롤링하는 웹 페이지까지 개발하고자 한다.

2.3 파이썬(Python)

2016년 3월, 서울에서 세계 최초 인공지능 프로그램인 구글의 알파고와 최고 바둑 실력자 이세돌과의 세기의 대결이 있었다. 4승 1패로 인공지능 프로그램인 알파고의 승리로 끝나며 인간의 지능을 뛰어 넘은 인공지능의 기술을 경험하였다. 머신러닝과 딥러닝 등 소프트웨어에 대한 관심이 부상하게 되었다. 개발 소프트웨어 도구들은 Java, C, C++, C#, Python, JavaScript, Visual Basic.NET, R, PHP 등 수많은 프로그래밍이 가능한 언어들이 존재한다. 하지만 최근 가장 각광을 받고 있는 언어는 파이썬이다. 2018년 9월에 발표한 킷허브(Github.io) 프로그래밍 언어 인기도(PYPL, Popularity of Programming Language)에 따르면 전 세계적으로 파이썬이 가장 많이 사용되는 언어이며 지난 5년 동안 파이썬이 가장 많이 성장했으며 PHP가 가장 많이 손실되었다고 한다. 실제 지표를 보면 파이썬은 14.5% 성장하였고 PHP는 -6.5% 감소한 것으로 나타났다. 2017년에 비하여 파이썬은 5.7% 성장

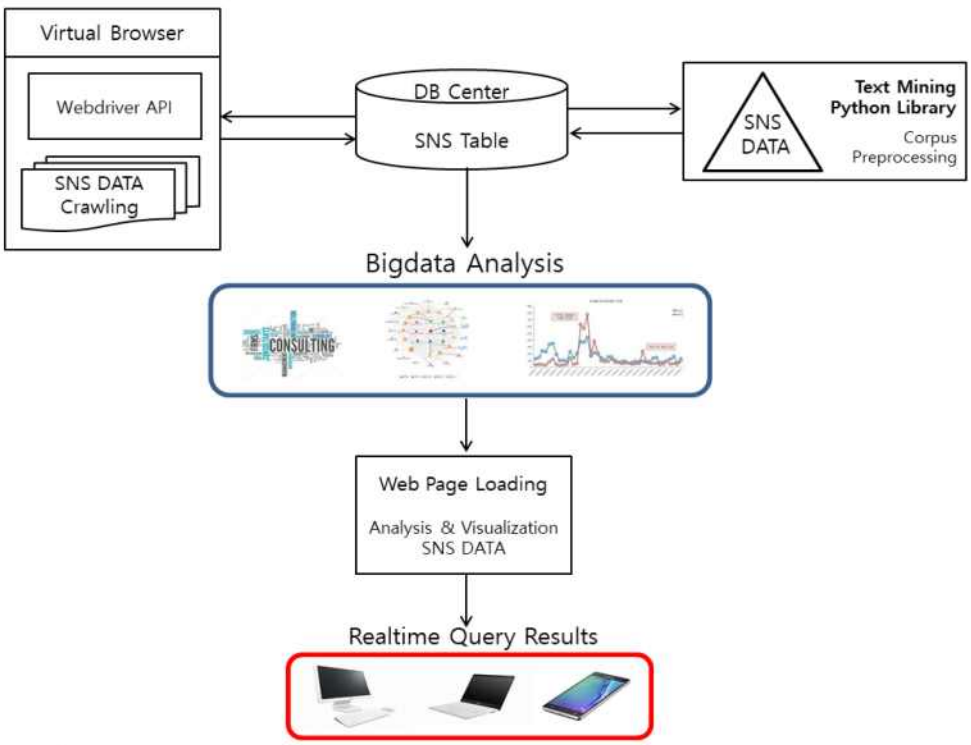

Fig. 2 System Flowchart for SNS Crawling System

하였으며 Java, PHP, VBA, 스칼라, Visual Basic, 델파이 등은 마이너스 성장을 하였다[27]. 파이썬은 C나 JAVA에 비하여 문법 구조가 간결하여 빠르고 쉽게 익힌다는 장점이 있다. 실제 코딩에서 들어쓰기(Indent)를 이용하여 반복문과 조건문등을 구분하여 가독성이 높으며 매우 많은 라이브러리를 제공하여 쉽게 구현이 가능하다.

Lee et al.[28]은 파이썬을 이용한 벡터미적분의 연산과 시각화를 연구하면서 해당 언어의 API를 소개하고 있으며 Yoo[29]는 파이썬을 이용한 초등학생 컴퓨터 사고력을 증진시키기 위한 SW 교육 학습 과정 설계 연구를 하였으며 교육과정 중 인터뷰를 통하여 언어에 대한 긍정, 부정적 반응의 내용을 제시하였다[28,29]. Lee[30]의 연구는 파이썬에서 공개된 API를 정리하였다. Visualization, Modern Analytics, Data Management에 사용 가능한 패키지를 소개하였으며 임베디드에 사용가능한 패키지도 함께 소개하고 있다[30]. 파이썬을 활용하여 빅데이터 연구를 진행한 Park[31]의 연구는 국내 최대 포털 네이버와 페이스북이 제공하는 개발자 API 방법을 소개하였으며, 파이썬을 이용한 크롤링 알고리즘을 일부 공개하였다[31]. 전체적인 시스템 구성을 알아보기 어려운 면이 있으며 대부분의 API 기능이 제한된 것으로 나타났다.

이렇듯 고객 정보 보호와 자사의 보안을 이유로 서비스 초창기에 오픈된 API 사용이 제한을 받게 된다. 본 연구는 현재 연구 단계에서 유튜브, 인스타그램, 트위터의 오픈된 API를 사용하고 크롤링에 필요한 제한된 패키지를 개발하여 실시간으로 SNS 크롤링 시스템을 완성하고자 한다. 또한, 연구 프레임워크를 제시하며 관련 소스 코드 또한 공개하여 이후 연구자에게 제공하자고 한다. 연구결과를 웹페이지에 탑재하여 크롤링 데이터를 다운로드 가능하게 설계하고자 한다.

