KIPS Tr. Software and Data Eng.
Vol.6, No.6 pp.309~314 pISSN: 2287-5905

https://doi.org/10.3745/KTSDE.2017.6.6.309

Storm-Based Dynamic Tag Cloud for Real-Time SNS Data

Siwoon Son[†] · Dasol Kim[††] · Sujeong Lee[†††] · Myeong-Seon Gil[†] · Yang-Sae Moon[††††]

ABSTRACT

In general, there are many difficulties in collecting, storing, and analyzing SNS (social network service) data, since those data have big data characteristics, which occurs very fast with the mixture form of structured and unstructured data. In this paper, we propose a new data visualization framework that works on Apache Storm, and it can be useful for real-time and dynamic analysis of SNS data. Apache Storm is a representative big data software platform that processes and analyzes real-time streaming data in the distributed environment. Using Storm, in this paper we collect and aggregate the real-time Twitter data and dynamically visualize the aggregated results through the tag cloud. In addition to Storm-based collection and aggregation functionalities, we also design and implement a Web interface that a user gives his/her interesting keywords and confirms the visualization result of tag cloud related to the given keywords. We finally empirically show that this study makes users be able to intuitively figure out the change of the interested subject on SNS data and the visualized results be applied to many other services such as thematic trend analysis, product recommendation, and customer needs identification.

Keywords : Dynamic Visualization, Big Data, Real-Time Processing, SNS Analysis, Tag Cloud

실시간 SNS 데이터를 위한 Storm 기반 동적 태그 클라우드

손 시 운[†] · 김 다 솔[††] · 이 수 정[†††] · 길 명 선[†] · 문 양 세[††††]

요 약

일반적으로 SNS (social network service) 데이터는 정형, 비정형 데이터가 섞여 빠르게 생성되는 빅데이터의 특성을 갖기 때문에 실시간 수집/저장/분석에 많은 어려움이 있다. 본 논문에서는 이러한 SNS 데이터의 분석에 활용할 수 있는 Apache Storm 기반 실시간 동적 데이터 시각화 기술을 제안한다. Storm은 대표적인 빅데이터 기술 중 하나로, 실시간으로 수집되는 데이터를 분산 환경에서 처리 및 분석하는 소프트웨어 플랫폼이다. 본 논문은 Storm을 사용하여 빠르게 발생하는 트위터(Twitter) 데이터를 수집 및 집계하고, 태그 클라우드를 통해 그 결과를 동적으로 표현하고자 한다. 이를 위해, 사용자가 요구하는 키워드를 입력받고 해당 키워드를 통한 시각화 결과를 실시간으로 확인할 수 있는 웹 인터페이스를 설계 및 구현한다. 또한, 각각의 태그 클라우드 결과를 비교하여 올바로 시각화되었는지 확인한다. 본 연구를 통해, 사용자는 관심있는 주제가 SNS에서 어떻게 변화하고 있는지 직관적으로 판단할 수 있게 되며, 시각화 결과는 주제별 트렌드 분석, 고객 니즈 파악 등 다른 서비스에도 활용이 가능하다.

키워드 : 동적 시각화, 빅데이터, 실시간 처리, SNS 분석, 태그 클라우드

1. 서 론

2000년대 후반 이후 각종 스마트기기의 대중화에 따라 소셜 네트워크 서비스(social networking service: SNS)[1]의 사용이 급증하였다. SNS는 사용자 간의 의사 표현, 정보 공유, 친목 도모 등을 목적으로 하며, 대표적으로는 페이스북(Facebook), 트위터(Twitter), 인스타그램(Instagram) 등이 있다. SNS 사용자들은 대부분 텍스트와 함께 사진, 영상, 링크 등 여러 타입의 데이터를 혼합하여 자신의 의견을 게시한다. 이러한 게시글은 전세계적으로 빠르게 생성되며, 해당 데이터는 연간 수십 엑사바이트(exabyte) 이상의 대용량이 된다. 예를 들어, 페이스북은 매일 약 25억 개의 컨텐츠가, 트위터는 매초마다 약 5,700개의 컨텐츠가 공유된다[2]. 이렇게 생성되는 SNS 데이터는 데이터의 다양성(variety), 빠른 생성속도(velocity), 대용량(volume)의 특징을 갖기 때문에 빅데이터[3]로 구분할 수 있다. 특히, 대다수의 사람들은 의사 표현을 위한 도구로 SNS를 많이 사용하기 때문에, SNS 데

