소셜 네트워크 크롤러 개발을 통한 클러스터링 기반 데이터 통계분석

강병일 황지영†
성균관대학교 컴퓨터공학과
{bi.kang, jjwhang} @skku.edu

Clustering-based Statistical Data Analysis of Real-world Social Networks

Byoungil Kang Joyce Jiyoung Whang†
Department of Computer Science and Engineering, Sungkyunkwan University

요 약

크롤러(crawler)란 웹상에 존재하는 웹페이지를 조직적으로 탐색하고, 정보를 자동으로 검색 및 색인하기 위한 조직적, 자동화된 컴퓨터 시스템을 의미한다. 소셜 네트워크상에서 사용자의 데이터가 무수히 쏟아지고 있는 시점에서 크롤러를 통해 데이터를 수집하고, 통계분석 하는 것은 의미 있는 결과를 도출해 내기에 좋은 방안이라 할 수 있다. 하지만 검색엔진 상에서의 실시간으로 검색되는 키워드를 랭킹 순으로 개별 사용자에게 제공하고 있으나 소셜 네트워크상에서의 이러한 기능은 매우 미비하다. 본 연구에서는 소셜 네트워크 서비스에 존재하는 핵심 키워드를 기준으로 해당 키워드와 연관된 데이터를 자동화하여 추출하는 크롤러를 개발하고, 이를 통해 수집된 데이터를 구조적으로 정형화 한다. 구조화 된 데이터를 유사집단 간 클러스터링(clustering)하고, 각 클러스터 별 데이터 특성을 사용자에게 제공하는 기법을 제안한다.

1. 서 론

소셜 네트워크 서비스를 통하여 수집한 방대한 양의 데이터를 기반으로 분석하는 경우 인간의 행동을 미리 예측할 수 있다. 빅데이터의 핵심은 거대한 양과 다양한 형태 그리고 빠른 생성 속도에서 가치가 있는 데이터를 창출해내는 것이라 할 수 있다.[1] 그 과정에서 사용되는 것을 크롤러(crawler)라고 하며 이는 웹상에 존재하는 많은 수의 웹페이지를 방문하며 각종 정보를 자동적으로 수집해오는 시스템을 의미한다. 본 논문에서는 소셜 네트워크에서 데이터를 추출하는 크롤러를 개발하고 이를 통해 데이터를 수집, 가공하여 원하는 데이터를 만들어 내는 것을 목표로 한다. 타겟으로 하는 소셜 네트워크는 널리 사용하는 인스타그램(Instagram)이며 이 네트워크의 핵심인 해시태그(hash-tag)를 크롤링(crawling)하여 키워드 별 태그를 수집하여 가공한다. 기존 인스타그램의 단일 태그만 검색하여 태그에 해당하는 게시물 정보만 볼 수 있는 점을 개선하기 위하여 크롤링한 게시물 별 해시태그들을 매트릭스(matrix)화 한다. 이후 게시물인 열을 기준으로 태그들의 유무에 따른 태그 간 거리 차이를 이용해 k-means 알고리즘을 적용하여 군집화 한다.[2] 각 군집별로 가장 출현빈도가 높은 태그들을 출력하고 태그들의 경향성을 파악한다. 이를 통해 각 군집별 태그들의 특성을 파악한다. 태그의 경향성을 사용자에 제공하고, 크롤러로 추출해낸 방대한 양의 데이터를 통계내고 분석하는 기법을 제안한다.

† 교신저자 (Corresponding author)

이 논문은 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(2016R1D1A1B03934766, NRF-2010-0020210). 또한, 과학기술정보통신부 및 정보통신기술진흥센터의 SW중심대학지원사업의 연구결과로 수행되었음(2015-0-00914).