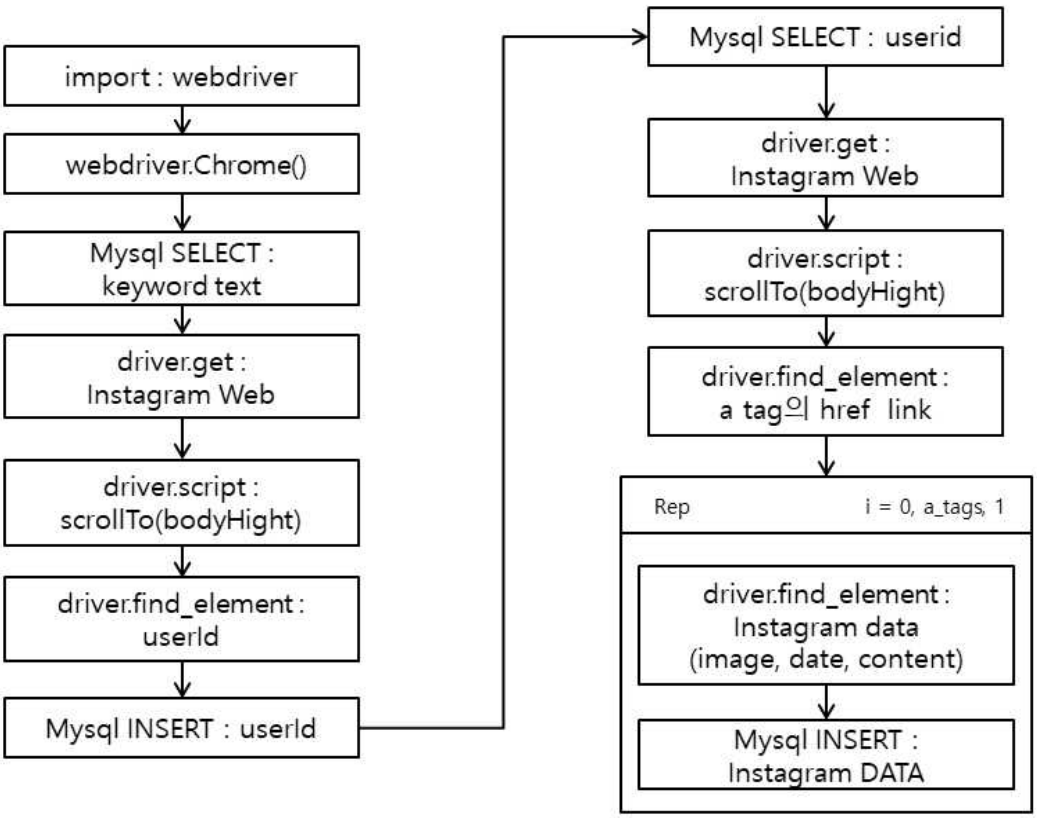

Fig. 3 Instagram Crawling Framework

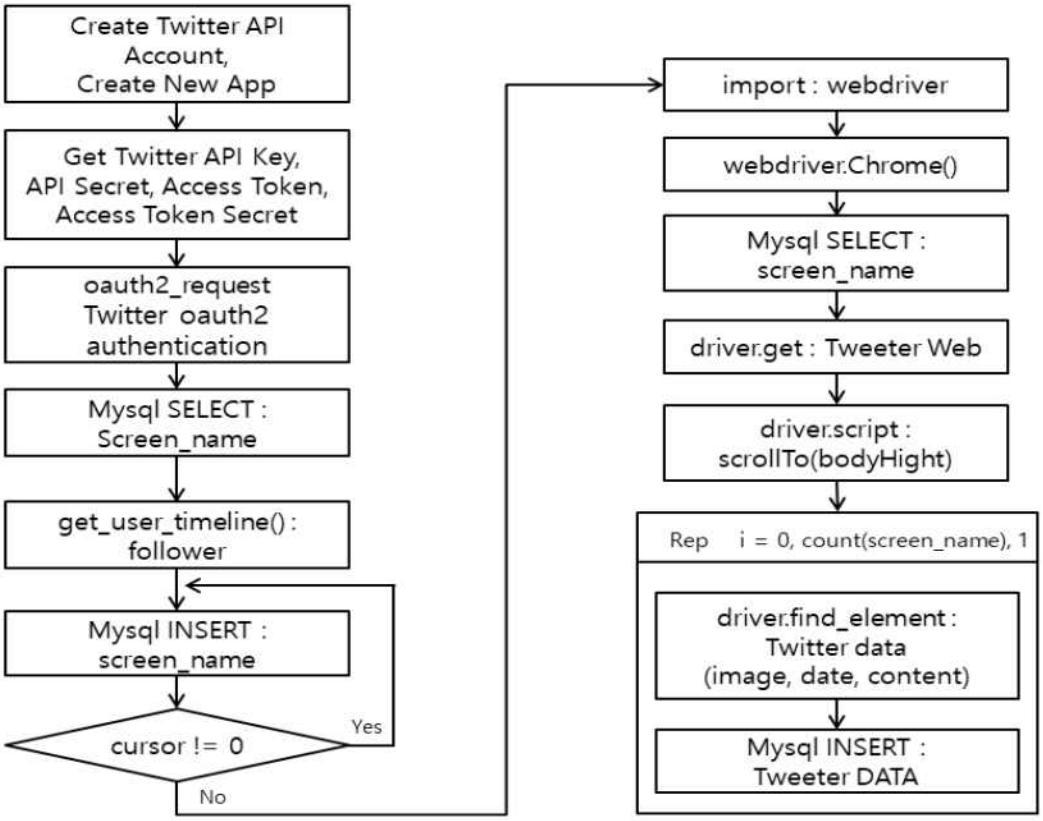

Fig. 4 Twitter Crawling Framework

3. 연구방법과 프레임워크

2017년 인터넷이용실태조사를 한국인터넷진흥원에서 진행하였다. 우리나라 가구당 인터넷 접속률은 전체 1,952만 가구에서 1,943만 가구가 인터넷에 접속하는 것으로 나타났다. 인터넷이용자는 뉴스, 잡지, 정보 검색을 통한 정보 획득 활동의 비중이 가장 높았으며 다음으로 인스턴트메신저 이용과 SNS 이용을 통한 커뮤니케이션 활동이었다[32]. 전 국민의 90.3%가 인터넷 이용률을 보이면서 정보 획득 활동의 인터넷 뉴스 기사의 댓글에 대한 관심도가 높을 수밖에 없을 것이다. 또한 많은 커뮤니케이션 활동으로 인해 소셜 데이터 생산량에도 긍정적 영향을 끼친다. 기업 측면에서 본다면 SNS 이용자는 고객의 니즈분석에 주요한 데이터임엔 틀림없다. 또한 데이터를 기반으로 하는 연구자들에게 연구 데이터 수집에 관한 어려움 및 한계점을 극복할 수 있다. 즉, 소셜 데이터 수집 시스템을 연구함으로써 특정 표본 추출의 통계치를 넘어 모집단 전체를 대상으로 진행되는 연구가 가능할 것으로 본다. 이에 본 연구는 실시간으로 생성되는 웹 상의 SNS 데이터를 자동 수집이 가능한 시스템을 구축하고자 한다.

Fig. 2는 본 연구의 시스템 흐름도이다. 파이썬 웹드라이브 라이브러리를 사용하여 가상 웹브라우저를 이용한 것이다. 실제 웹 사이트에 접속하여 마치 사람이 원하는 데이터를 수집하듯 특정 위치의 데이터를 수집하는 방법이다. 수집된 SNS 데이터는 파이썬 라이브러리의 한글 처리 과정을 거쳐 텍스트마이닝으로 처리되며 실제 필요한 정보만을 DB에 저장하게 된다. 이러한 과정을 수 없이 반복하며 데이터의 크롤링이 진행되며 사용자 인터페이스를 통하여 필요한 데이터를 연구에 활용하게 된다.

Fig. 3은 인스타그램의 크롤링 프레임워크이다.