※ 이 논문은 2017년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임(No.R7117-17-0214, 데이터 스트림 정제를 위한 지능형 샘플링 및 필터링 기술 개발).
† 준 회 원 : 강원대학교 컴퓨터과학과 박사과정
†† 준 회 원 : 강원대학교 컴퓨터과학과 학사과정
††† 비 회 원 : 강원대학교 컴퓨터과학과 학사과정
†††† 종신회원 : 강원대학교 컴퓨터과학과 교수
Manuscript Received : November 10, 2016
Accepted : December 1, 2016
* Corresponding Author : Yang-Sae Moon(ysmoon@kangwon.ac.kr)

이터에는 특정 시점의 이슈, 특정 주제에 대한 사람들의 반응이나 의견이 쉽게 드러난다. 따라서, 이를 활용한 분석 및 시각화 결과는 이슈 변화, 상품 트렌드, 고객 니즈 등을 쉽게 파악하는데 효과적이다.

본 논문에서는 SNS 데이터의 시각화에 중점을 둔다. 빅데이터의 시각화 기술에는 데이터를 수치적으로 표현하는 단순 차트 및 그래프뿐만 아니라, 개체들의 관계를 나타내는 다이어그램이나 트리, 분포를 알 수 있는 히트맵 등이 있다. 이 중, 태그 클라우드[4]는 단어들을 나열하여 구름 형태로 시각화하는 기술로, 단어의 빈도수에 따라 색이나 크기를 다르게 표현하기 때문에 비중 있는 단어를 좀 더 직관적으로 파악할 수 있다. Fig. 1은 태그 클라우드의 대표적인 사례를 나타낸다[5]. 일반적인 태그 클라우드 기술은 고정된 문서 집합에서 단어 수를 계산하여 시각화해주는 방식이다. 그러나, 이러한 방법으로는 실시간으로 생성되는 SNS 데이터의 변화를 즉각적으로 표현하는데 제약이 따른다.

Fig. 1. A Representative Example of Tag Cloud

SNS 데이터의 변화를 직관적으로 표현하기 위해, 본 연구에서는 실시간 데이터 처리 환경을 활용한 동적 시각화 기술을 제안한다. 본 논문에서는 여러 SNS 중 트위터 데이터를 대상으로 시각화를 수행하며, 해당 데이터의 수집·분석을 위해 Apache Storm[6, 7] 기반 구현 및 실험 환경을 구축하였다. Storm은 다수의 서버로 구성된 클러스터에서 실시간으로 수집되는 데이터를 분산 처리하는 오픈소스 플랫폼이다. 특히, Storm은 기존 Hadoop [8]의 맵리듀스(MapReduce)에 비해 데이터 수집·처리 속도가 빠르기 때문에 트위터, 야후, 바이두 등에서 빅데이터 스트림 처리에 활용하고 있다[6]. Storm과 더불어, 전체 시스템의 사용 편의를 위해 시각화 주제 입력 및 분석 결과 확인을 위한 웹 인터페이스를 설계 및 구현하였다. 이를 통해 사용자는 시각화 주제에 따라 실시간으로 변화하는 동적 태그 클라우드를 확인할 수 있다.

본 논문에서 제안하는 시스템은 크게 세 단계로 동작한다. 먼저, (1) 데이터 실시간 수집 단계는 트위터로부터 사용자 키워드를 포함하는 트윗을 실시간으로 수집한다. 다음으로, (2) 주제별 해시태그 집계 단계는 Storm을 통해 수집한 데이터에서 해시태그(hashtag)를 추출하고 그 수를 집계한다. 마지막 단계는, 계산된 해시태그 집계 결과를 태그 클라우드에 실시간으로 반영하여 시각화하는 (3) 실시간 동적 시각화 단계이다. 각 단계의 동작 방식은 제3장에서 자세히 설명한다.

제안 시스템은 기존 시각화 기술로는 표현이 어려운 비정형 데이터의 변화를 실시간으로 확인할 수 있다는 장점을 갖는다. 특히, 트위터 데이터의 실시간 수집/처리를 위해 Storm의 기능적 특성을 최대한 활용하여 분석 환경을 구축하였다. 또한, 기존 태그 클라우드의 한계점을 개선하여 실시간, 동적 태그 클라우드 시각화 방법을 새롭게 제시하였다. 본 연구 결과는 실시간으로 변화하는 사회 이슈 파악, 특정 주제에 대한 반응 및 의견 분석, 예측하지 못한 돌발적 상황에 대한 정보 수집 등 여러가지 동적 분석 환경에 활용 가능하다.

본 논문의 구성은 다음과 같다. 제2장은 본 연구와 관련된 주요 기술들을 설명한다. 제3장은 본 논문이 제안하는 동적 태그 클라우드 기술의 전체 시스템 구조도를 설명한다. 제4장에서는 해당 시스템의 구현 결과를 평가한다. 마지막으로 제5장에서 연구의 결론을 맺는다.

