Storm 기반 실시간 SNS 데이터의 동적 태그 클라우드

손시운°, 김다솔, 이수정, 길명선, 문양세
강원대학교 컴퓨터과학과
e-mail: {ssw5176, kimds0926, sujeonglee, gils, ysmoon}@kangwon.ac.kr

Storm-based Dynamic Tag Cloud of Real-time SNS Data

Siwoon Son° , Dasol Kim, Sujeong Lee, Myeong-Seon Gil and Yang-Sae Moon
Dept. of Computer Science, Kangwon National University

요 약

최근 SNS(social networking service)의 사용이 급증함에 따라 SNS에서 발생하는 데이터의 분석이 활발해졌다. 하지만 SNS 데이터는 빠르게 생성되며 정형화 되어 있지 않은 빅데이터이기 때문에 그대로 수집할 경우 분석하기가 어렵다. 본 논문은 분산 스트리밍 처리 기술인 Storm을 사용하여 트위터에서 실시간으로 발생하는 데이터를 수집 및 집계하고, 태그 클라우드를 사용하여 집계 결과를 동적으로 시각화하고자 한다. 또한 사용자가 쉽게 키워드를 입력하고 시각화 결과를 실시간으로 확인할 수 있도록 웹 인터페이스를 구현한다. 그리고 결과를 통해 태그 클라우드의 결과가 시간에 따라 바르게 시각화되었는지 확인한다. 본 논문은 빠르게 발생하는 SNS 데이터로부터 각 키워드와 관련된 정보를 시각화하여 각 사용자에게 제공할 수 있는 우수한 결과라 사료된다.

1. 서론

최근 모바일 기기의 대중화로 인해 SNS(social networking service)의 사용이 급증하였다. SNS는 사용자 간의 의사 표현, 정보 공유, 친목 도모 등을 목적으로 하며, 대표적으로는 페이스북(Facebook), 트위터(Twitter), 인스타그램(Instagram) 등이 있다. SNS에는 대용량의 다양한 데이터가 빠르게 생성되기 때문에 SNS 데이터를 빅데이터라 할 수 있다. 현재 산학연에서는 이러한 SNS 데이터를 분석하는 연구가 활발히 진행 중이다[1]. 특히, SNS 데이터를 분석하여 시각화한다면 트렌드를 쉽게 파악할 수 있다.

본 논문은 SNS 데이터의 시각화에 중점을 둔다. 시각화 기술 중, 태그 클라우드[2-4]는 단어들을 나열하여 구름 형태로 표현하되 단어의 빈도수에 따라 크기 별로 나타내어, 비중 있는 단어를 더 직관적으로 분석할 수 있다. 하지만 기존의 태그 클라우드 기술은 단순히 문서 집합으로부터 단어의 수를 계산하여 정적으로 시각화한다. 이러한 방법으로는 실시간으로 생성되는 SNS 데이터를 시각화하기에는 제약이 있다.

Apache Storm[5]은 다수의 서버로 구성된 클러스터에서 실시간으로 수집되는 데이터를 분산 처리하는 스트리밍 소프트웨어 플랫폼이다. 따라서 Storm을 사용하여 실시간으로 빠르게 생성되는 SNS 데이터를 분석한다면, 기존의 단일 서버 시스템에 비해 더 많은 양의 데이터를 처리할 수 있다. 본 논문은 이러한 Storm을 통해 대표적인 SNS인 트위터로부터 해시태그를 집계하여, 결과가 실시간으로 반영되는 동적 태그 클라우드를 시각화하고자 한다.