WebDriver의 목표는 최신 고급 웹 응용 프로그램 테스트 문제에 대한 향상된 지원을 제공하는 잘 디자인 된 객체 지향 API를 제공하는 것으로 Java, C#, Python, Ruby, Perl PHP, JavaScript 등에 개발 환경에 프로젝트를 설정할 수 있다. WebDriver 구동의 테스트는 웹 브라우저에서 가능하며 Firefox Driver, Internet Explorer Driver, chrome Driver, Opera Driver, IOS Driver, Android Driver 등이 지원 가능하다[33]. 본 연구의 인스타그램 크롤링은 Chrome Driver를 사용하였으며 파이썬의 개발 언어를 사용하였다. Fig. 3을 살펴보면 인스타그램 웹 페이지의 검색창에서 특정 해시태그를 입력하면 검색 결과 페이지의 마지막 콘텐츠까지 브라우저 스크롤을 이동시킨다. 각각의 콘텐츠의 계정 정보를 이용하여 url 즉, 연결 링크를 설정하여 개개인의 인스타그램에 접속한다. 최근 생성된 페이지부터 접속하여 이미지, 날짜, 텍스트 등을 크롤링하여 텍스트마이닝 과정을 거쳐 DB에 저장한다.

Fig. 4는 트위터의 크롤링 프레임워크이다. 트위터 크롤링 역시 WebDriver를 이용하여 가상 웹브라우저에서 크롤링되는 방법을 선택하였다. 먼저 접속이 가능한 특정 계정정보를 통하여 접속 인증을 받아 낸다. Screen_name인 ID에 의해 특정 트위터의 Follower 정보를 수집한다. 수집된 Follower ID를 이용하여 Url 생성하며 탐색 페이지 링크를 설정한다. 해당 웹페이지내의 데이터를 탐색하여 관련 내용을 수집하며 텍스트마이닝 처리 후 관련 DB에 저장한다.

Fig. 5는 유튜브의 크롤링 프레임워크이다. 유튜브도 접속을 위한 Secret Key를 발급받아 특정 검색어를 입력하여 관련 동영상 링크를 먼저 크롤링한다. 해당 동영상 식별키인 id를 추출하여 관련 동영상에 접속한다. 콘텐츠의 작성일자, 조회수, 좋아요 수, 싫어요 수, 댓글 수를 추출한다. 추출된 댓글 수만큼의 댓글을 comment Threads() 함수를 이용하여 해당 개수만큼 반복하여 DB에 저장한다.

