한국컴퓨터정보학회지
제21권 제1호, 2013. 6.

빅데이터와 빅데이터 정제 기술

• 박호진 ((주)와이즈넛 연구소)
• 권영현 ((주)와이즈넛 연구소)
• 안영민 ((주)와이즈넛 연구소)

I. 서론

최근 ICT 분야에서 빅데이터(Big Data)가 화두로 떠오르고 있으며, ICT 분야의 세미나 · 컨퍼런스에서도 빅데이터라는 주제가 점차 중심에 자리를 잡고 있다.

특히 스마트 단말기의 보급 및 다양한 소셜미디어의 출현으로 데이터의 종류가 다양해 졌으며, 폭발적인 데이터의 생성을 이끌고 있다. 또한, 기업들의 고객 데이터 수집 및 사물정보(센서, 모니터링)와 같은 스트리밍 정보 등 실시간성 데이터도 증가하고 있는 추세이다.

그렇다면 빅데이터를 어떻게 정의할 수 있을까? McKinsey[1]은 "기존 데이터베이스 관리도구로 데이터를 수집, 저장, 관리, 분석할 수 있는 역량을 넘어서는 대량의 정형 또는 비정형 데이터 집합"으로 정의하고 있으며, IDC[2]는 2011년 "다양한 종류의 대규모 데이터로부터 저렴한 비용으로 가치를 추출하고 데이터의 초고속 수집, 발굴, 분석을 지원하도록 고안된 차세대 기술 및 아키텍처"로 정의하고 있다. 즉, 빅데이터란 아주 큰 데이터에 대한 수집, 저장에만 초점이 맞추어진 것이 아니라 데이터의 분석, 가치 있는 데이터 창출, 시각화까지 포함하고 있다는 것을 알 수 있다.

2011년 TDWI Research[3]에서는 빅데이터의 특성을 3V(Volume, Variety, Velocity)로 정의하고 있다.

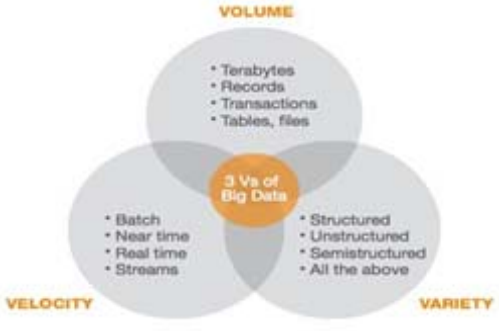

그림 1. TDWI Research의 3V

Volume은 데이터의 물리적인 크기를 의미한다. 통계에서 일정 수준 이상의 표본 데이터가 있어야 의미있는 데이터를 얻을 수 있듯이 빅데이터에서도 다량의 분석 데이터는 분석 결과에 대한 신뢰도를 높일 수 있다. 빅데이터를 규정하는 데이터의 사이즈에 대해 많은 이견이 있지만, 일반적으로는 수 페타바이트에서 수백 페타바이트 이상의 크기를 빅데이터라고 정의하고 있다.

Variety는 데이터의 다양성을 의미한다. 빅데이터는 기업 내의 DBMS 저장·관리되는 정형 데이터, XML/HTML, 웹 로그과 같은 반정형 데이터 뿐만 아니라 텍스트 문서 및 멀티미디어와 같은 비정형 데이터의 형태로 이루어져 있으며, 이

러한 데이터에 대한 분석을 할 수 있어야 한다. 특히, 최근에는 다양한 소셜미디어의 등장으로 비정형 데이터가 폭발적으로 늘어나고 있는 추세이다.

Velocity는 데이터의 생성 속도를 의미한다. 소셜미디어와 같이 다양한 사용자들이 생성하는 데이터에 대한 수집, 정제, 분석하는 과정을 실시간 혹은 적절한 시간내에 처리가 가능해야 한다.

IT 분야의 전문컨설팅기업인 Gartner[4]에서는 빅데이터의 특성을 3V(Volume, Variety, Velocity)에 1C(Complexity)를 추가로 정의하고 있다.

그림 2. Gartner Group의 3V, 1C

Complexity는 데이터의 복잡성을 의미하며, Variety 특성과 연관성이 있다. 데이터의 다양성으로 인하여 하나의 접근 방식으로 처리하는 것이 아니라 데이터의 종류 혹은 형태에 따라 적합한 기술들을 적용할 수 있어야 한다.