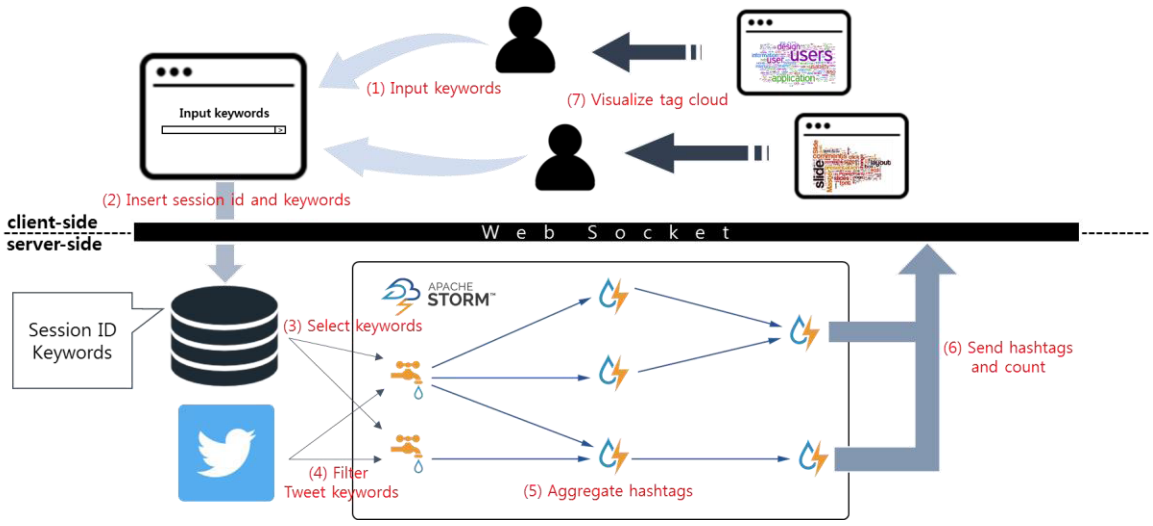
* 이 논문은 2016년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No.R7117-16-0214, 데이터 스트림 정제를 위한 지능형 샘플링 및 필터링 기술 개발).

2. 관련 연구

실시간 데이터 처리 기술은 실시간으로 발생하는 스트림 데이터를 수집과 동시에 처리하는 기술로, 대표적으로 Apache Storm, Apache Spark 등이 있다. 이 중, Storm은 다수의 서버를 클러스터로 구성하여 실시간으로 데이터를 처리함으로써, 필요에 따라 클러스터를 확장(scale-out)할 수 있다. Storm은 작업을 처리할 때 데이터의 입력부터 출력까지 일련의 작업을 토폴로지(Topology)로 표현하며, 토폴로지는 크게 스파우트(Spout)와 볼트(Bolt)로 구성된다. 먼저, 스파우트는 스트리밍 데이터의 발생지로부터 데이터를 입력받아 토폴로지 내에서 다루는 데이터 형태인 튜플(Tuple)로 바꾸어 볼트에게 전달한다. 다음으로, 볼트는 전달 받은 데이터를 처리하고, 다음 볼트로 전달하거나 결과를 출력 또는 저장 장치로 보낸다.

일반적인 Storm은 토폴로지 내에서 데이터를 튜플 단위로 처리한다. 하지만 이 방법으로는 튜플 간의 관계 및 연산을 처리하기가 어렵다. 따라서 Storm은 일정 기간 또는 특정 단위의 수로 튜플을 모아 배치로 처리하는 트라이덴트(Trident)[6]를 제공한다. 트라이덴트는 Storm 상에서 실시간으로 집계 연산을 수행하기 위한 고수준 추상화 기술이다. 이러한 트라이덴트를 사용하면 Storm에서도 Join, Filter, Grouping, Aggregation, Fuction 등의 연산을 수행할 수 있다. 본 논문에서는 배치 단위로 단어를 집계하기 위해 일반적인 Storm이 아닌 트라이덴트를 사용한다.

다양한 시각화 기술 중에서 태그 클라우드는 문서 집합에서 나타나는 단어들을 빈도수에 따라 시각적으로 표현하는데 매우 유용하다. 초기의 태그 클라우드 연구는 각 단어들을 발생순, 단어순, 빈도순 등의 순서로 나열하고 빈도 수에 따라 단어의 크기를 나타내



(그림 1) 동적 태그 클라우드 시스템의 전체 구조도.

었다[2, 3]. 하지만 이러한 방법은 빈도에 상관없이 모든 단어가 같은 크기의 공간을 차지하며, 단어가 많아짐에 따라 빈 공간이 넓어져 높은 빈도를 갖는 단어의 직관성이 낮아진다. 이를 개선한 연구로 전체 공간을 단어의 빈도 수에 따라 나누어 태그 클라우드를 표현하였다[4]. 그러나 이 방법 또한 모든 단어의 방향이 수평형으로 같기 때문에 단어 사이의 공백이 클 수 밖에 없다. 최근, 시각화를 다루는 D3.js[7]에서 제공하는 태그 클라우드와 같이 단어 사이의 공백에 적합하게 단어의 방향을 바꾸어 태그 클라우드를 시각화한다. 이 방법은 단어가 서로 겹치지 않으면서도 단어 사이의 공백이 최소화될 수 있다.