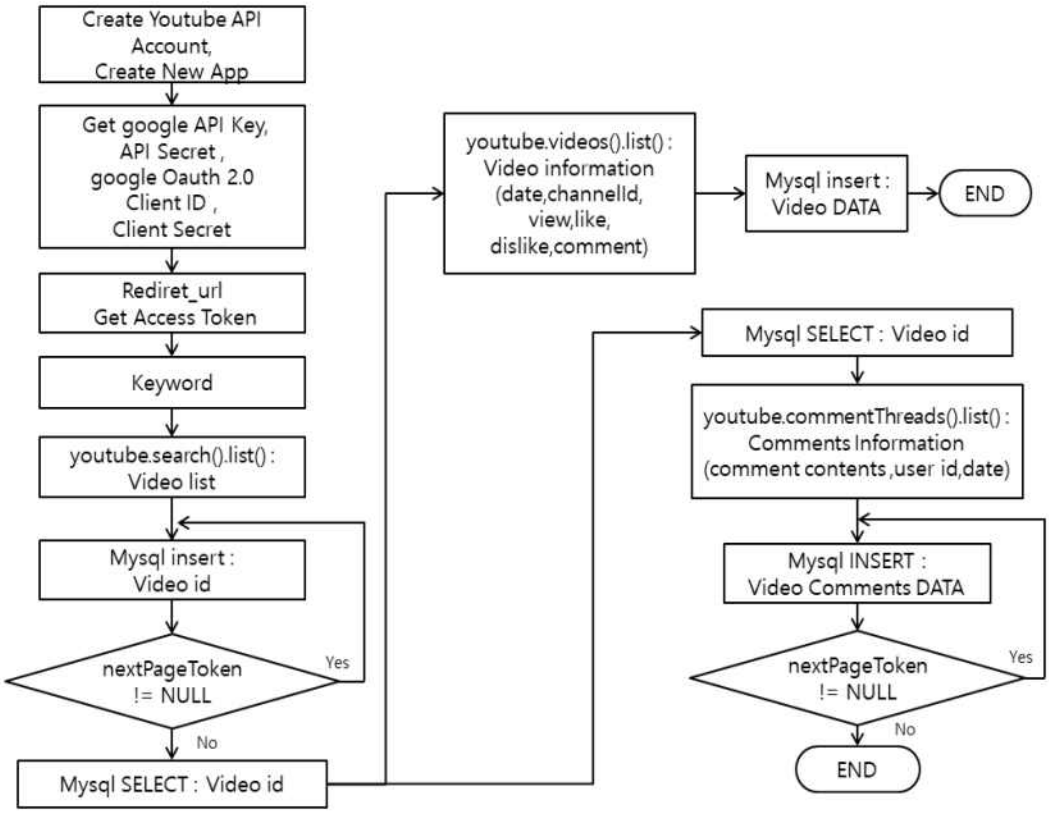

Fig. 5 Youtube Crawling Framework

4. 연구 알고리즘 실험과 결과

본 연구는 현재 전 세계적으로 서비스 중인 SNS를 대상으로 연구자가 원하는 데이터를 자동으로 수집 할 수 있는 시스템 구축하였다. 앞서 제시한 연구 프레임워크와 같이 세 가지 SNS를 대상으로 시스템을 구축하였으며 개발된 패키지 소스와 실제 구동되는 웹 페이지 함께 제시한다.

```
#Import Selenium webdriver library
from selenium import webdriver
#Set virtual browser to chrome
driver = webdriver.Chrome('/opt/google/chrome
        /chromedriver')
#Run query to select Keyword text
sql = "select keyword from instagram"
#Exploring Instagram pages using search keyword
line = "https://www.instagram.com/explore/tags/"+txt2
driver.get(line)
#Page scrollbar down to full
driver.execute_script("window.scrollTo(0, document.body
        .scrollHeight);")
#Crawling the user id of postings that contain search
        terms
userid = driver.find_element_by_css_selector("user_id")
        .text
#Save user id
sql = "insert into instagram set userid= %s"

#Run query to select saved user id
sql = "select no, userid from useridBOX where state
        = %s order by no desc"
#Exploring instagram pages using user id
line = "https://www.instagram.com/"+ userid
driver.get(line)
#Page scrollbar down to full
driver.execute_script("window.scrollTo(0, document
        .body.scrollHeight);")
#Crawls all links to user-generated postings.
allList = find_element_by_tag_name('a').get_attribute
        ("href")
#Use the link to navigate to the page and crawl the
        Instagram data.
for aurl in allList:
  driver.get(aurl)
  #Instagram image
  image =
driver.find_element_by_css_selector("img").get_attribute
        ("src")
  #Instagram date
  date = driver.find_element_by_css_selector("date")
        .get_attribute("datetime")
  #Instagram contents
  content = driver.find_element_by_class_name("content")
        .text
  #Save data
  sql = "insert into instagram set imgae = %s,
        date = %s, content = %s"
```

Fig. 6 Instagram Crawling Algorithm

셀레니움 웹드라이버(Selenium-WebDriver)를 이용하여 크롬 브라우저를 가상 웹브라우저로 설정하였다. 셀레니움 웹드라이버는 페이지 자체가 다시로드 되지 않고 페이지 요소가 변경될 수 있는 동적 웹 페이지를 보다 잘 지원하도록 개발되었다. WebDriver의 목표는 최신 고급 웹 응용 프로그램 테스트 문제에 대한 향상된 지원을 제공하는 잘 디자인 된 객체 지향 API를 제공하는 것이다. 미리 저장된 키워드를 테이블에서 출력하고 그 키워드를 포함한 인스타그램 게시글을 탐색하기 위해 인스타그램 페이지로 이동한다. 처음에는 인스타그램 게시글의 작성자 id만을 크롤링하여 테이블에 저장한다. 작성자 id가 모두 저장되면 이번에는 작성자 id를 이용하여 작성자의 모든 게시글을 크롤링한다. 작성자의 페이지로 이동하여 스크롤을 맨 아래로 내린 후 각 게시글의 이미지, 작성일자, 게시글 내용을 크롤링한 후 저장한다.

```
#Import Twitter oauth2 authentication
import oauth2
def oauth2_request(consumer_key, consumer_secret,
        access_token, access_secret):
  consumer = oauth2.Consumer(key=consumer_key,
        secret=consumer_secret)
```

```
    token = oauth2.Token(key=access_token, secret=access
            _secret)
    client = oauth2.Client(consumer, token)
    return client

def get_user_timeline(client, screen_name, cursor):
    base = "https://api.twitter.com/1.1/followers/list.json"
    fields = "?cursor=%s&screen_name=%s" % (cursor,
             screen_name)
    url = base + fields
    response, data = client.request(url)

#Twitter oauth2 authentication
client = oauth2_request(CONSUMER_KEY,
         CONSUMER_SECRET, ACCESS_TOKEN,
         ACCESS_SECRET)
#Run query to select saved user.
sql = "select screen_name from tweeter"
#Requesting the user's followers list .
cursor = -1
while cursor:
    tweets = get_user_timeline(client, screen_name
             , cursor)
    #Save user's name and user's number of followers
    for tweet in tweets['users']:
        sql = "insert into tweeter set screen_name = %s,
              friends_count = %s"

#Import selenium webdriver library
from selenium import webdriver
#Set virtual browser to chrome
driver =
webdriver.Chrome('/opt/google/chrome/chromedriver')
#Run query to select saved user id
sql = "select screen_name from tweeter"
#Exploring twitter pages using user name
url = 'https://twitter.com/' + username
driver.get(url)
# Page scrollbar down to full and then crawl the twitter data.
driver.execute_script("window.scrollTo(0,
document.body.scrollHeight);")
while content:
    #Twitter contents
    tweets = ele.find_element_by_css_selector
             ('tweeter contents').text
    #Twitter images
    img = ele.find_element_by_css_selector
             ('tweeter image').get_attribute('src')
    #Twitter date
    timedata = ele.find_element_by_css_selector
             ('tweeter date')
    #Save data
    sql = "insert into tweeter set imgsrc = %s,
          date = %s, content=%s"
```