이처럼 최근 다양한 기업에서 빅데이터의 제4의 특성을 정의하고 있으며, IBM은 정확성(Veracity), Forrester는 가변성(Variability), Oracle은 가치(Value)를 선정하였다. 즉, 빅데이터에서는 데이터의 수집, 저장, 분석 뿐만 아니라 분석된 데이터가 유효한 의미로서 새로운 가치를 창출할 수 있는가에 대한 관점도 고려가 되어야 한다는 것을 알 수 있다. 최근 데이터 과학자(Data Scientist)의 필요성이 증가되는 현상과 유사하다고 볼 수 있을 것이다.

위에서 살펴본 빅데이터의 특성을 고려한 빅데이터 기술은 분석 인프라와 분석 기술로 나눌 수 있다. 빅데이터 분석 기술에는 텍스트 마이닝(Text Mining), 평판 분석(Opinion Mining), 소셜 네트워크 분석(Social Network Analytics), 군집 분석(Cluster Analysis) 등이 최근 주목을 받고 있다. 이러한 기술들은 이미 기계학습 및 데이터 마이닝 분야에서 사용되었던 기술이며, 분산 환경에서 적합한 구조로 변경하여 적용하고 있다.

빅데이터에서 가치 있는 데이터를 분석하는 기술도 매우 중요하지만, 데이터를 정제하는 작업도 중요하다고 할 수 있다. 일부 빅데이터 전문가들은 데이터를 정제하는데 80%의 노력을 투자하라고 말하고 있다. 예를 들어, 트위터(Twitter) 데이터를 이용하여 평판 분석을 수행한다고 가정하자. 트위터에는 다른 사용자가 작성한 트윗(Tweet)을 다른 트위터 사용자에게 공유하기 위한 리트윗(Retweet) 기능을 제공하고 있다. 만약, 리트윗이 많이 된 트윗이 긍정이나 부정을 포함하는 트윗이라면 평판 분석의 결과는 편향될 수 밖에 없다.

본 논문에서는 빅데이터 현황 및 빅데이터 정제 기술의 하나로 빅데이터에 산재되어 있는 중복된 데이터를 검출할 수 있는 중복 데이터 검출(Near Duplicate Detection) 기술에 대해서 상세히 설명하고자 한다.

II. 빅데이터 현황

1. 빅데이터 인프라 기술

빅데이터의 3가지 특성(3V)을 수용하기 위해서는 분산 처리 환경이 요구된다. 빅데이터의 크기 특성만 보더라도 페타바이트급 데이터를 단일 기기에서 저장한다는 것은 불가능하다. 본 장에서는 빅데이터 인프라 기술인 Hadoop, R, NoSQL, Mahout에 대해서 알아본다.

1.1. Hadoop

하둡(Hadoop)[5]은 더그 커팅과 마이크 카파렐라에 의해 개발된 프로젝트로, 정형/비정형 빅데이터를 분산 처리할 수 있는 오픈소스 프레임워크이다. 하둡은 구글의 GFS(Google File System)을 대체할 수 있는 HDFS(Hadoop Distributed File System)와 MapReduce를 포함하고 있다. 하둡은 HDFS 상에 정형/비정형 데이터를 분산 저장하고, MapReduce를 이용하여 분산 처리한다. 정형 데이터에 대해서는 기존 RDBMS에서도 처리가 가능하지만, 웹로그와 같은 비정형 데이터를 RDBMS에 저장하기는 데이터 크기가 매우 크며, 지속적인 스토리지 확장과 라이선스 구입 등 비정형 빅데이터 처리를 위한 분산 환경 구성 시 많은 비용이 필요하다. 이러한 이유에

서 저렴한 비용으로 빅데이터를 처리할 수 있는 하둡이 최근 주목을 받고 있다.

1.2. R

Ross Ihaka와 Robert Gentleman에 의해 개발된 R[6]은 통계 계산과 시각화(Visualization)를 위한 프로그래밍 언어이자 소프트웨어 환경이다. R은 기본적인 통계 라이브러리를 포함하여 약 3,400개 이상의 패키지가 공개되어 있으며, 구글, 페이스북, 아마존 등 빅데이터 분석이 필요한 기업에서 널리 사용되고 있다. R은 기본적으로 Single core / In-memory 기반으로 동작하지만, 추가 패키지를 통하여 이러한 제약을 해결할 수 있다. 또한, R과 하둡을 연동한 패키지도 지원되고 있다.

