감성과학, Vol.15, No.4, pp.425-440, December 2012

맵리듀스를 이용한 통계적 접근의 감성 분류

Statistical Approach to Sentiment Classification using MapReduce

강문수 · 백승희 · 최영식†

Mun-Su Kang · Seung-Hee Baek · Young-Sik Choi†

한국항공대학교 전자 및 정보통신공학부 컴퓨터공학과 IT연구소

Department of Computer Engineering, Korea Aerospace University

Abstract

As the scale of the internet grows, the amount of subjective data increases. Thus, A need to classify automatically subjective data arises. Sentiment classification is a classification of subjective data by various types of sentiments. The sentiment classification researches have been studied focused on NLP(Natural Language Processing) and sentiment word dictionary.

The former sentiment classification researches have two critical problems. First, the performance of morpheme analysis in NLP have fallen short of expectations. Second, it is not easy to choose sentiment words and determine how much a word has a sentiment. To solve these problems, this paper suggests a combination of using web-scale data and a statistical approach to sentiment classification.

The proposed method of this paper is using statistics of words from web-scale data, rather than finding a meaning of a word. This approach differs from the former researches depended on NLP algorithms, it focuses on data. Hadoop and MapReduce will be used to handle web-scale data.

Key words : Sentiment classification, Statistical, Cloud, Hadoop, MapReduce

요 약

인터넷의 규모가 커지면서 주관적인 데이터가 증가하였다. 이에 주관적인 데이터를 자동으로 분류할 필요가 생겼다. 감성 분류는 데이터를 여러 감성 종류에 따라 나누는 것을 말한다. 감성 분류 연구는 크게 자연어 처리와 감성어 사전 구축을 중심으로 이루어져 왔다.

이전의 감성 분류 연구는 자연어 처리 과정에서 형태소 분석이 제대로 이루어지지 않는 문제와 감성어 사전 구축 시 등록할 단어를 선별하고 단어의 감성 정도를 정하는 데에 명확한 기준을 정하기 힘든 문제가 있다. 이러한 어려움을 해결하기 위하여 감성 분류에 대용량 데이터와 통계적 접근의 조합을 제안한다.

본 논문에서 제안하는 방법은 단어의 의미를 찾는 대신 수많은 데이터에서 등장하는 표현들의 통계치를 이용하여 감성 판단을 하는 것이다. 이러한 접근은 자연어 처리 알고리즘에 의존하던 이전 연구와 달리 데이터에 집중한다. 대용량 데이터 처리를 위해 하둡과 맵리듀스를 이용한다.

주제어: 감성 분류, 통계적, 클라우드, 하둡, 맵리듀스

이 논문은 2012년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. 2011-0012297)

† 교신저자 : 최영식 (한국항공대학교 전자 및 정보통신공학부 컴퓨터공학과 IT연구소)

E-mail : choimail@kau.ac.kr

TEL : 02-3158-1419

1. 서론

인터넷에서는 수많은 사용자가 수많은 정보를 생성하고 소비한다. 뉴스와 같이 사실을 기술한 정보부터 일기장과 같이 매우 개인적인 내용의 정보까지 사용자가 생성하고 소비하는 정보의 종류 역시 다양하다. 최근에는 다양한 소셜 네트워크 서비스가 등장하였고 사용자들에게 인기가 높아 소셜 네트워크 서비스에서 매우 많은 양의 정보가 생성되고 소비된다.

소셜 네트워크 서비스를 포함하여 감성을 포함하는 정보들은 사람들의 의견을 나타내는 정보로서 개인이 상품에 대한 의견이나 이미지를 알아보는 일부터 회사에서는 제품에 대한 이미지나 평가를 감성을 포함한 정보를 통해 얻을 수 있다. 따라서 감성을 포함하는 정보를 분석하는 일은 쓰임새가 다양하고 중요도가 높다.

이러한 이유로 감성을 포함한 정보를 분석하고 긍정, 부정을 판단하려는 감성 분석 및 분류에 대한 연구가 많이 이루어졌다. 현재까지의 방법들은 자연어 처리와 감성어 사전 구축을 핵심 요소로 감성 분류를 시도하였다.