2. 개발 환경

2.1 PhantomJS, CasperJS

PhantomJS는 헤드리스 브라우저(headless browser)로 그래픽 유저 인터페이스가 존재하지 않는 웹브라우저이다. 웹킷(WebKit)을 활용하여 사파리(Safari)와 크롬(Chrome)과 유사한 브라우징 환경이다. CasperJS는 이를 활용한 툴로 PhantomJS 라이브러리로 구성된다. CasperJS의 프로그래밍 형식은 크게 3가지 단계로 구성할 수 있으며 각 단계는 시작(start)함수, 단계(then)함수, 실행(run)함수로 구성된다. 시작(start) 함수는 Casper에 대한 환경을 설정하고 주어진 URL에 접속한다. 그 이후 단계(then)함수를 통해 순차적으로 웹 브라우저에서의 행동을 정의한다. 그 후 실행(run)함수를 통해 행동이 정의된 Casper를 실행한다. 본 논문에서는 CasperJS를 사용하여 PhantomJS를 제어하고 PhantomJS를 기반으로 웹 브라우징과 크롤링을 하여 데이터를 수집하는 크롤러를 개발한다.

2.2 JAVA

객체 지향적 프로그래밍 언어로써 플랫폼 독립적인 컴파일러로 자바 언어로 작성된 프로그램을 바이트코드의 바이너리 형태로 변환한다. 바이트코드는 JVM이라는 가상 머신이 필요하며 운영체제 구분 없이 동일한 형태로 실행한다. 본 논문에서는 타 컴파일러와 호환성이 높은 JAVA특징을 이용하여 크롤링한 데이터에 대하여 열은 게시글, 행은 출현한 모든 태그를 속성으로 구조화 한다.

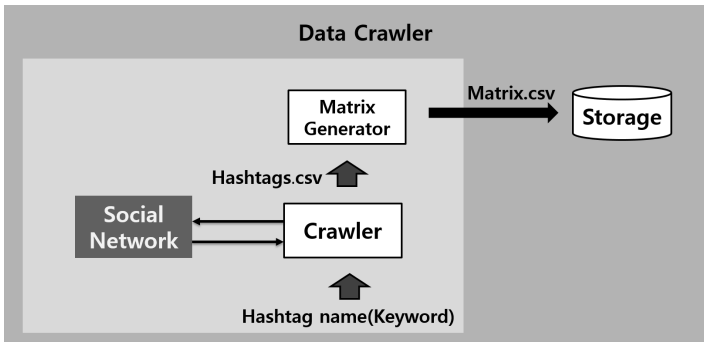
2.3 R studio

R studio는 일반적인 데이터 집합과 행렬이 포함된 매트릭스에 대한 연산과 데이터를 이용한 다양한 시각화가 가능한 유틸리티 세트로서 행렬 구조 및 데이터프레임을 기반으로 둔다. 본 논문에서는 JAVA환경에서 완성한 태그와 게시물의 매트릭스를 R studio 환경에서 k-means 알고리즘을 이용하여 클러스터링하고 전체 및 클러스터별 태 대한 통계치 분석 및 시각화한다.

3. 시스템 모델

3.1 Data Crawling

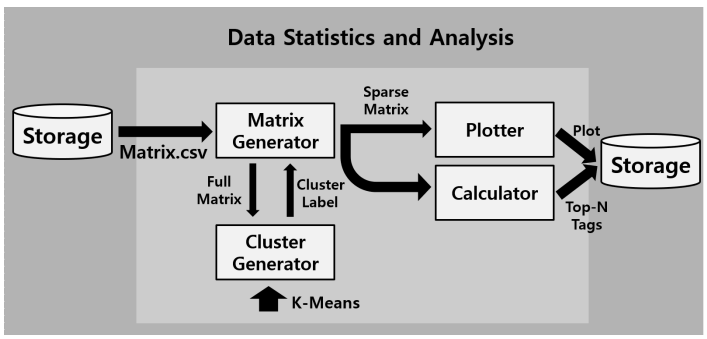
데이터 크롤링을 위한 모델은 <그림 1>과 같다. 크롤러는 사용자로부터 태그 키워드를 입력받는다. 입력받은 크롤러는 인스타그램 상에서의 입력한 키워드와 함께 개제된 태그를 모두 수집하고 이를 열은 게시글 별 개제된 태그로 저장하여 전달한다. 전달받는 매트릭스 구성 컴포넌트는 열은 게시글 행은 함께 개제된 모든 태그로 존재유무를 각 엔트리에 레이블 하여 정형화 한다.