1.3. NoSQL

NoSQL은 No SQL 혹은 Not-Only SQL 을 의미하며, RDBMS와는 달리 데이터 간의 관계를 정의하지 않는 것이 특징이다. NoSQL은 기존 RDBMS 비해 대용량의 데이터를 저장할 수 있으며, 분산형 구조를 가지고 있으며, 스키마가 고정되지 않은 Key-Value 저장 방식을 취하고 있다. 2000년 Eric Brewer에 의해 제시된 CAP 이론[7]에서 분산형 시스템은 일관성(Consistency), 유효성(Availability), 분산 수용성(Partition Tolerance) 세 가지 특징을 가지고 있으며, 이중 두 가지 특징만 만족할 수 있다는 이론이다.

그림 3. CAP 이론

NoSQL도 분산형 구조를 가지고 있기 때문에 CAP 이론을 따르며, 일관성(Consistency) 혹은 유효성(Availability) 보다는 분산 수용성(Partition Tolerance)에 중점을 두고 있다. 따라서, 데이터의 일관성이 중요한 시스템에서는 적합하지 않으며, 이러한 시스템들은 상용 RDBMS를 사용해야 한다.

1.4. Mahout

아파치 마훗(Mahout)는 하둡과 연동되는 빅데이터를 위한 기계학습 오픈소스 프로젝트이다. 마훗은 기본적으로 분류(Classification), 군집(Clustering), 추천 및 협업 필터링(Recommenders / Collaborative Filtering)을 지원했으며, 최근 패턴 마이닝(Pattern Mining), 회귀 분석(Regression), 차원 감소(Dimension Reduction), 진화 알고리즘(Evolutionary Algorithms) 등을 확장하고 있다. 이러한 기계학습은 MapReduce 환경에서 수행하는 것을 목적으로 하고 있으며, 단일 기기에서도 동작할 수 있다. 또한, RDBMS 나 NoSQL을 데이터 소스로 쉽게 연동할 수 있다.

2. 빅데이터 분석 기술

빅데이터 분석 기술은 매우 다양하지만, 최근 주목을 받고 있는 텍스트 마이닝(Text Mining), 평판 분석(Opinion Mining), 소셜 네트워크 분석(Social Network Analytics), 군집 분석(Clustering Analysis)에 대해서 알아본다.

2.1. 텍스트 마이닝(Text Mining)

자연 언어 처리에 기반한 텍스트 마이닝은 비정형 텍스트 데이터에서 의미 있는 정보를 추출하고, 그 정보와 다른 정보와의 연계성 분석 및 주제 분석(Topic Analysis) 등의 비정형 텍스트 데이터에 대한 의미를 분석할 수 있는 기술이다. 데이터 마이닝(Data Mining)은 정형 데이터에서 데이터의 경향, 패턴 등의 유용한 정보를 발견하는 기술이라면, 텍스트 마이닝은 비정형 텍스트 데이터에서 유용한 의미를 분석하는 기술이다. 텍스트 마이닝의 주요 기술 분야는 정보 추출(Information Extraction), 문서 요약(Document Summarization), 문서 분류(Document Classification), 문서 군집(Document Clustering), 주제 분석(Topic Analysis) 등이다. ㈜와이즈넛에서는 비정형 텍스트 데이터에 대한 주제어 추출(Main Keyword Extraction) 및 LDA(Latent Dirichlet Allocation)[8]에 기반한 주제 추출

(Topic Analysis) 등의 기술을 다루고 있다.

2.2. 평판 분석(Opinion Mining)

SNS, 리뷰, 블로그, 커뮤니티 등에서 정형화되지 않는 제품 및 서비스에 대한 긍정(Positive), 부정(Negative), 중립(Neutral) 등의 평판(Reputation)을 분석, 판별하는 기술이다. 평판 분석은 제품 및 서비스에 대한 사용자의 선호도, 인지도, 품질 관리 등에 활용되고 있다. 평판 분석을 위해서는 평판을 나타내는 문장 표현, 단어 등의 지식 자원의 구축이 필요하며, 이러한 지식 자원은 분석 대상의 도메인에 따라 그 의미가 달라질 수 있다는 것을 인지하고 있어야 한다. ㈜와이즈넛에서는 LSP(Lexico Semantic Pattern) 형태의 문법을 기술하여 평판 문장을 분석할 수 있는 기술을 보유하고 있다.