이러한 방법으로 감성 분류를 하는 것은 자연어 처리가 잘 이루어지고 좋은 감성어 사전이 있다면 매우 훌륭한 감성 분류 결과를 낼 수 있지만 글, 특히 한글의 특성상 단어의 변형이 매우 많고 문법이 어려우며 단어만으로 문맥을 파악하기 어려워 자연어 처리에서 좋은 결과를 보이기 어려운 점이 있다. 또한 이모티콘과 신조어를 사전에 반영하기 어렵고 단어에 감성 정도를 매긴다는 일이 어려워 사전 구축 역시 쉽지 않다. 이러한 어려움들로 인해 현재까지의 감성 분류는 많은 연구들로 큰 발전이 있었지만 그 발전에 비해 만족할만한 성능 개선은 이루어지지 않았다.

본 논문에서는 대용량 데이터 처리와 통계적 접근 방법으로 위의 문제점들을 해결하려 한다. 형태소 분석과 같은 자연어 처리를 거치지 않고, 수많은 데이터를 읽고 그 데이터에서 등장하는 표현들을 이용해 단어의 감성을 판단하려 한다. 사전 구축 시 사전에서 단어에 대한 긍정, 부정에 대한 점수를 매기는 일은 데이터의 통계치를 이용하여 부여한다. 이러한 접근법은 기존의 문제점들을 해결하면서 시간이 지남에 따라 점차 성능이 좋아지는 감성 분류 시스템을 기대할 수 있다.

본 논문은 다음과 같이 구성하였다. 다음 장에서 관련 연구로 지금까지 진행되어 온 감성 분류 방법과 대용량 데이터 처리 방법에 대해 소개하고, 세 번째 장에서는 제안하는 방법의 아이디어와 시스템 구축을 위한 설계에 대해 설명한다. 네 번째 장에서는 이 시스템을 이용하여 실험한 내용과 그 결과를 설명하고 분석한다. 마지막 다섯 번째 장에서는 본 논문의 결론과 앞으로의 연구 과제에 대해 기술한다.

2. 관련 연구

이번 장에서는 현재까지 진행된 감성 분류의 주요 연구 주제인 자연어 처리와 감성어 사전에 대해 알아보고 대용량 데이터 처리를 위한 프레임워크(Framework)인 하둡(Hadoop)과 맵리듀스(MapReduce)에 대해 알아본다.

2.1. 감성 분류

현재까지 감성 분류 연구의 주된 연구 방향은 글을 분석하여 의미 단위를 찾고, 찾은 의미 단위와 함께 의미 단위의 변형들을 사전에 등록하는 것이다. 판단하고자 하는 글을 입력으로 받으면 사전에 있는 정보들을 조합하여 입력으로 받은 글이 긍정적인지 부정적인지 판단하게 된다. 이에 따라 감성 분류에 관한 연구들은 자연어 처리와 감성어 사전 구축 두 가지에 대한 연구가 활발히 이루어졌다.

2.1.1. 자연어 처리

자연어 처리는 크게 형태소 분석, 구문 분석, 의미 분석 3단계로 나눈다. 이 중 가장 먼저 선행되어야 하는 것이 형태소 분석인데 형태소 분석의 결과에 따라 구문 분석과 의미 분석의 결과도 크게 좌우된다. 따라서 형태소 분석은 자연어 처리에서 가장 핵심이 되는 요소로서 감성 분류 연구들 중 가장 활발히 진행되고 있다.

형태소 분석은 자연어 처리의 가장 기본적이고 핵심적인 과정으로 형태소 분석 이후 이를 토대로 구문 분석이나 의미 분석이 이루어지기 때문에 반드시 먼저 이루어져야 한다. 형태소 분석은 다음과 같은 작업이 복합되어 있다(강승식, 2003).

① 어절을 구성하는 형태소들을 분리한 후에
② 형태론적 변형이 일어난 형태소의 원형을 복원하고
③ 형태소 사전과 분석 규칙에 의해 옳은 분석 후보를 선택한다.

형태소 분석 하는 방법은 크게 3가지로 볼 수 있다.