Fig. 7 Twitter Crawling Algorithm

트위터 API 웹페이지에서 회원가입 후 앱을 생성하고, Customer Key와 Customer Secret key, Access Token과 Access Secret key를 발급받는다. 트위터 API 인증을 위해 oauth2 라이브러리를 import 하고, **oauth2_request** 함수를 적용하여 인증한다. 트위터는 사용자 id(screen_name)를 사용하여 **get_user_timeline** 함수를 이용해 cursor값이 0이 아닐때까지 사용자의 follower들의 id를 크롤링하고 저장한다. follower들의 id가 다 저장되면 그 id를 테이블에서 출력한 후 id를 이용하여 사용자 페이지로 이동하여 사용자들의 모든 트윗 정보를 크롤링 한다. 스크롤을 끝까지 내린 후 트위터 내용, 이미지, 작성시간을 크롤링 한 후 테이블에 저장한다.

```
#Import google-API library
from apiclient. discovery import build
#Youtube API authentication
YOUTUBE_API_KEY =          "YOUTUBE_API_KEY"
youtube = build('youtube', 'v3', developerKey =
          YOUTUBE_API_KEY)

#Crawl Youtube video list using keyword text
keytext = "BTS"
nextPT = 1
while nextPT:
    item = youtube.search().list(
        part='snippet',
        q=keytext,
        type='video',
        maxResults=50
    ).execute()
    #Video id
```

```
    videoid = item['id']['videoId']
    #Next page Token
    nextPT = item['nextPageToken']
    #Save video id
    sql = "insert into youtube set videoid = %s,
            keyword = %s"


#Run query to select saved video id
sql = "select videoid from youtube"
#Crawl video information using video id.
item = youtube.videos().list(
    part='snippet,statistics',
    id=videoid,
).execute()
#Video date작성일자
writedate = item['snippet']['publishedAt']
#Video channel id
channel_id = item['snippet']['channelId']
#Video number of view
viewcnt = item['statistics']['viewCount']
#Video number of like
likecnt = item['statistics']['likeCount']
#Video number of dislike
dislikecnt = item['statistics']['dislikeCount']
#Video number of comments
commcnt = item['statistics']['commentCount']
#Save video information
sql = "update youtube set date = %s, channelid = %s,
        view = %s, like=%s, dislike=%s, comm=%s,
        where videoid=%s"

#Run query to select saved video id
sql = "select videoid from youtube"
#Crawl video comments using video id
nextPT = 1
while nextPT:
    item = youtube.commentThreads().list(
        part='snippet,replies',
        videoId=videoid,
        maxResults=100
    ).execute()
    #Save video comments
    sql = "insert into youtube_comm set videoid = %s,
            commt = %s, userid = %s, date = %s"
```

Fig. 8 Youtube Crawling Algorithm

유튜브 API 웹페이지에서 회원가입 후 앱을 생성하고, API key와 API secret key를 발급받고, client id, client secret key를 발급받는다. Redirect_url에 작성한 url에서 Access Token을 획득한다. 하지만 여기서는 공개된 데이터만 크롤링하므로 Access Token은 사용하지 않는다.

유튜브에서는 키워드 관련 동영상 리스트를 크롤링하여 각 동영상의 댓글을 크롤링하였다.

유튜브 API 인증 라이브러리를 import 하고 build 함수를 이용해 인증하였다. 임의의 키워드 단어를 하나 선택하고 **youtube.search().list()** 함수를 이용하여 키워드 검색에 따른 동영상 리스트를 추출하고 각 동영상의 id를 저장하였다. 그리고 동영상의 댓글 정보를 확인하기 위해 동영상의 id와 **youtube.videos().list** 함수를 이용하여 동영상의 작성일자, channel id, 조회수, 좋아요 수, 싫어요 수, 댓글 수 정보를 크롤링하고 테이블에 저장하였다.

그리고 다른 작업으로는 동영상의 id와 **youtube.commentThreads().list** 함수를 이용하여 다음 페이지 토큰(nextPageToken)이 있을 동안 각 동영상의 댓글 내용, 댓글 작성자 id, 댓글 작성일자 등의 정보를 크롤링 하는 것을 반복하고 테이블에 저장하였다.

본 연구 결과는 웹페이지[2]로 게시하였다. Fig. 9는 연구 결과의 검색 웹 페이지로 SNS 종류 선택과 해당 기간을 설정 후 수집을 요하는 키워드를 입력해서 검색버튼을 클릭하면 크롤링이 진행된다. 웹페이지는 PHP와 JavaScript를 이용하여 구현하였으며 연구용 서버에서 파이썬을 이용하여 크롤링이 가능한 시스템이다.