2. 관련 연구

2.1 Apache Storm

실시간 데이터 처리 기술은 실시간으로 발생하는 스트림 데이터를 수집 즉시 처리하는 기술이며, 대표적으로 Apache Storm, Apache Spark [9], EsperTech Esper [10] 등이 있다. 이 중 Storm은 분산된 다수의 서버에서 빠르게 데이터를 처리하는 실시간 분산 처리 기술이며, 필요에 따라 클러스터를 확장(scale-out) 가능하다는 장점이 있다. Storm은 Apache Hadoop [8]과 유사한 마스터/슬레이브 구조를 갖는다. Fig. 2는 이와 같은 Storm의 동작 구조를 나타낸다. 그림에서, 님버스(Nimbus)는 마스터, 슈퍼바이저(Supervisor)는 슬레이브에 해당하며, 이들은 주키퍼(Zookeeper)를 통해 관리가 된다. 님버스는 슬레이브 역할을 수행하는 슈퍼바이저에 작업을 배포/할당하며, 주기적으로 슈퍼바이저의 상태를 파악하고 장애에 대응한다. 슈퍼바이저는 님버스가 할당한 작업을 직접 수행하며 님버스에 상태를 보고한다.

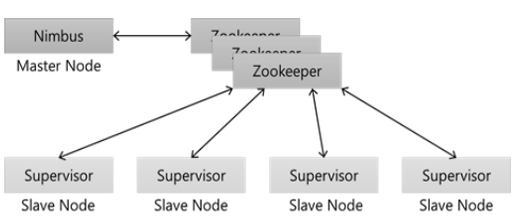


Fig. 2. An Architecture of Apache Storm

Storm은 작업을 처리할 때 데이터의 입력부터 출력까지 일련의 작업을 토폴로지(Topology)로 표현한다. 토폴로지는 스파우트(Spout)와 볼트(Bolt)로 구성된다. 먼저, 스파우트는 스트리밍 데이터의 발생지로부터 데이터를 입력 받아 Storm에서 사용하는 데이터 형태인 튜플(Tuple)로 바꾸어 볼트에게 전달한다. 다음으로, 볼트는 스파우트로부터 전달 받은 데이터를 처리하여 다음 볼트로 전달하거나, 결과를 출력 또는 저장 장치로 보낸다. 일반적으로 Storm은 튜플 단위로 작업을 처리[7]하는데, 이 방식으로는 집계나 그룹핑 등의 연산을 처리하기

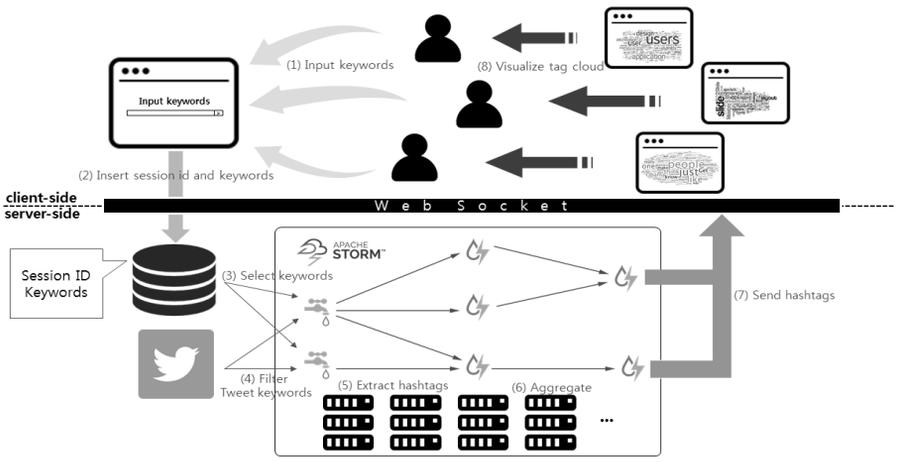


Fig. 3. Overall Architecture of Dynamic Tag Cloud System

어렵다. 이를 해결하기 위해 Storm은 고수준 추상화 기술인 트라이덴트(Trident)[11]를 제공한다. 트라이덴트는 튜플을 일정량 모아 배치로 처리하는 방식을 사용하며, 이를 통해 JOIN, AGGREGATION, GROUPING, FUNCTION, FILTER 등의 연산을 수행할 수 있다. 본 논문에서는 배치 단위로 단어의 수를 계산하기 위해 튜플 단위 연산이 아닌 트라이덴트를 사용한다.