3. 동적 태그 클라우드 시스템의 설계

3.1. 동적 태그 클라우드 시스템의 전체 구조

(그림 1)은 본 논문에서 제안하는 동적 태그 클라우드 시스템의 구조도이다. 시스템은 가운데의 웹 소켓(Web socket)을 기준으로 위는 클라이언트, 아래는 서버의 형태로 구성되어있다. 다음은 시스템의 사용자를 기점으로 구조도를 자세히 설명한 것이다.

(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 키워드 입력과 토폴로지는 비동기적으로 동작한다.

(4) **Filter Tweet keywords**: 스파우트는 키워드들을 필터링한다. 즉, 트위터로부터 키워드가 포함된 트윗만을 실시간으로 수집한다.

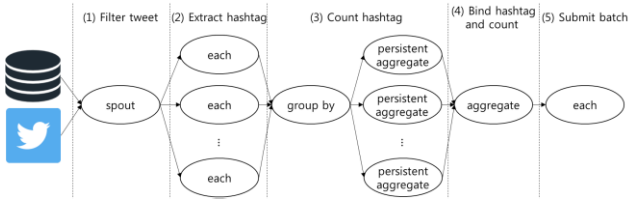
(5) **Aggregate hashtags**: 해시태그의 수를 집계한다.

(6) **Send hashtags and count**: 집계한 해시태그를 웹 서버로 전달한다.

(7) **Visualize tag cloud**: 세션 ID를 사용하여 사용자를 구분하고, 각 사용자에게 태그 클라우드를 시각화하여 제공한다.

3.2. 해시태그 집계를 위한 Storm 토폴로지 설계

다음은 트윗을 수집하고 해시태그를 집계하는 Storm 토폴로지에 대한 자세한 설계이다. 본 논문에서는 일반적인 Storm이 아닌 트라이덴트를 사용하였다. 이는 제안하는 시스템에서 수집한 해시태그들을 일정 시간마다 배치 형태로 집계하기 위함이다. (그림 2)는 이러한 트라이덴트 토폴로지의 설계이며, 다음은 이를 설명한 것이다.



(그림 2) 집계를 위한 Storm 트라이덴트 토폴로지.

(1) **Filter tweet**: 스파우트는 데이터베이스에서 키워드를 검색하고, 트위터로부터 키워드들이 포함된 트윗들을 필터링한다. 이때, 일정 기간의 배치 단위로 데이터를 집계할 수 있도록 기간 내에 발생한 데이터는 같은 배치 ID를 부여한다. 또한 필터링된 트윗이 어떤 키워드로 필터링이 되었는지 알아야 한다. 이는 다수의 사용자가 각자 여러 키워드를 입력할 것이며, 집계된 결과를 각 사용자에게 바르게 보내기 위함이다. 스파우트는 끊임없이 반복하여 실시간으로 배치를 생성한다.

(2) **Extract hashtag**: 스파우트로부터 필터링된 트윗에서 해시태그를 추출한다. 하나의 트윗에는 하나 이상의 해시태그가 포함될 수 있으며, 따라서 배치 ID, 해시태그, 트윗에 포함된 키워드의 리스트를 다음 볼트로 전달한다.

(a) 배치가 처리되지 않은 초기 상태. (b) 첫 번째 배치가 처리된 상태. (c) 두 번째 배치가 처리된 상태. (d) 5분 간 시각화한 결과.

(그림 3) 태그 클라우드 시각화를 위한 두 번째 페이지.

(3) **Count hashtag**: 배치 ID 및 해시태그를 키(key)로 그룹핑(grouping)하고, 각 키에 대해 집계한다. 이 결과로 같은 배치 ID 내에서 해시태그와 그 수가 계산된다.

(4) **Bind hashtag and count**: 배치 ID 내의 모든 해시태그들을 집계한다.

(5) **Submit batch**: 웹 소켓을 통해 앞에서 집계한 배치를 웹 서버로 전달한다. 이때 전달하는 데이터의 형태는 JSON을 사용하였다.