Fig. 10은 날짜의 변화에 따라 검색어 기준 빈도를 시계열로 나타낸 그래프이다. 세 가지 SNS 모두 출력된 화면으로 나타난다. Fig. 9에서 검색어 “추석”이라 넣었을 때 Fig. 10을 살펴보면 트위터는 한가위 명절 전까지 빈도가 증가하면서 명절 기간에는 급격히 낮아진 것을 볼 수 있다. 그와 반대로 인스타그램과 유튜브는 명절 전에 완만한 증가 폭을 보이다가 명절 기간 내 급격히 증가한 것을 볼 수 있다. 이는 SNS별 사용자 특

2) http://222.97.207.181/dashboard/pages/python_crawling.php

Fig. 9 Search Web Page

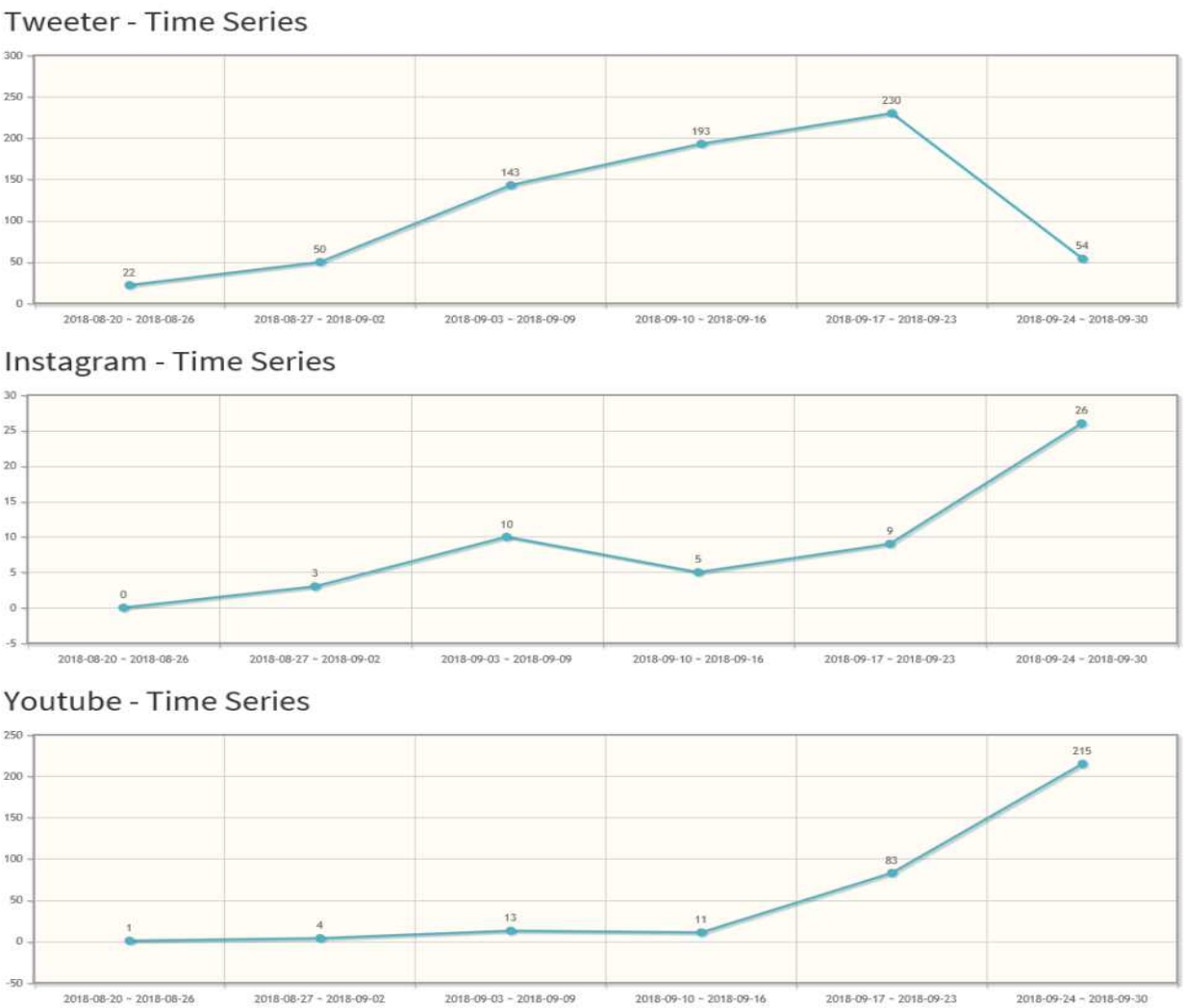

Fig. 10 Time Series Web Page


YOUTUBE DATA

2018-08-20 ~ 2018-08-26 | 2018-08-27 ~ 2018-09-02 | 2018-09-03 ~ 2018-09-09 | 2018-09-10 ~ 2018-09-16

2018-09-17 ~ 2018-09-23 | 2018-09-24 ~ 2018-09-30

2018-09-17, 2018-09-23

no	유튜브	URL	작성일자
1	지지율 추석을앞두고 특별쇼를하는구	https://www.youtube.com/watch?v=Jndc98h5YMA	2018-09-17 16:59:23
2	남북회담 추석이라뭉테기가마때기로들고가겠 카툰회담이네	https://www.youtube.com/watch?v=fYWFWzVWej0	2018-09-17 21:52:26
3	고개 근데 끄덕이는거 뭔가 어색 잘보내 추석 클로징 표정	https://www.youtube.com/watch?v=CzUVBZL1-gU	2018-09-21 14:15:56
4	그림 너뮤 노래 대단 못봐줘서 미앙해 어뜩해 열시 영상 으허어 이쁜걸 죠하 추석잘보내 하는거 활동	https://www.youtube.com/watch?v=C9nGJs9vA-A	2018-09-22 21:23:21

Fig. 11 Data Crawling Web Page

성으로 풀이된다. 2018년 6월 정보통신정책연구원(KISDI)의 보고에 따르면 인스타그램과 유튜브는 10대, 20대, 30대의 사용이 40, 50, 60대 사이에 비하여 많이 사용하는 것으로 조사되었으며 반면, 연령이 높을수록 트위터의 이용률이 높은 것으로 나타났다[34]. 이처럼 추석 연휴 기간 젊은 층들의 SNS 사용이 증가하였고 반면 연령이 높을수록 SNS 사용이 줄어든 것으로 해석할 수 있다.