2.2 태그 클라우드

다양한 시각화 기술 중에서도 태그 클라우드 기술은 문서 집합으로부터 특정 단어의 빈도수를 시각적으로 표현하는데 매우 유용하다. 초기의 태그 클라우드 연구는 문서 집합에 존재하는 단어들을 발생순, 단어순, 빈도순 등의 순서로 나열하고 빈도 수에 따라 단어의 크기를 다르게 나타내었다[12, 13]. 그러나, 이러한 방법은 빈도에 상관없이 모든 단어가 같은 공간을 차지하며, 단어가 많아짐에 따라 빈 공간이 넓어져 높은 빈도를 갖는 단어의 직관성이 낮아진다. 이를 개선하기 위해 전체 공간을 단어의 빈도 수에 따라 나누어 태그 클라우드를 표현하는 연구[14]가 수행되었으나, 모든 단어의 방향이 수평형으로 같기 때문에 단어 사이의 공백이 크다는 단점이 있다.

최근 웹을 기반으로 데이터 시각화를 지원하기 위해 다양한 자바스크립트 라이브러리가 등장했다. 특히, D3.js[15]에서 제공하는 태그 클라우드는 단어 사이의 공백에 맞게 단어를 회전하여 태그 클라우드를 표현한다. 이 방법을 이용하면 단어가 서로 겹치지 않으면서 단어 사이의 공백을 최소화할 수 있다. 이 같은 장점에 따라 본 논문에서도 이러한 D3.js의 태그 클라우드 라이브러리를 사용한다.

3. 동적 태그 클라우드 시스템의 설계 및 구현

본 절에서는 제안하는 동적 태그 클라우드 시스템을 제시한다. 먼저 제3.1절에서는 전체 시스템 구조도에 설명하고, 제3.2절에서는 시스템의 핵심이 되는 Storm 토폴로지를, 제3.3절에서는 시각화를 위한 웹 인터페이스와 데이터베이스 스키마를 설계한다.

3.1 동적 태그 클라우드 시스템의 전체 구조

Fig. 3은 본 논문에서 제안하는 동적 태그 클라우드 시스템의 구조도이다. 시스템은 가운데의 웹 소켓(Web socket)을 기준으로 위는 클라이언트, 아래는 서버의 형태로 구성되어 있다. 이와 같이 클라이언트-서버 구조를 채택한 이유는 다양한 키워드를 서로 다른 사용자로부터 받아 Storm 클러스터에서 분석하고, 각 사용자에게 결과를 제공하기 위함이다. 기본적으로 사용자는 자신이 원하는 키워드를 입력하고, 그 결과를 실시간으로 변하는 태그 클라우드로 제공받게 된다.

Fig. 3의 동작 절차를 사용자 중심으로 자세히 설명하면 다음과 같다.

(1) Input keywords: 시스템의 사용자가 키워드를 입력한다. 입력한 키워드는 트위터 스트리밍 데이터 중에서 이 키워드를 포함하는 트윗을 필터링하는 목적으로 사용된다.
(2) Insert session id and keywords: 사용자가 하나 이상의 키워드를 입력하여 검색 버튼을 클릭하면 사용자의 세션 ID와 키워드들이 서버에 전송된다. 세션 ID는 추후 시각화에서 사용자를 구분하는데 사용된다. 웹 소켓 서버는 전달받은 세션 ID와 키워드들을 데이터베이스에 저장한다.
(3) Select keywords: 스파우트는 폴링(polling) 방식으로 데이터베이스에서 주기적으로 모든 키워드들을 검색한다. 이로써 사용자의 입력과 토폴로지는 비동기적으로 동작한다.
(4) Filter Tweet keywords: 스파우트는 Twitter4J[16]를 통해 키워드들을 필터링한다. 즉, 트위터로부터 키워드가 포함된 트윗만을 실시간으로 수집한다.
(5) Extract hashtags: 수집한 트윗으로부터 해시태그를 추출한다.
(6) Aggregate: 해시태그의 수를 집계한다.
(7) Send hashtags: 집계한 해시태그를 웹 서버로 전달한다.
(8) Visualize tag cloud: 세션 ID를 사용하여 사용자를 구분하고, 각 사용자에게 태그 클라우드를 시각화하여 제공한다.

Fig. 3의 시스템 구조에서의 핵심은 태그를 집계하는 부분과 이를 시각화하는 부분이다. 따라서, 제3.2절은 트윗을 수집하고 해시태그를 집계하는 Storm 토폴로지를 설명한다.

그리고, 제3.3절은 사용자 관리 및 태그 클라우드의 시각화를 위한 웹 서버 및 데이터베이스를 설명한다.