4. 동적 태그 클라우드 시스템의 평가

본 절에서는 앞서 설계한 동적 태그 클라우드 시스템을 구현하고, 그 결과를 확인한다. 본 논문은 한 대의 님버스 서버와 여덟 대의 슈퍼바이저 서버로 구성된 Storm 클러스터와 한 대의 웹 및 데이터베이스 서버에서 구현 및 실행하였다. 소프트웨어 플랫폼으로 운영체제 및 그 버전은 CentOS 7 64bit, Storm은 Apache Storm-1.0.2, 웹 서버는 Apache Tomcat 7, 데이터베이스는 MariaDB 5.5를 사용하였다.

사용자 인터페이스는 두 개의 웹 페이지로 단순하게 구성하였다. 첫 번째 페이지는 사용자로부터 키워드를 입력 받으며, 두 번째 페이지는 Storm 토폴로지로부터 주기적으로 단어와 그 수를 받아 태그 클라우드로 시각화한다. (그림 3)은 시각화를 위한 두 번째 페이지로써, 예시를 위해 사용한 키워드는 *BigData*, *빅데이터*, *Hadoop*, *하둡*, *Real-time*, *Visualization*, *시각화*, *Tag Cloud*, *태그 클라우드*로 총 9개를 사용하였다. 본 논문에서는 태그 클라우드 시각화를 위해 D3.js 라이브러리를 사용하였으며, 5초 단위로 배치를 처리하도록 스파우트의 배치 시간을 설정하였다. 먼저 (그림 3)의 (a)는 두 번째 페이지의 초기 상태이며 아직 어떠한 배치도 처리되지 않아 태그 클라우드가 시각화되지 않았다. 다음으로 그림 (b)는 그림 (a)의 상태에서 5초가 지나 첫 번째 배치가 처리되어 태그 클라우드가 시각화된 것이다. 첫 번째 배치에는 총 8 개의 단어가 입력되었으며, 단어 *BigData*는 다른 단어에 비해 가중치가 높아 더 큰 글자로 표현되었다. 그리고 그림 (c)는 그림 (b)의 상태에서 5초가 지나 두 번째 배치가 처리되어 태그 클라우드가 변경된 것이다. 두 번째 배치에서는 7 개의 단어가 추가로 입력되었으며, 첫 번째 배치에서 입력된 단어 *IoT*와 *Cloud*는 두 번째 배치에서도 추가로 입력되어 더 큰 글자로 표현되었다. 마지막으로 그림 (d)는 5분의 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.

5. 결론 및 향후 연구

본 논문에서는 Storm을 사용하여 SNS 데이터를 실시간으로 수집하고, 그 결과를 동적인 태그 클라우드로 시각화하였다. 먼저, Storm을 통해 사용자가 입력한 키워드를 토대로 SNS 데이터를 수집하고, 배치 단위로 집계하기 위해 Storm의 트라이덴트 토폴로지를 설계하였다. 다음으로, 사용자가 키워드를 입력하고 태그 클라우드 결과를 확인할 수 있도록 웹 인터페이스를 구현하였다. 마지막으로, 실시간으로 수집된 데이터가 태그 클라우드에 동적으로 반영되는 것을 확인하였다. 따라서 본 논문을 통해 사용자는 빠르게 발생하는 SNS 데이터로부터 키워드와 연관성 높은 데이터를 직관적으로 확인할 수 있다.

하지만 구현 결과, 데이터베이스 및 트위터와 연결하는 스파우트의 지연 시간이 높게 나타났다. 따라서 향후 데이터베이스와 트위터를 각각 연결하여 스파우트의 부하를 줄이도록 Storm 토폴로지를 재설계할 예정이다.

참고문헌

[1] J. S. Kim, M. H. Yang, Y. J. Hwang, “Customer Preference Analysis Based on SNS Data,” In *Proc. of the Second Int'l Conf. on Cloud and Green Computing*, pp. 609-613, Nov. 2012.

[2] 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*, pp. 25-28, Oct. 2006.

[3] 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.

[4] 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.

[5] 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 ACM SIGMOD Int'l Conf. on Management of Data*, pp. 147-156, June 2014.

[6] Apache Storm Trident Tutorial, http://storm.apache.org/releases/current/Trident-tutorial.html.

[7] D3.js, https://d3js.org/.