Fig. 11은 Fig. 9의 검색 웹 페이지의 검색 결과로 유튜브 댓글을 크롤링 한 결과를 나타내고 있다. Fig. 10의 시계열 날짜 구간과 일치하게 설계하였으며 상담의 날짜 구간을 선택하면 구간별 댓글을 확인할 수 있다. 연구용으로 설계된 웹페이지를 감안하여 텍스트 복사가 가능하며 다양한 활용이 이후 연구자에게 도움이 될 것으로 본다.

5. 결론 및 향후 과제

본 연구는 오픈 소스 소프트웨어인 파이썬을 이용한 SNS 크롤링 시스템을 구축하고자 진행되었다. 그 결과와 의미를 종합적으로 정리해 본다.

성별, 연령별 구분 없이 인터넷사용자가 지속적으로 증가하면서 세계적인 빅데이터 시장은 매년 10%이상씩 성장하고 있다. 인터넷사용자들의 직접 생산하는 소셜 데이터도 당연히 증가하고 있다. 이렇듯 정부나 기업 입장에서 고객의 니즈 분석에 SNS 데이터 분석은 필수적인 분석 데이터로 자리잡고 있다. 하지만 분석 보다 선행되어야 하는 데이터 수집은 분석 이상으로 더 중요한 과정이다. 이러한 과정에 필요한 API와 자체 개발된 알고리즘을 함께 공유하였음으로 이후 연구자들에게 많은 도움이 될 것으로 기대된다. 또한, 하나의 SNS가 아닌 인스타그램, 유튜브, 트위터 등 세 가지의 콘텐츠를 동시에 연구하므로 세대별, 성별 등 사용자들의 특징을 연구할 수 있는 환경이 만들어졌다고 본다.

본 연구의 크롤링 시스템의 실무적 관점에서 본다면 트위터는 팔로우 된 사용자를 기준으로 본 연구 시스템이 접속하여 개개인이 작성한 콘텐츠를 크롤링하였으며 인스타그램의 경우 검색어에 따른 게시글 작성자의 ID를 추출하여 각각의 사용자 계정을 활용하여 작성한 게시글 전체를 크롤링하였다. 또한, 유튜브는 검색어에 따른 동영상 콘텐츠 리스트에서 각각의 동영상의 댓글을 크롤링하였으며 검색어 리스트를 별도로 관리되고 있다.

기존 프로그래밍 언어별 공개된 API는 일부 환경에서만 적용되며 실제 연구자 손에 들어오기까지 과정은 그리 쉽지만은 않은 과정이다. 오픈

된 API를 최대한 활용하고 결과를 얻기까지 필요한 알고리즘은 자체 개발하여 연구 크롤링 시스템을 완성하였고 관련 소스 프로그램을 공유하므로 기대효과가 클 것으로 본다.

연구 결과를 명확히 확인하기 위하여 웹 페이지를 설계하였다. 포털 검색 하듯 필요한 키워드와 기간을 설정하면 사용자가 원하는 SNS의 콘텐츠를 쉽게 수집할 수 있다. 수집된 데이터를 세부 기간별로 구분하여 자료의 빈도를 시계열 그래프로 구현하였음으로 쉽게 자료의 양을 알아볼 수 있도록 설계하였다. 또한 실제 결과물인 크롤링 데이터는 시계열 그래프와 같은 세부 기간으로 설정하여 기간별 데이터를 직접 핸들링 할 수 있도록 웹 페이지가 구현되었다.

본 연구의 미진한 부분 또한 앞으로 보완되어야할 과제이다. 먼저 대중적 인기를 얻고 있는 페이스북과 카카오스토리에 대한 크롤링 연구가 더 진행되어야 할 것이다. 페이스북은 최근 개인정보 유출 사건이 빈번히 일어나면서 다양한 방법의 보안과 잦은 웹 구조의 변화로 시스템으로 구축하기에 한계점을 갖고 있는 건 사실이다. 하지만 연구자의 그 동안 연구 경험을 갖고 끊임없는 시간적 투자로 가능할 것으로 본다. 또한 단순 크롤링 된 키워드 빈도만을 이용하여 시계열 그래프를 표현하였다. 다양한 분석 방법을 연구하여 실시간 분석이 가능하다면 보다 많은 연구 기여점을 만들 것으로 본다.

마지막으로 본 연구의 결과 웹페이지는 게재일로부터 1년의 서비스가 진행됨을 알려드리며 연구에 필요한 모든 것을 함께 게재하였으니 참고 바란다.

References

[1] D. Laney, “3D Data Management: Controlling Data Volume, Velocity and Variety,” Gartner, 2001.

[2] Kim, S. H., Chang, S. H., and Lee, S. W., “Consumer Trend Platform Development for Combination Analysis of Structured and Unstructured Big Data,” Journal of Digital Convergence, Vol. 15, No. 6, pp. 133-143, 2017.

[3] Chol, J. H., and Jun, S. H., “Bayesian Inference for Technology Analysis of Artificial Inteilglence,” Journal of Korean Institute of Intelligent Systems, Vol. 28, No. 4, pp. 411-416, 2018.