2.3. 소셜 네트워크 분석(Social Network Analytics)

소셜 네트워크 분석은 SNS 상의 비정형 데이터에서 사용자 간의 연결 구조 및 영향력, 트렌드 등을 분석하고 추출하는 기술이다. 개인의 영향력 및 그룹(연령, 성별, 지역 등)의 관심사를 분석하여 마케팅에 사용되고 있다. ㈜와이즈넛에서는 2012년 대선에서 후보자별 호불호 종합결과, 인품, 자질, 지지도, 긍정, 부정 등의 감성 분석(Sentiment Analysis) 기술을 활용한 2012 대선 후보 SNS 분석서비스[9]를 출시했다.

2.4. 군집 분석(Clustering Analysis)

군집 분석은 SNS 상의 비정형 데이터를 분석하여 유사 특성을 가지는 사용자 군(Group)을 발굴하는 기술이다. 즉, 소셜 네트워크 상에서 사용자들이 작성한 내용을 분석하여 비슷한 관심사를 가지는 사용자 군을 생성하는 기술이다.

III. 빅데이터 정제 관련 연구

빅데이터에서 분석 기술도 중요하지만 분석 전 데이터에 대한 정제도 매우 중요하다. 본 장에서는 빅데이터 정제 기술의 하나인 중복 데이터 검출 및 최적화 기술의 선행 연구에 대해서 알아본다.

1. 중복 데이터 검출 기술

SNS, 뉴스, 블로그 등의 데이터는 리트윗 기능이나 퍼오기 기능을 제공하기 때문에 중복 데이터가 많이 존재하게 된다. 이러한 중복 데이터는 불필요한 저장 공간을 필요로 하게 되며, 분석에서도 많은 시간이 소요되고, 분석 결과에 대한 신뢰도를 떨어트릴 수 있다. 이전 연구에서는 텍스트 데이터를 Feature Vector로 표현하고, Vector 간의 유사도를 측정하여 중복 데이터를 검출하는 Shingles[10], Document vector[11] 등이 연구되었으며, 최근에는 텍스트 데이터를 하나의 Fingerprint로 표현하여 중복 데이터를 검출하는 Mod-p Shingles[12], Min-Hash[13], Simhash[14] 등이 연구되었다.

2. 중복 데이터 검출 최적화 기술

중복 데이터 검출은 각각의 데이터를 Feature Vector 혹은 Fingerprint로 변환하고, 이를 비교하는 방식으로 수행된다. 빅데이터의 방대한 데이터를 모두 비교한다면, 중복 데이터 검출에 필요한 시간이 많이 소요되기 때문에 검출 비교 대상을 한정하여 비교 횟수를 최적화할 수 있는 기술도 활발이 연구되고 있다. 비교 횟수를 줄이기 위해서는 데이터의 특정 자질을 추출하여 데이터 군집(Group)을 생성하고, 생성된 군집 내에서만 중복 데이터 검출 기술을 적용함으로써 검출 시간을 획기적으로 줄일 수 있다. 최적화 기술로는 I-Match[15], Multi-level Indexing[16] 등이 연구되었다.

IV. N-Gram TF 기반 문서 군집화

본 논문에서는 기존 I-Match, Multi-level Indexing의 문제점을 해결할 수 있는 중복 데이터 검출의 최적화를 위한 N-Gram TF 기반 문서 군집화 방법에 대해서 소개한다.

I-Match 기법은 단어의 IDF(Inverse Document Frequency)를 이용하여 문서의 대표 단어를 추출하고, 추출된 단어들을 SHA-1 해시로 생성하여 SHA-1 해시가 동일한 문서들을 중복 데이터로 판단하였다. 문서의 대표 단어는 모든 단어를 IDF 값으로 정렬한 후 중간(Mid), 상위(Upper), 하위(Lower), 양끝(Dual Extremes)의 비율로 추출하여 사용하였다. 단어의 IDF 값은 I-Match 수행 전 전체 데이터를 대상으로 먼저 계산하는 방식과 말뭉치로 학습된 IDF 값을 사용하는 방식을 제안하고

있으며, 전자의 방식은 데이터의 크기가 커지면 커질수록 IDF 계산 비용이 크게 증가하기 때문에 후자의 방식을 권고하고 있다. I-Match는 단어의 IDF를 말뭉치로부터 학습해야 하기 때문에 학습된 도메인과 다른 도메인에 적용할 경우에는 부정확한 IDF 값이 될 수 있으며, 데이터 희소성 문제도 발생된다. 또한, 문서에서 일부 단어만을 중복 데이터 검출에 사용하기 때문에 검출 정확도도 낮을 것으로 예상된다.