1) 규칙 기반
2) 통계 기반
3) 규칙과 통계 혼합

규칙 기반 방법에는 중의성 해결 규칙 기반 시스템과 변형 규칙 기반 시스템이 있다. 중의성 해결 규칙 기반 시스템은 태깅(Tagging)하고자 하는 단어의 좌우 문맥을 참조하여 긍정적 또는 부정적으로 중의성을 해결하는 규칙을 사용한다. 변형 규칙 기반 시스템은 초기 태깅 오류를 최소화하기 위해 오류를 올바른 태그로 변형시키는 규칙을 사용한다(강승식, 2003).

규칙 기반 방법으로 형태소를 분석한 감성 분류 연구 중 김명규(2010)는 단어를 품사 단위로 분리하고, 감성 분류 대상의 특성을 분석하여 몇 가지로 정의함으로 분리된 단어의 품사들과 대상의 특성을 조합 및 매칭(Matching)함으로 긍정 및 부정을 판단한다. 아래 표를 통해 품사와 특성의 매칭 패턴을 나타내었다.

Table 1. Matching Patterns between Word Class and Properties of Product

Word Class	Property
Noun Phrase	Entity Product Property Sentiment Word Sense Word
Predicate (Adjective/Verb)	Existence Word Evaluation Word Sentiment Word Sense Word
Adverb	Negative Polar Word Stress Word

이러한 규칙 기반 방법은 형태소 분석을 포함한 자연어 처리가 선행되어야 하는데, 자연어의 특성상 여러 가지 처리하기 힘든 문제점이 존재한다. 문제가 되는 대표적인 원인 몇 가지를 살펴보면 아래와 같다.

1) 언어 파괴 현상
2) 중의성
3) 이모티콘, 신조어

언어파괴 현상은 축약형과 변형, 다양한 어미 등의 자연 언어적 특성에 의한 것으로 경우의 수가 너무나 다양하여 규칙을 적용하기 매우 어렵다. 또한 사전에 등록되어 있는 단어라 하더라도 실제적인 의미가 사전에 등록된 것과 다를 수 있는 중의성 때문에 감성 판단을 제대로 하지 못하는 경우가 많다. 이모티콘이나 신조어는 사용자가 글을 쓸 때 매우 자주 사용하고 감성을 담고 있는 정도가 큰 경우가 많기 때문에 감성 분류에 매우 중요한 요소이지만 규칙을 통한 판단 방법으로는 이를 반영하거나 처리하기 쉽지 않다.

통계 기반 방법은 확률모델을 사용하는 시스템과 신경망 및 퍼지망을 사용하는 시스템 등이 있는데, 대부분의 형태소 분석은 규칙 기반에 통계 기반을 혼합하여 사용한다.

규칙과 통계의 혼합 버전의 연구에서는 통계적 자료를 활용하여 이 문제들을 보완하지만 이 역시 형태소 분석을 통해 품사 단위로 형태소 분리를 한 후에 이루어지는 과정이어서 형태소 분석에서 발생하는 오류를 해결하기는 힘들다. 이를 위해 별도의 시스템을 구성하거나 복수의 사전을 구성하는 등의 대안이 있지만 이는 시스템 구성이 지나치게 비대해질 수 있는 문제를 안고 있다(심준혁 등, 2000).

2.1.2. 감성어 사전

현재까지 대부분의 감성 분류 연구에서 쓰인 감성어 사전은 자연어 처리와 함께 생성되어 명사, 동사, 형용사의 품사를 가지는 단어들을 사전에 저장한다. 사전에 등록되는 단어들은 자연어 처리 과정을 통해 얻은 단어와 이 단어의 변형인데, 어떠한 단어를 사전에 넣을지, 어떠한 변형을 적용할지 등의 기준은 사람이 임의로 정한다.

감성 분류 연구에서 사용하는 감성어 사전은 다음과 같은 방식들로 구축한다.

홍진표와 차정원(2008)은 트레이닝 데이터에서 형태소 분석을 통해 구축한 단어 모음을 구축하고 이 단어들의 다양한 변형을 적용하여 사전에 저장한다. 이 때 어떠한 변형을 적용할 것인지는 형태소 패턴 분석을 통해 빈도수가 높은 순서대로 적용한다.