[4] Wu, Y., Wang, N., Kropczynski, J. and Carroll, J. M., “The Appropriation of GitHub for Curation,” PeerJ Computer Science, Vol. 3, pp. 134, 2017.

[5] Park, O. M., Moon, O. K., Wui, H. K., and Jung, Y. C., “Semantic Web Services Technologies Towards Future Converged Services,.” The Journal of The Korean Institute of Communication Sciences, Vol. 27, No. 5, pp. 30-35, 2010.

[6] D. Fensel, M. Kerrigan, and M. Zaremba, Implementing Semantic Web Services, Springer, 2008.

[7] https://www.kisti.re.kr

[8] https://www.taglive.net

[9] Yoon, Y. K., “A Study on Contents Curation of Portal Sites,” Journal of the Korea Entertainment Industry Association, Vol. 8, No. 4, pp. 31-43, 2014.

[10] Nam, M, J., Lee, E. J., and Shin, J. Y., “A Method for User Sentiment Classification Using Instagram Hashtags,” Journal of Korea Multimedia Society, Vol. 18, No. 11, pp. 1391-1399, 2015.

[11] Min, S. G., and Kim, S. H., “Study on Curation Service Design through Mobile Information Visualization Analysis,” JCD, Vol. 63, pp. 296-305, 2018.

[12] Lee, S. W., Lee, S. M., and Joo, H. M., “A Study of Global Social Network Service Diffusion: An Examination of Facebook Diffusion,” Information Society and Media, Vol. 19, No. 1, pp. 1-22, 2018.

[13] Lee, I. S., Kim, K. K., and Lee, A. R., “A Big Data Analysis Methodology for

Examining Emerging Trend Zones Identified by SNS Users : Focusing on the Spatial Analysis Using Instagram Data," Information Systems Review, Vol. 20, No. 2, pp. 63-85, 2018.

[14] Park, H. W., and Choi, K. H., "Doing Social Big Data Analytics: A Reflection on Research Question, Data Format, and Statistical Test-Convergent Aspects," Journal of Digital Convergence, Vol. 14, No. 12, pp. 591-597, 2016.

[15] Hwang, Y. Y., Lee, K. S., and Choi, S. A., "A Study on the Difference between Young and Old Generation of SNS Behavior," Journal of the Korea Industrial Information Systems Research, Vol. 20, No. 1, pp. 63-77, 2015.

[16] www.jobkorea.co.kr

[17] Kim, H. J., Lee, T. H., Ryu, S. E., and Kim, N. L., "A Study on Text Mining Methods to Analyze Civil Complaints: Structured Association Analysis," Journal of the Korea Industrial Information Systems Research, Vol. 23, No. 3, pp. 13-24, 2018.

[18] 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 Industrial Information Systems Research, Vol. 20, No. 2, pp. 113-124, 2015.

[19] Lee, J. H., and Lee, H. K., "Designing Real-Time Web Mining and Analyzing System," The Journal of Internet Electronic Commerce Resarch, Vol. 18, No. 1, pp. 115-131, 2018.

[20] H. Chen, R. H. L. Chiang, and V. C. Storey, "Business Intelligence and Analytics: From Big Data to Big Impact," MIS Quartly, Vol. 36, No. 4, pp. 1165-1188, 2012.

[21] H. Baars, and H. G. Kemper, "Management Support with Structured and Unstructured Data an Integrated Business Intelligence Framework," Information Systems Management, Vol. 25, No. 2, pp. 132-148, 2008.

[22] A. Gandomi, and M. Haider, "Beyond the Hype: Big Data Concepts, Methods, and Analytics," International Journal of Information Management, Vol. 35, No. 2, pp. 137-144, 2015.

[23] Seo, D. M., and Jung, H. M., "Intelligent Web Crawler for Supporting Big Data Analysis Services," Journal of the Korea Contents Association, Vol. 13, No. 12, pp. 575-584, 2013.

[24] Choi, S. J., and Kim, J. B., "Examine the Relationships Between Portal Article of Naver and Real Time Search Word Using Web Crawling," AJMAHS, Vol. 7, No. 11, pp. 787-794, 2017.

[25] Kim, H. S., Han, N., and Lim, S, J., "Web Crawler Service Implementation for Information Retrieval Based on Big Data Analysis," Journal of Digital Contents Society, Vol. 18, No. 5, pp. 933-942, 2017.

[26] Park, S. J., "A Topic Analysis of SW Education Text Data Using R," Journal of The Korean Association of Information Education, Vol. 19, No. 4, pp. 517-524, 2015.

[27] http://pypl.github.io/PYPL.html

[28] Lee, J, H., Ji, Y. R., and Chae, S. C., "Application of Symbolic Computation, Visualization, and Stochastic Simulation by Using Python in Science Education," School Science Journal, Vol. 12, No. 1, pp. 85-96, 2018.

[29] Yoo, I. H., "The Design of SW Education for Elementary School Using Python and Robots," Journal of The Korean Association of Information Education, Vol. 9, No. 1, pp. 149-155, 2018.

[30] Lee, J. H., "Spyder(Scientific PYthon Development EnviRonment)," The Korean

Institute of Electrical Engineers, Vol. 65, No. 5, pp. 41-48, 2016.
[31] Park, S. J., “A Study on the Utilization of Big Data Using Python,” Journal of Korean Society of Technical Education and Training, Vol. 23, No. 1, pp. 31-40, 2018.
[32] www.kisa.or.kr
[33] www.seleniumhq.org/docs/03_webdriver.jsp
[34] www.kisdi.re.kr

이 종 화 (Jong-Hwa Lee)

- 정회원
- 부경대학교 경영학 석사
- 부경대학교 경영학 박사
- 관심분야 : Big Data, Mining, Content Analysis, Crawling