3.2 해시태그 집계를 위한 Storm 토폴로지

본 논문에서는 일반적인 Storm이 아닌 트라이덴트를 사용하였다. 이는 제안하는 시스템에서 수집한 해시태그들을 일정 시간마다 배치 형태로 집계하기 위함이다. Fig. 4는 이러한 트라이덴트 토폴로지의 설계이다. 토폴로지는 데이터베이스에 저장된 사용자들의 키워드로 트윗을 필터링하고, 각 배치 별로 트윗의 해시태그를 집계한다.

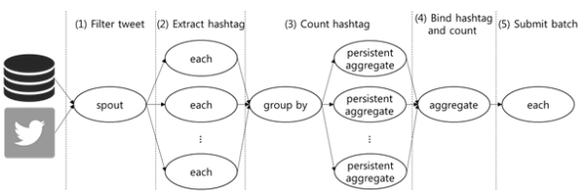


Fig. 4. Storm Trident Topology for Aggregating Hashtag

Fig. 4의 각 단계에 대한 자세한 설명은 다음과 같다.

(1) Filter tweet: 스파우트는 트위터로부터 데이터베이스에 저장된 키워드들이 포함된 트윗들을 필터링한다. 이때, 일정 기간의 배치 단위로 데이터를 집계할 수 있도록 기간 내에 발생한 데이터는 동일한 배치 ID를 부여한다.
(2) Extract hashtag: 스파우트로부터 필터링된 트윗에서 해시태그를 추출한다. 하나의 트윗에는 하나 이상의 해시태그가 포함될 수 있으며, 배치 ID, 해시태그, 트윗에 포함된 키워드의 리스트를 다음 트윗으로 전달한다.
(3) Count hashtag: 배치 ID 및 해시태그를 키(key)로 그룹핑하고, 각 키에 대한 집계를 수행한다. 이 결과로 같은 배치 ID 내에서 해시태그와 그 수가 계산된다.
(4) Bind hashtag and count: 배치 ID 내의 모든 해시태그들을 집계한다.
(5) Submit batch: 앞서 집계한 배치를 웹 소켓을 통해 웹 서버로 전달한다. 이때 전달하는 데이터의 형태는 JSON을 사용하였다.

앞서 토폴로지 구성 단계 중 Filter tweet 단계에서는 필터링된 트윗이 어떤 키워드로 필터링이 되었는지 알아야 한다. 이는 다수의 사용자 지원을 위한 것으로, 사용자 기준으로 입력 키워드를 구분하고, 각 키워드별로 집계된 결과를 해당 사용자에게 올바르게 전달하기 위함이다. 그러나, 기존 트위터 API는 키워드 재추출 기능을 제공하지 않으므로 트윗의 재탐색이 필요하며, 이를 위해 본 논문에서는 스파우트에 관련 기능을 추가하였다. Fig. 5는 Filter tweet 단계의 스파우트의 알고리즘이며, 실제 스파우트 내의 `emitBatch()` 메서드에 구현하였다. 먼저, 일정 기간의 배치를 위해 라인 2에서 시작 시간을 초기화하고, 라인 3에서 현재 시간과 시작 시간의 차가 배치 시간을 초과할 경우 반복을 종료한다. 다음으로, 라인 4에서 키워드를 기반으로 트윗을 필터링하고, 라인 5~9에서 트윗을 재탐색하여 트윗에 포함된 키워드를 다시 추출한다. 마지막으로, 라인 10은 배치 ID, 트윗, 트윗에 포함된 키워드의 리스트를 튜플로 묶어 볼트에게 전달한다. 이 알고리즘은 Storm 실행 시 끊임없이 반복되어 실시간으로 배치 데이터를 생성하며, 이후 단계에서 이를 처리하여 시각화를 수행한다.

```
Procedure FilterTwitSpout()
Input:
   BID: a batch id.
   BT: a batch time.
   KWs: a list of keywords selected from database.
   Q: a queue connected with Twitter.
Output:
   BID: a batch id.
   T: a filtered tweet.
   CKWs: the keywords list contained in T.
1. begin
2.    start := current_time();
3.    while current_time() − start < BT do
4.       T := pop(Q, KWs);
5.       for each word k of KWs do
6.          if (T contains k) then
7.             add(CKWs, k);
8.          end-if
9.       end-for
10.      emit_tuple(BID, T, CKWs);
11.   end-while
12. end-procedure
```

Fig. 5. An Algorithm of Trident Spout for Twit Filtering

기존 태그 클라우드 기술은 문서 내의 단어, 각 단어의 빈도수를 사용하여 시각화를 수행한다. 본 논문에서는 Storm을 통해 지속적으로 배치 데이터의 빈도를 계산하고, 기존 태그 클라우드의 단어와 빈도수에 최신 결과를 반영하여 태그 클라우드 수정한다. 이 방법을 통해 태그 클라우드는 일정 시간마다 동적으로 바뀌게 된다.