또 다른 방법으로 명재석 등(2008)은 감성 분류를 하려는 대상에 맞게 추가 사전을 구축하는 것으로 대상의 특징을 기술하는 어휘 사전과 언어적 측면의 부가적인 정보를 가지는 부가 어휘 사전을 구축하는 방식이다. 기술 어휘 사전은 대상의 분류나 주제어, 서술어 등을 구조화하여 저장하고, 부가 어휘 사전은 '매우', '조금' 과 같이 의미적인 강도에 영향을 주는 언어적 측면의 어휘들을 저장한다. 사전에 단어를 추가할 때는 어휘의 빈도수에 근거하여 사전에 등록한다.

감성 분류에서 감성어 사전을 구축하는 작업은 한글에서뿐만 아니라 영어에 대해서도 활발히 이루어진다. 영어 감성어 사전 역시 위의 연구들과 유사하게 기준이 되는 단어 모음을 임의로 정하고 이 단어 모음에서 변형과 유의어 등을 사전에 추가하는 방식으로 사전을 구축한다(Bracewell, D.B., 2010).

사전 구축에는 방법들이 조금씩 다르지만 핵심적인 요소이자 문제점을 다음과 같이 정리할 수 있다.

1) 형태소 분석이 필요
2) 사람의 수동적인 작업이 필요
3) 사전에 등록할 단어를 선정하기 어려움
4) 신조어나 이모티콘 등을 반영하기 어려움

사전에 단어를 추가하기 위해서는 문장의 표현들을 형태소 단위로 분리하고 여러 가지 변형과 조합을 등록하는 방식이어서 형태소 분석 과정이 사전에 반드시 필요하게 된다. 형태소 분석이 들어가게 되면 2.1.1에서 보인 문제점들을 다시 포함하게 되어서 사전 구축에서 역시 그와 유사하거나 연관된 문제점들이 나타날 수 있다.

또한 사전을 구축할 때 사람이 어떤 단어를 등록할지를 결정하기 때문에 반드시 수동적인 작업이 필요하고 이와 함께 사전에 등록할 단어를 선정하는 데에 어려움이 생길 수 있다.

마지막으로 새롭게 등장하는 어휘나 이모티콘과 같은 표현들은 사전에 반영하기 어렵고 이를 처리하기 위해 또 다른 모듈이나 사전을 두는 식의 방법을 취하기 때문에 시스템이 비대해질 수 있는 문제점이 있다.

2.2. 대용량 데이터 처리

급속히 늘어나는 데이터로 인해 클라우드 컴퓨팅이라는 이름으로 대용량 데이터 처리에 관한 연구가 매우 활발히 진행되고 있다. 하둡은 대표적인 대용량 데이터 처리 프레임워크로 아파치 오픈 소스 프로젝트이다. 하둡과 함께 대용량 데이터를 처리하기 위한 분산 병렬 처리 프레임워크로는 구글의 맵리듀스(MapReduce)가 있다.

2.2.1. 하둡

하둡은 아파치의 오픈 소스 프로젝트로 구글의 구글 파일 시스템(GFS)에 대응하는 오픈 소스 진영의 분산 파일 시스템이다(Tom White, 2010).

하둡은 여러 대의 컴퓨터를 클러스터로 구성하고 이를 한 대의 컴퓨터처럼 사용하는 개념이다. 컴퓨터들을 클러스터로 묶어 대용량 저장 공간과 높은 컴퓨팅 리소스를 가능하게 한다. 저장 공간과 컴퓨팅 리소스로 하둡은 다음의 요구 사항이 필요한 곳에 사용될 수 있다(Tom White, 2010).

▷ 파일 사이즈가 큰 대용량 파일을 처리
▷ 낮은 응답 지연 시간보다 높은 처리율이 중요
▷ 클러스터 구축 비용이 비교적 저렴

하둡은 대용량 데이터 처리를 위해 HDFS라는 분산 파일 시스템을 제공한다. HDFS는 Hadoop Distributed File System의 약어로 일반적인 목적의 파일 시스템을 추상화하여 대용량 스토리지(Storage) 시스템을 제공한다(Tom White, 2010).

HDFS는 하둡의 분산 파일 시스템으로 아래와 같은 대용량 데이터 처리의 특징을 가진다.