<그림 1. 데이터 크롤러 시스템 모델 >

3.2 Data Statistics and Analysis

클러스터링 기반 데이터 통계분석을 위한 모델은 <그림 2>와 같다. 매트릭스 구성 컴포넌트는 클러스터 구성 컴포넌트로 매트릭스를 전달하면 게시글 간 태그의 존재유무에 따른 거리차이로 k-means 알고리즘을 구현하여 각 게시글을 클러스터링하고 각 게시글 별 클러스터 레이블을 반환한다. 레이블된 매트릭스는 플로터와 계산을 통해 가장 클러스터별 분포를 도식화 하고 전체 및 클러스터 별 가장 빈도수가 많은 Top-N 태그를 빈도수 순으로 출력한다.



<그림 2. 데이터 통계분석 모델 >

3.3 전체 시스템 모델

전체 시스템 구성은 <표 1>과 같다. CasperJS기반 크롤러인 태그 추출 모듈과 JAVA기반 매트릭스 구성 모듈 R 기반 클러스터링 및 클러스터 별 Top-N 랭킹 제공모듈로 구성된다.

<표 1. 전체 모델>

연관태그 추출 모듈 (Crawler)	• 인스타그램에서 키워드와 함께 게시된 모든 태그 수집.
매트릭스 구성 모듈 (Matrix Generator)	• 크롤러로 얻은 해시태그의 Sparse matrix와 Full matrix 표현.
클러스터 별 랭킹 제공 모듈 (Data Statistics and Analysis)	• 매트릭스 구성 모듈로 얻은 Matrix의 Full matrix로 게시물 별 태그 유무에 따른 거리 계산. • K-Means 알고리즘을 이용한 클러스터링. • 전체 및 클러스터별 태그 분포 산점도 표현. • 전체 및 군집 별 태그 빈도수 계산. • 높은 빈도수의 태그 출력 및 태그의 경향성 분석.

4. 크롤러

4.1 인스타그램 DOM 환경

게시물을 저장할 때 사용자가 태그를 입력하는 환경인 인스타그램은 키워드를 통해 해당 키워드를 태그로 하고 있는 모든 게시글들을 가져온다. DOM환경을 실제로 확인하면 이미지와 함께 게시글들이 검색되며 img 엘리먼트로 표현되어 있다. 엘리먼트에 포함되어 있는 내부 속성들은 게시글의 내용을 기록하는 alt와 이미지의 소스를 기록하는 src로 이루어져 있다. 크롤러 개발에 있어 이미지의 소스를 가져오지 않고 게시글의 태그를 가져오는 것을 목적으로 하므로 alt attribute의 태그를 비롯한 텍스트를 가져와 가공을 통해 태그만을 추출한다.

4.2 키워드 게시글 개수 및 전체 게시글 호출

키워드의 게시글 개수 및 전체 게시글을 호출하는 방법은 <표 2>와 같다. 태그를 저장하는 URL에 접속하고 xpath를 이용하여 게시글의 개수를 출력하는 엘리먼트의 텍스트 속성 내용을 가져온다. 페이지의 형태는 가장 상위에 인기 게시글 9개, 그 이후 가장 최신 게시글 9개를 먼저 표시하며 더 많은 게시글을 탐색하기 위해서는 스크롤을 내려 게시글을 추가적으로 불러와야 한다. '더 읽어들이기(Load more)' 버튼으로 정해진 횟수 동안 대기시간을 부여하고 스크롤을 내려 게시글을 호출한다.