4. 동적 태그 클라우드 시스템 평가

본 절에서는 앞서 제3장에서 설계한 동적 태그 클라우드 시스템을 구현하고 결과를 확인한다. 본 논문은 1대의 님버스 서버와 8대의 슈퍼바이저 서버로 구성된 Storm 클러스터와 1대의 웹·데이터베이스 서버를 시스템 구현에 사용하였다. 각 서버의 상세 사양은 Table 1과 같다.

Table 1. The Specifications of Dynamic Tag Cloud System

Part	Specification
Hardware	• Nimbus server: Intel Xeon 2.4GHz 8 Core, 16GB RAM • Supervisor server: Intel Xeon 2.4GHz 6 Core, 16GB RAM • Web and Database server: Intel Xeon 3.1GHz 4 Core, 8GB RAM
Software	• Operating system: CentOS 7 64bit • Processing environment: Apache Storm 1.0.2 • Web server and Database: Apache Tomcat 7, MariaDB 5.5

사용자 인터페이스는 두 개의 웹 페이지로 구성하였다. 첫 번째 페이지는 사용자로부터 키워드를 입력 받으며, 두 번째 페이지는 Storm 토폴로지로부터 주기적으로 단어와 그 수를 받아 태그 클라우드로 시각화한다. Fig. 6은 키워드

를 입력 받는 첫 번째 웹 페이지의 예시 화면이다. 키워드는 탭(tab) 또는 콤마(,) 키로 구분하여 입력하면 리스트에 추가되며, 이는 공백으로 구분된 여러 단어를 하나의 키워드로 구분할 수 있도록 한다. 사용자가 키워드를 모두 입력하면 검색(search) 버튼을 눌러 다음 페이지로 이동할 수 있다. 검색 버튼을 클릭하면 키워드 리스트가 서버에 전송되며, Storm 토폴로지가 주기적으로 처리한 배치를 두 번째 페이지에서 시각화한다. 본 논문에서는 5초 단위로 배치를 처리하도록 스파우트의 배치 시간을 설정하였다.

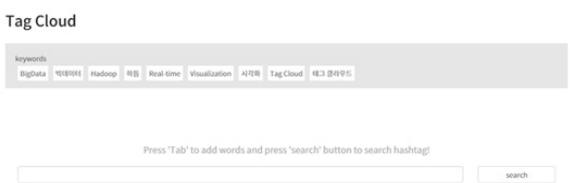
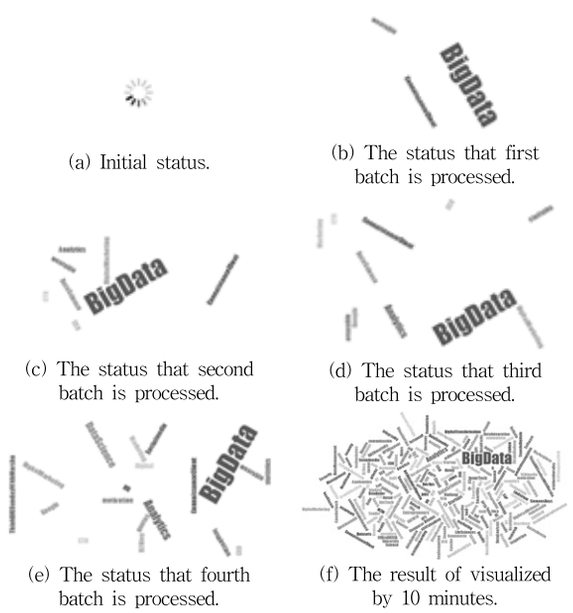


Fig. 6. Web Page to Input Keywords

Fig. 7은 시각화를 위한 결과 페이지로, 예시를 위해 사용한 키워드는 "*BigData*, *빅데이터*, *Hadoop*, *하둡*, *Real-time*, *Visualization*, *시각화*, *Tag Cloud*, *태그 클라우드*"이다. 앞서 언급한 바와 같이, 본 논문에서는 태그 클라우드 시각화를 위해 D3.js 라이브러리를 사용한다. Fig. 7(a)는 두 번째 페이지의 초기 상태이며 아직 처리 결과가 없어 태그 클라우드가 표현되지 않았다. 다음으로, Fig. 7(b)는 Fig. 7(a)에서 5초가 지나 첫 번째 배치 데이터가 처리되어 태그 클라우드가 표현된 것이다. 첫 번째 배치 데이터에는 총 3 개의 단어가 입력되었으며, 단어 "*BigData*"는 다른 단어에 비해 가중치가 높아 더 큰 글자로 표현되었다. 그리고, Fig. 7(c)는 Fig. 7(b)에서 5초가 지나 두 번째 배치 데이터가 처리되어 태그 클라우드가 변경된 것으로, 두 번째 처리에서는 5 개의 단어가 추가



(a) Initial status. (b) The status that first batch is processed.