1) 매우 큰 파일
2) 스트리밍 방식의 데이터 액세스
3) 범용 하드웨어

하둡은 매우 큰 파일들을 많은 양의 데이터를 쓰고 이를 분석하기 위한 용도로 많이 이용한다. 하둡 클러스터 구축은 주로 범용 하드웨어를 이용하는데 이는 저가의 노드 장애가 발생할 확률이 높은 범용 하드웨어 클러스터에서 실행되도록 설계되었기 때문이다(Shvachko, K. 등, 2010).

하둡 분산 파일 시스템은 마스터와 슬레이브로 노드를 구분되어 마스터는 파일 시스템의 메타 데이터를, 슬레이브는 실제 파일과 디렉토리를 저장한다. 마스터와 슬레이브를 하둡에서는 네임 노드(Name node)라 하고 슬레이브를 데이터 노드(Data node)라고 한다(Tom White, 2010). 이를 아래의 그림과 같이 도식화할 수 있다.

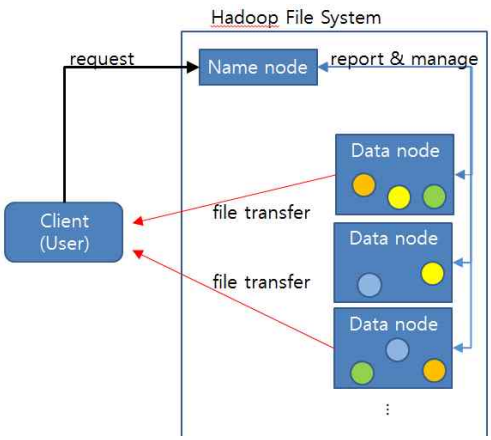


Figure 1. Hadoop File System

2.2.2. 맵리듀스

맵리듀스는 대용량 데이터를 병렬로 처리하기 위한 프레임워크이자 프로그래밍 모델이다. 맵리듀스를 이용하여 방대한 데이터 처리 작업을 여러 노드로 분할하여 병렬로 처리할 수 있다. 맵리듀스는 map과 reduce 두 단계로 프로그램을 구성하며 맵리듀스 프로그래밍은 map과 reduce의 내용을 구현하는 것이다. Map과 reduce는 키-값 데이터를 받아 각 단계에 정의된 내용을 처리한 후 키-값 쌍 형태의 데이터를 출력한다(Dean & Ghemawat, 2008).

맵리듀스의 구조는 하둡 분산 파일 시스템의 구조인 마스터와 슬레이브 구조로 되어 있다. 마스터는 프로그램을 슬레이브 노드들에 할당하고 이를 관리한다. 마스터와 슬레이브는 맵리듀스에서 각각 잡트래커(Job tracker), 태스크트래커(Task tracker)로 부른다. 즉, 잡트래커는 사용자에게 프로그램을 받아 이를 태스크트래커들에게 분배하고 전체적인 작업을 관리한다. 태스크트래커들은 map 또는 reduce를 수행한다(Dean & Ghemawat, 2008). 이를 도식화하면 아래와 같다.

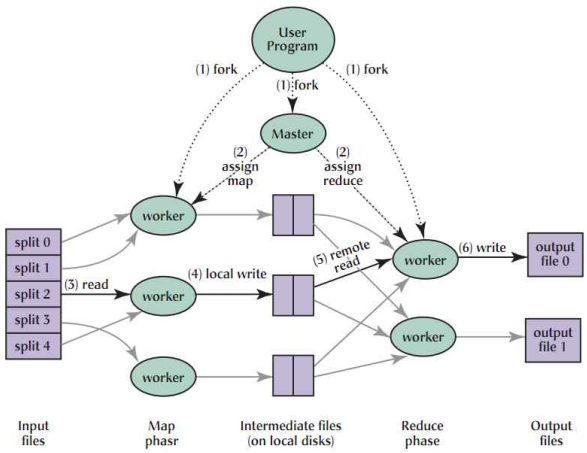


Figure 2. MapReduce Execution Diagram(Dean & Ghemawat, 2008)

3. 맵리듀스를 이용한 감성 분류

이번 장에서는 맵리듀스를 이용한 통계적 접근의 감성 분류를 위한 방법을 제안하고 그에 대한 설계를 보인다. 먼저 감성 분류를 하기 위한 사전 구축과 감성 판단의 기준을 설정한다. 관련 연구에서 보인 감성 분류는 자연어 처리와 잘 정제된 감성어 사전을 이용하지만 본 논문에서 제안하는 방법은 자연어 처리를 거치지 않고 모든 단어를 감성어로 인식하여 사전을 구축하고 이를 통계적 수치에 근거하여 감성을 판단한다.