Multi-level Indexing 기법은 대용량 프로그램에서 함수의 중복 여부를 검출하기 위해 고안된 방법으로 함수의 라인 수(Line based index, L-Index)와 함수에 대한 Simhash 값에서 1인 비트의 개수(Bit based index, B-Index)를 정의하고 있으며, 이 두 가지를 혼용한 방법(Multi-level index, M-Index)을 제안하였다. 또한, 각 인덱스 방법에 윈도우 크기를 조절하여 중복 데이터 검출 범위를 설정할 수 있다. Multi-level Indexing는 윈도우 크기에 따라 성능에 영향을 받는다. 예를 들어, 각 인덱스 방법의 윈도우 크기를 크게 설정할 경우 비교 대상의 많아지며 성능이 저하된다. 반대로, 크기를 작게 설정할 경우에는 대상 범위가 작아지기 때문에 검출 정확도가 떨어진다. 또한, 유사한 크기의 데이터에 적용할 경우에는 L-Index도 유사한 값을 가지게 되므로 비교 대상이 많아질 수 밖에 없다.

㈜와이즈넛에는 위 두 가지 알고리즘의 문제점을 해결하기 위하여 Unigram TF 기반 군집화(RUG)[17]을 제시하였으며, 이를 N-Gram TF 기반 문서 군집화로 확장하였다. 제안 방법은 중복 데이터를 검출하기 이전에 데이터를 군집화하는 방법으로 사용하였으며, 군집된 문서들 내의 중복 데이터 검출은 Simhash를 사용하였다. 빅데이터의 중복 데이터 검출의 성능 및 품질을 보장하기 위해서는 아래와 같은 특징을 만족시켜야 한다.

- 데이터가 여러 군집으로 고르게 분포되어야 한다. 즉, 각각의 군집에 포함되는 데이터 양이 균일해야 한다. (성능 측면)
- 실제 중복인 데이터는 동일 군집으로 할당이 되어야 한다. (품질 측면)

위의 특성을 만족하기 위하여 본 논문에서는 N-Gram TF를 고안하였으며, 데이터를 N-Gram으로 분리한 후 고빈도 M개의 N-Gram들을 문서의 군집 자질로 사용하였다. IDF를 사용하는 것보다 고빈도 TF를 사용하면 아래와 같은 장점들이 있다.

(1) IDF는 신규로 추가되는 데이터에 따라 민감하게 반응한다. 즉, 전체 IDF를 다시 계산해야할 필요가 있다.
(2) 말뭉치로 IDF를 학습할 경우에는 도메인이나 언어에 종속적이게 된다.
(3) 데이터에 내용이 추가/삭제/변경되는 경우에도 고빈도 TF가 덜 민감하게 반응한다.

본 논문에서 제안한 알고리즘은 아래와 같은 순서로 동작하며, [그림 4]는 처리 과정을 도식화 한 것이다.

① 데이터를 White Space와 특수문자 기준으로 나누어 단어 리스트로 생성한다. (T)
② (T)에 대한 Simhash를 생성한다. (S)
③ (T)에 대한 N-Gram을 생성하고, 고빈도 M개의 N-Gram 리스트를 추출한다. (M)
④ (M)을 SHA-1 해시로 생성한다. (G)
⑤ (G)가 동일한 데이터에 대해서 군집을 생성한다.
⑥ 동일 군집 내의 전체 데이터 (S)에 대해서 Hamming Distance 이용하여 중복 여부를 판별한다.