<표 2. 게시글 개수 및 전체 게시글 호출 구조 >

```
casper.start 'https://www.instagram.com/explore/tags/' + hashtag + '/'
casper.then function
    numberOfContents = casper.getElementInfo xpathOfElement.text
    casper.echo "게시글의 개수 : "+numberOfContents
casper.then function
    this.wait time, function
        this.capture 'AfterLoad.png'

    this.clickLabel '더 읽어들이기','a'
    var cnt = countRead();
    for each i <=cnt do
        this.wait time, function
            this.echo "scrolled"
            this.scrollToBottom()
    end for
```

4.3 DOM 환경의 게시글 내용 추출

게시글 내용을 추출하기 위해 DOM의 환경에 접속하여 img 엘리먼트의 alt 속성을 호출한다. evaluate함수를 통해 DOM 환경에 함수를 전달하여 alt를 호출하고 Casper로 게시글의 내용을 모두 전달한다.

<표 3. DOM 게시글 내용 추출 구조>

```
casper.then function
    for  each currentLen < 10000 do
        images = this.evaluate function 'number'
                instaImages = document.getElementsByTagName 'img'
                allSrc = []
                for each i < number + 1  do
                        allSrc.push instaImages[i].alt
                return JSON.stringify allSrc
                number : currentLength
                end for
        getTag images
    end for
```

4.4 태그 추출

게시글의 내용 추출한 데이터로부터 해시태그만을 추출한다. 게시글 내용에는 사용자가 작성한 모든 내용을 포함하고 있다. 전체 내용 중 해시태그만을 데이터로 추출한다. 해시태그는 #으로 시작하여 공백 혹은 #까지의 내용으로 substring함수를 사용해 #과 공백문자를 기준으로 단일 태그를 스플릿하여 추출한다.

<표 4. 태그 추출 구조>

```
function getTag images
    str = images.replace /,|₩"|₩.|₩₩n/gi," "
    start , end, strArray [], finish = str.lastIndexOf "#"
    stream = fs.open pathOfFile
    while end < finish do
        start = str.indexOf "#", end
        end = str.indexOf "#", start+1
        if end >= finish then break
        tmp1 = str.indexOf " ",start+1
        if tmp1 < end then end = tmp1
        sentence = str.substring start, end
        len = strArray.push sentence
        start = end
    end while
    Write to File.
```

5. 클러스터링 기반 데이터 통계분석

5.1 매트릭스 구성

크롤러로 추출한 태그 데이터를 게시글 별로 입력받아(게시글 번호, 게시글이 가지고 있는 태그 번호)로 각 태그 별 출현 빈도수가 적고 희박해 시간 복잡성을 고려하여 희소행렬을 만든다.

5.2 클러스터링 및 랭킹

R 환경에서 제공하는 k-means 함수를 사용한다. 게시글의 태그 데이터를 통해 두 개의 집단으로 클러스터링하고 클러스터링 이전의 전체 데이터와 클러스터 별 태그의 빈도수를 계산하여 빈도가 높은 순으로 정렬하여 태그와 빈도를 출력한다.

6. 실험 결과

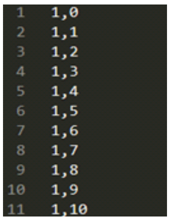
데이터는 특정 키워드를 ‘SKKU’로 지정하고 크롤러로 키워드가 태그된 1100개의 게시글의 모든 태그를 추출하여 사용하였다.

6.1 크롤링

크롤러로 키워드를 입력하여 나온 결과의 일부는 <그림 3-(a)>와 같다. 각 열은 하나의 게시글 기준으로 키워드와 함께 개제된 모든 태그 데이터가 추출된다. <그림3-(b)>은 추출 태그를 희소행렬로 매트릭스 구성한 결과이다.