3.1. 제안하는 감성 분류 방법

본 논문에서 제안하는 감성 분류 방법은 관련 연구에서 보인 감성 분류 방법과 다음의 두 가지 큰 차이를 보인다. 첫 째, 형태소 분석을 포함 자연어 처리를 하지 않는다. 둘 째, 데이터에 등장하는 모든 단어가 사전에 등록된다.

관련 연구에서 보인 감성 분류는 데이터를 자연어 처리를 통해 형태소 단위로 단어를 분리하고 이를 잘 정제된 감성어 사전을 이용하여 감성을 판단한다. 본 논문의 제안은 자연어 처리를 하지 않고 통계에 근거하는 것이 감성 판단에 더 효과적이라고 가정한다.

3.1.1. 사전 구축

관련 연구에서 보인 사전들은 형태소 분석을 거친 단어의 원형을 사전에 등록하고 원형과 변형들을 사용한다. 본 논문에서 제안하는 시스템의 사전은 이와 다르게 데이터에서 등장하는 모든 표현을 자연어 처리를 거치지 않고 사전에 등록한다. 따라서 사전에 등록한 단어들은 형태소 단위가 아닐 수도, 단어의 변형이 아닐 수도 있다. 사람들이 쓰는 표현들을 모두 감성어로 여겨 단어들을 모두 사전에 등록하는 것이다.

이러한 방식으로 사전을 구축하면 아래와 같은 사전 예를 보일 수 있을 것이다.

Table 2. Dictionary of Former Research

Dictionary
재미있다
즐겁다
슬프다
우울하다
⋮

Table 3. Dictionary of all the representations

Dictionary
재밋어요
이게왜
^^;;
안습
⋮

위의 두 사전 중 왼쪽은 관련 연구에서 보인 잘 정제된 감성어 사전을 보인 것이고 오른쪽은 본 논문에서 제안하는 방식으로 구축한 사전의 예다. 왼쪽의 사전은 사람들이 감정 표현에 사용하는 형용사들을 등록한 사전이다. 반면 오른쪽의 사전은 사람들이 표현

한 단어 그대로를 사전에 등록하여 '재밋어요'와 같이 맞춤법이 틀린 표현이나 '이게왜' 같이 문맥적으로 애매한 표현, 이모티콘 '^^;;', 신조어 '안습' 등이 사전에 등록될 수 있다.

이러한 방식으로 사전을 구축하는 이유는 앞서 관련 연구들의 감성 분류에서 자연어 처리의 성능이 감성 분류의 성능을 좌우하기 때문이다. 자연어 처리가 잘 되는 시스템에서는 감성 분류가 매우 잘 이루어질 수 있지만 앞선 연구들에서 자연어 처리의 성능을 살펴보면 만족할만한 결과를 내지 못한다. 이는 사람이 작성한 글은 매우 다양한 변수와 경우의 수 때문에 이를 규칙으로 정해두기 어렵기 때문이다.

자연어 처리를 하지 않는 또 다른 이유는 위처럼 좋은 결과가 나오기 어려운 이유도 있지만 자연어 처리로 문맥적 의미 파악이 어려워지기 때문도 있다. 사람들이 쓰는 표현은 그 자체로 뉘앙스와 의미를 포함하고 있는데 자연어 처리는 이를 제거하고 단어의 원형을 찾아가기 때문에 표현의 의미를 놓칠 수 있다.

사전에 단어를 저장할 때는 이후에 감성 분류를 하기 위한 통계적 근거를 저장하고 있어야 하기 때문에 단어의 빈도수를 단어와 함께 저장한다. 단어가 긍정적인 내용의 데이터에서 나왔을 경우 각 단어에 대한 긍정 빈도수에 1을 더하고 부정적인 내용의 데이터에서 나오는 경우에는 부정 빈도수에 1을 더한다.