(c) The status that second batch is processed. (d) The status that third batch is processed.

(e) The status that fourth batch is processed. (f) The result of visualized by 10 minutes.

Fig. 7. Web Page to Visualize Tag Cloud

로 입력되었다. Fig. 7(d)와 7(e)는 계속해서 데이터가 처리되어 더 많은 단어가 시각화되었고, 각각 단어 "*Analytics*"가 추가로 입력되어 더 큰 글자로 표현되었다. 마지막으로 Fig. 7(f)는 10분의 시간 동안 시각화된 태그 클라우드이다. 이 중 단어 "*BigData*"의 가중치가 가장 높기 때문에 가장 큰 글자로 표현되었으며, 계속해서 배치가 처리되므로 이는 변경될 수 있다. Fig. 7은 5초 단위의 수집이나 이는 사용자가 1초 단위 혹은 그보다 작은 시간으로 설정이 가능하다.

Fig. 8는 다른 키워드를 사용하여 20분 간 시각화한 결과이다. 먼저 Fig. 8(a)의 태그 클라우드를 위해 사용한 키워드는 자동차 제조업체인 "*Toyota*, *Volkswagen*, *Mercedes-Benz*, *BMW*, *Hyundai*, *Ford*, *Kia*, *Volvo*, *Nissan*, *Ferrari*, *Lamborghini*, *Chevrolet*"이며, 수집된 단어 중 "*ParisMotorShow*"가 가장 많은 빈도수를 나타냈다. 다음으로 Fig. 8(b)의 키워드로 모바일 기기 제조업체인 "*Samsung*, *Apple*, *LG*, *Huawei*, *Lenovo*, *Xiaomi*, *Motorola*, *HTC*"를 사용하였으며, 단어 "*Apple*"이 가장 많은 빈도수를 나타냈다.

(a) The tag cloud of the car manufacturer. (b) The tag cloud of the mobile device manufacturer.

Fig. 8. Examples of the Diverse Tag Clouds

실험 결과를 통해, 제안 시스템은 트위터로부터 수집한 해시태그를 빈도에 따라 태그 클라우드로 잘 시각화함을 알 수 있다. 또한, 태그가 새롭게 수집될 때마다 태그 클라우드가 동적으로 변경되며, 여러 사용자가 입력한 키워드를 동시에 처리하여 서로 다른 시각화 결과를 각 사용자에게 제공한다. 따라서, 제안하는 실시간 동적 태그 클라우드 시스템은 특정 주제에 대한 SNS의 상황을 판단하기에 효과적이다. 제안 시스템은 데이터 수집에 트위터 API를, 시각화에 D3.js를 사용하였으며, 태그 클라우드의 정확도와 시각화 우수성은 이들 라이브러리에 의존한다. 따라서, 보다 높은 수준의 정확도와 시각화 달성을 위해서는 보다 우수한 API 및 라이브러리를 사용할 수 있다.

5. 결론 및 향후 연구

본 논문에서는 Storm을 사용하여 SNS 데이터를 실시간으로 계산하고 이를 동적인 태그 클라우드로 시각화하였다. 먼저, Storm을 통해 사용자가 입력한 키워드를 토대로 SNS 데이터를 수집하고, 배치 단위로 집계하기 위해 Storm의 트라이덴트 토폴로지를 설계하였다. 그리고 사용자가 키워드를 입력하고 태그 클라우드 결과를 확인할 수 있도록 웹 인터페이스를 설계하였다. 마지막으로, 이를 구현하여 실시간으로 계산된 동적 태그 클라우드 결과를 확인하였다. 본 시

스템을 활용하면, 빠르게 발생하는 SNS 데이터에서 발생하는 여러 이슈 및 변화를 직관적으로 판단할 수 있게 된다. 그러나, 구현 결과 데이터베이스 및 트위터와 연결하는 스파우트의 지연 시간이 높게 나타났다. 현재는 하나의 스파우트만 동작하기 때문에, 스파우트의 지연 시간은 곧 전체 시스템의 성능에 영향을 미친다. 따라서, 향후 데이터베이스와 트위터를 각각 연결하여 스파우트의 지연 시간을 줄이도록 Storm의 토폴로지를 재설계할 예정이다.