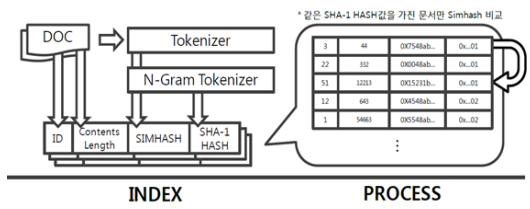

그림 4. N-Gram TF 군집 흐름도

위에서 설명한 알고리즘에서 N 과 M 값은 성능과 품질에 관련이 있으며, 언어 구성에 따라 다르게 설정을 해야 한다. 예를 들어, 한글으로만 구성된 데이터에서 N 과 M 값이 1인 경우 최대로 나뉠 수 있는 군집은 11,172 개(자음/모음의 조합으로 표현할 수 있는 최대 글자 수)이며, 영문으로만 구성된 데이터에서는 26개의 군집이 생성된다. 앞에서도 설명하였듯이 군집의 개수가 작을 경우 하나의 군집에 포함되는 데이터의 양이 많아지게 되므로 성능이 저하될 수 있다. 반대로

N 과 M 값이 큰 경우에는 너무 많은 군집이 생성되면서, 실제 중복인 데이터들이 각기 다른 군집에 포함될 수 있기 때문에 품질이 저하될 수 있다. 일반 한글 데이터에 대해서는 N이 1이고 M이 2인 경우에 최적의 군집 결과를 보였다.

위의 과정 ⑥에서 군집내의 중복 데이터 검출 최적화를 위하여 M-Index를 적용할 수 있다. 이는 RUG로 군집된 문서를 M-Index 기법으로 한번 더 군집함으로써 빅데이터에 대한 중복 데이터 검출의 성능을 더욱 향상시킬 수 있다.

V. 실 험

본 실험에서는 군집 정확도와 중복 데이터 검출 성능에 대해서 평가를 실시하였다.

군집 정확도는 M-Index와 비교를 수행하였으며, 실험 데이터는 군집 정확도 평가를 위하여 100개의 문서에 대해서 11가지의 단어 변화량(단어 추가/삭제/변경)를 추가하여 생성하였다.

표 1. RUG 와 M-Index의 군집 정확도

알고리즘	Baseline	M-Index	RUG
군집개수	100	100	103

[표 1]의 군집 정확도는 RUG 보다 M-Index 가 더 좋은 품질을 보였다. RUG에서는 3개의 군집이 각각 2개의 군집으로 나누어지는 현상을 보였으며, 이러한 현상은 짧은 데이터에서 발견되었으며, 중복 데이터 검출 품질에도 영향을 줄 수 있다.

성능 비교는 ㈜와이즈넛의 Search Formula-1 V5 Indexer에 중복 데이터 검출 기능을 추가하여 측정하였다. 서버 환경은 Intel(R) Xeon(R) CPU E5620 @ 2.40GHz, 8G RAM 이며 OS는 CentOS 6.2를 사용하였다. 실험 데이터는 10만, 30만, 50만, 100만, 200만, 300만 건의 뉴스 데이터를 이용하여 측정하였다.

표 2. 데이터 건수 별 중복 데이터 검출 성능

	색인시간 (분)			군집개수	
	Indexing	M-Index	RUG	M-Index	RUG
10만	1.31	1.95	**1.31**	104,691	104,732
30만	4.22	13.90	**4.47**	317,846	317,937
50만	7.02	51.43	**7.25**	524,144	524,371
100만	15.17	140.89	**17.41**	974,919	975,276
200만	35.15	387.54	**37.48**	1,875,463	1,876,180
300만	53.65	694.67	**56.23**	2,862,111	2,863,050

[표 2] 색인시간의 Indexing은 중복 데이터 검출을 제외한 색인DB 생성 시간이다. M-Index와 RUG는 색인시간을 포함하여 중복 데이터 검출 시간을 측정하였다. 즉, 300만건에 대해서 색인DB 생성 시간을 제외한 중복 데이터 검출 시간은 M-Index이 약 641분 소요되었으며, RUG는 약 3분 소요되어 약 213배의 성능 개선 효과를 보였다. 군집개수에서는 300만건 데이터에 대해서 약 0.03% 의 차이를 보였으며 이는 일반적인 상황에서 수용할만한 오차로 보여진다.