1차대회	제천	대학배구	성균관대	SKKU	
교육심리	창의력대마	skku	성균관대	집에보내주세요교수님	
대학생	우리	어느새	skku	actart	06학번
NAKPA	SKKU	성균관	첫학회발표		
별승미지	송민두	skku	미친팸	세일	
SKKU	신문방송학	홍대	급		
셀카	셀스타그램	셀피	얼스타그램	취준	취업
seoul	skku	pyeongchan	exchange	henglimyeosaeyo	
skku					
성대스리가	축구	3연승	단체샷	skku	졸잼꿀잼
2017110					
육회	daily	먹스타그램	육회스타그	졸논	실화냐
젊다	젊음이묻음	영원한거젊	강연	팀장회의	

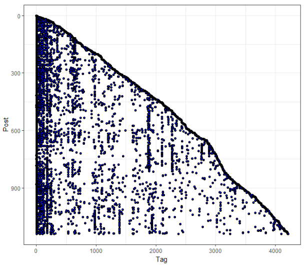
<그림3-(a) 인스타그램 해시태그 크롤링 결과>



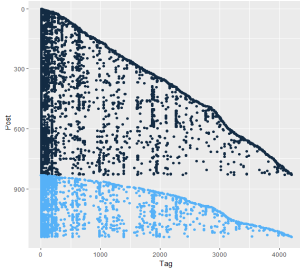
<그림3-(b)희소행렬>

6.2 클러스터링 및 Top-N

희소행렬로 구성된 태그 데이터를 각 게시글 간 태그의 유무를 통하여 2개의 군집으로 k-means 클러스터링 한다. 게시글 별 태그의 유무를 산점도로 표현한 결과가 <그림 4-(a)>, 게시글을 각 군집 별 태그 유무 산점도로 표현한 결과가 <그림 4-(b)> 이고, 전체 및 군집 별 태그의 빈도수를 도출하고 Top-20을 추출한 결과가 <그림 5>이다. 붉은색 태그를 통해 첫 번째 군집과 두 번째 군집의 태그의 경향성에 차이를 보이는 것을 확인할 수 있다.



<그림 4-(a)게시글-태그 산점도>



<그림 4-(b)클러스터별 산점도>

Freq	
1100	SKKU
312	성균관대
159	성균관대학교
140	성대
123	seoul
106	daily
91	korea
80	일상
79	selfie
65	축제
60	감성
58	Fujifilm
56	sungkyunkwan
53	University
49	데일리
48	design
48	졸업
45	southkorea
45	film
44	인물사진

<그림 5-(a)전체>

Freq	
312	성균관대
311	SKKU
79	성대
79	성균관대학교
40	sungkyunkwan
38	University
37	seoul
37	daily
34	일상
32	축제
29	데일리
26	서울
24	혜화
23	sungkyunkwanuniversity
22	korea
22	졸업
22	eskara
21	대학로
19	한국
18	가을

<그림 5-(b)군집1>

Freq	
789	SKKU
86	seoul
80	성균관대학교
69	daily
69	korea
63	selfie
61	성대
59	감성
58	Fujifilm
46	일상
44	design
44	film
44	life
43	인물사진
42	southkorea
42	suwon
42	여행기록
42	여행추억
42	감성사진
42	present

<그림 5-(c)군집2>

7. 결론 및 향후 연구

소셜 네트워크 데이터의 크롤러를 개발하고 이를 클러스터링 하였다. 그리고 전체 및 군집 별 빈도수를 계산해 Top-N 태그를 도출하였다. 유사한 경향성을 갖는 집단끼리의 군집을 통하여 전체보다 군집 별 유의미한 태그 추천이 가능해졌다. 향후 키워드 및 태그의 이음동의어에 대하여 동일 단어로 묶음 처리하여 빈도수를 도출하는 연구를 진행한다.

참 고 문 헌

[1] J. J. Whang, D. F. Gleich, and I. S. Dhillon. “Overlapping Community Detection Using Neighborhood-Inflated Seed Expansion,” *TKDE*, Vol.28, 2016.

[2] J. J. Whang, I. S. Dhillon, and D. F. Gleich. “Non-exhaustive, Overlapping k-means,” *SDM*, pp 936-944, 2015.