References

[1] Social Networking Service [Internet], https://en.wikipedia.org/wiki/Soci al_networking_service.

[2] Social Mining Part 1: How Big Data is transforming customer insights [Internet], http://www.incite-group.com/data-and-insights/social-mining-part-1-how-big-data-transforming-customer-insights.

[3] J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, and A. Byers, “Big Data: The Next Frontier for Innovation, Competition, and Productivity,” Technical Report, McKinsey Global Institute, 2011.

[4] M. A. Hearst and D. Rosner, “Tag Clouds: Data Analysis Tool or Social Signaller?,” In *Proc. of the 41st Int'l Conf. on System Sciences*, pp.1-10, Jan., 2008.

[5] Example of Tag Cloud [Internet], https://www.nngroup.com/articles/tag-cloud-examples/.

[6] Apache Storm [Internet], http://storm.apache.org/.

[7] A. Toshniwal, S. Taneja, A. Shukla, K. Ramasamy, J. M. Patel, S. Kulkarni, J. Jackson, K. Gade, M. Fu, J. Donham, N. Bhagat, S. Mittal, and D. Ryaboy, “Storm@Twitter,” In *Proc. of the Int'l Conf. on Management of Data*, ACM SIGMOD, pp.147-156, June, 2014.

[8] Apache Hadoop [Internet], http://hadoop.apache.org/.

[9] Apache Spark [Internet], http://spark.apache.org/.

[10] EsperTech Esper [Internet], http://www.espertech.com/esper/.

[11] Apache Storm Trident Tutorial [Internet], http://storm.apache.org/relea ses/current/Trident-tutorial.html.

[12] Y. Hassan-Montero and V. Herrero-Solana, “Improving Tag-Clouds as Visual Information Retrieval Interfaces,” In *Proc. of the Int'l Conf. on Multidisciplinary Information Sciences and Technologies*, Oct., 2006.

[13] M. A. Hearst and D. Rosner, “Tag Clouds: Data Analysis Tool or Social Signaller?,” In *Proc. of the Hawaii Int'l Conf. on System Sciences*, Jan., 2008.

[14] O. kaser and D. Lemire, “Tag-Cloud Drawing: Algorithms for Cloud Visualization,” In *Proc. of World Wide Web Workshop on Taggings and Metadata for Social Information Organization*, Mar., 2007.

[15] D3.js [Internet], https://d3js.org/.

[16] Twitter4j [Internet], http://twitter4j.org/en/index.html.

손 시 운

e-mail : ssw5176@kangwon.ac.kr
2014년 강원대학교 컴퓨터과학과(학사)
2016년 강원대학교 컴퓨터과학과(석사)
2016년~현 재 강원대학교 컴퓨터과학과 박사과정
관심분야 : 데이터마이닝, 빅데이터, 하둡 에코시스템

김 다 솔

e-mail : kimds0926@kangwon.ac.kr
2014년~현 재 강원대학교 컴퓨터과학과 학사과정
관심분야 : 빅데이터, 실시간 데이터 처리

이 수 정

e-mail : sujeonglee@kangwon.ac.kr
2013년~현 재 강원대학교 컴퓨터과학과 학사과정
관심분야 : 빅데이터, 실시간 데이터 처리

길 명 선

e-mail : gils@kangwon.ac.kr
2007년 강원대학교 컴퓨터과학과(학사)
2009년 강원대학교 컴퓨터과학과(석사)
2009년~2012년 강원대학교 중앙정보전산원
2012년~현 재 강원대학교 컴퓨터과학과 박사과정
관심분야 : 데이터마이닝, 시계열 분석, 빅데이터 분석, 하둡 에코시스템

문 양 세

e-mail : ysmoon@kangwon.ac.kr
1991년 한국과학기술원 전산학과(학사)
1993년 한국과학기술원 전산학과(석사)
2001년 한국과학기술원 전산학과(박사)
1993년~1997년 현대전자산업(주) 주임연구원
2001년~2002년 (주)현대시스콤 선임연구원
2002년~2005년 (주)인프라벨리 기술위원(이사)
2005년~2008년 한국과학기술원 첨단정보기술연구센터 연구원
2008년~2009년 미국 퍼듀대학교 방문연구원
2012년~2013년 강원대학교 기획부처장
2014년~2016년 강원대학교 IT대학 부학장
2005년~현 재 강원대학교 컴퓨터과학과 교수
관심분야 : 데이터마이닝, 스트림데이터, 저장 시스템, 데이터베이스 응용, 빅데이터 분석, 프라이버시 보호 마이닝