VI. 결 론

최근 ICT 분야에서 빅데이터가 화두에 떠오르고 있으며, 빅데이터라고 하는 것은 단순 대용량 데이터의 저장만 의미하는 것이 아니라 가치와 의미가 있는 데이터를 분석하는 범위를 포함한다. 빅데이터 분석에는 다양한 분석 기술이 존재하지만, 분석 대상 데이터의 정제에도 많은 노력을 투자해야 분석 결과에 대한 정확도와 신뢰도를 높일 수 있을 것 이다.

본 논문에서는 빅데이터에 대해 중복 데이터를 고속으로 검출할 수 있는 RUG 알고리즘을 제안하였으며, 기존 알고리즘 대비 획기적인 성능을 개선하는 효과를 보였다. 이는 실제 빅데이터 처리에서 적용할 수 있는 수준이라고 판단된다.

또한, 중복 데이터 검출 기술은 유사 문서 탐색, 표절 검출, 스팸 필터링, 스토리지 내 중복 파일 제거 등 다양한 분야에 응용될 수 있을 것이다.

참고문헌

[1] James Manyika & Michael Chui, “Big data: The next frontier for innovation, competition, and productivity,” McKinsey Global Institute, May 2011.

[2] John Gantz & David Reinsel, “Extracting Value from Chaos,” IDC IVIEW, June 2011.

[3] TDWI Research, “Big Data Analytics Report,” 2011.

[4] Gartner Group, ‘Big Data’ Is Only the Beginning of Extreme Information Management, April 2011.

[5] Apache Hadoop Project, *http://hadoop.apache.org*

[6] The R Project for Statistical Computing, *http://www.r-project.org*

[7] Eric A. Brewer, “Towards robust distributed systems,” Proceedings of the nineteenth annual ACM symposium on Principles of distributed computing, p.4, 2000.

[8] David M. Blei, Andrew Y. Ng, Michael I. Jordan, “Latent Dirichlet Allocation,” The Journal of Machine Learning Research, Vol.3, pp.993-1022, 2003.

[9] 2012 대선 후보 SNS 분석서비스, *http://www.2012president.kr*

[10] A. Broder, S. C. Glassman, M. Manasse, and G. Zweig. “Syntactic clustering of the web,” Computer Networks. 29(8-13):1157-1166, 1997.

[11] T. C. Hoad and J. Zobel. “Methods for identifying versioned and plagiarised documents,” Journal of the American Society for Information Science and Technology, 54(3):203-215, Feb. 2003.

[12] A. Mittelbach, L. Lehmann, C. Rensing, and R. Steinmetz. “Automatic detection of local reuse. In EC-TEL”, volume 6383 of Lecture Notes in Computer Science, pages 229–244. Springer, 2010.

[13] O. Chum, J. Philbin, and A. Zisserman. “Near duplicate image detection: min-hash and tf-idf weighting,” In British Machine Vision Conference, 2008.

[14] M. Charikar. “Similarity estimation techniques from rounding algorithms,” In Proc. 34th Annual Symposium on Theory of Computing (STOC 2002), pages 380-388, 2002.

[15] A. Chowdhury, O. Frieder, D. Grossman, and M. C. McCabe. “Collection statistics for fast duplicate document detection,” ACM Transactions on Information Systems, 20(2):171-191, 2002.

[16] S. Uddin, C.K. Roy, K.A. Schneider, and A. Hindle, “On the Effectiveness of Simhash for Detecting Near-Miss Clones in Large Scale Software Systems,” Proc. WCRE, pp. 13-22, 2011.

[17] 권영현, 윤도현, 안영민, “대표 Unigram 군집화를 통한 유사중복 검출 최적화,” 한국정보과학회 KCC2012, 2012.

저 자 소 개

박 호 진

2000: 한국해양대학교
자동화정보공학부 공학사
2002: 한국해양대학교
컴퓨터공학과 공학석사
현 재: ㈜와이즈넛 연구소 차장
관심분야: 자연언어처리, 정보검색

권 영 현

2008: 건국대학교
인터넷학과 공학사
2010: 건국대학교
신기술융합학부
공학석사
현 재: ㈜와이즈넛 연구소 대리
관심분야: 검색엔진,
클라우드 컴퓨팅

안 영 민

2000: 충북대학교
컴퓨터공학과 공학사
2002: 충북대학교
컴퓨터공학과 공학석사
2008: 충북대학교
컴퓨터공학과 공학박사
현 재: ㈜와이즈넛 연구소 부장
관심분야: 자연언어처리,
데이터 마이닝