입력 데이터가 긍정적인 내용의 데이터인지 부정적인 내용의 데이터인지를 정하는 기준은 데이터의 특성에 따라 달라진다. 예를 들어 영화평을 입력 데이터로 받는다면 영화에 대한 점수를 기준으로 각 영화평을 긍정인지 부정인지로 나눌 수 있다. 점수의 기준을 정할 때는 데이터를 살펴보면서 데이터 셋의 특성으로 판단하거나 임의의 실험으로 가장 적절한 점수 기준을 정할 수 있다.

이러한 방법으로 구축한 실제 사전의 예를 아래에서 볼 수 있다.

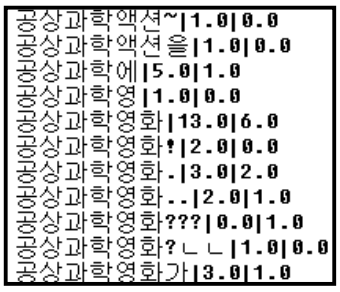

공상과학액션~|1.0|0.0
공상과학액션을|1.0|0.0
공상과학에|5.0|1.0
공상과학영|1.0|0.0
공상과학영화|13.0|6.0
공상과학영화!|2.0|0.0
공상과학영화.|3.0|2.0
공상과학영화..|2.0|1.0
공상과학영화???|0.0|1.0
공상과학영화?ㄴㄴ|1.0|0.0
공상과학영화가|3.0|1.0

Figure 3. A Sample of Dictionary

단어, 긍정 빈도수, 부정 빈도수를 가지는 사전으로 '|'를 구분자로 사용한 사전 예다.

3.1.2. 감성 판단 방법

감성 분류에서 감성은 매우 다양한 종류가 있지만 본 논문에서는 긍정적인 내용과 부정적인 내용 두 가지로 분류 기준을 설정한다. 따라서 본 논문의 감성 분류는 데이터 글이 긍정적인 내용인지 부정적인 내용인지를 분류하는 것이 된다.

감성 판단은 사전을 생성할 때 저장한 단어의 긍정 빈도수와 부정 빈도수를 근거로 한다. 글에서 쓰인 단어의 빈도수를 이용하여 데이터 글이 긍정 내용일 확률과 부정 내용일 확률을 각각 구하여 비교함으로 글의 감성을 판단한다.

본 논문에서 감성 판단을 위한 방법은 긍정 빈도수와 부정 빈도수의 비율이다. 빈도수를 비율로 표현한 이유는 긍정 빈도수와 부정 빈도수가 모두 높은 단어들의 비중을 낮추기 위함인데, 이는 '매우'과 같은 단어는 긍정 빈도도 높고 부정 빈도도 높아 이 단어가 긍정, 부정을 판단하는 데에 도움이 되지 못하고 오히려 역효과를 가져올 수 있다는 문제점을 해결할 수 있다. 이에 대한 식은 아래와 같다.

$$\frac{\text{긍정(부정)빈도수 합}}{\text{긍정 빈도수 합} + \text{부정 빈도수 합}} \tag{1}$$

긍정, 부정 빈도수의 합은 판단해야 할 데이터 글에 등장한 단어들의 각각 긍정과 부정의 빈도수를 모두 더한 것을 말한다. 위와 같이 식을 변형함으로 단어가 긍정의 특성을 강하게 나타내는지 부정의 특성을 강하게 나타내는지 알 수 있다. 데이터의 긍정, 부정 판단은 위의 식을 이용해 아래와 같이 표현한다.

fp_w 단어 w의 긍정 빈도수라 하고, fn_w를 부정 빈도수라고 한다면,

$$\sum_{w \in \text{글에 등장한 단어 집합}} \frac{fp_w - fn_w}{fp_w + fn_w} \tag{2}$$

식2에서 $fp_w - fn_w$이 양수이면 긍정, 음수이면 부정, 0이면 판단 불가로 처리한다.

위의 식2는 한 단어의 빈도수 분포에 대한 고려는 있지만 전체 데이터 셋에서 긍정 내용의 글과 부정 내용의 글의 비율에 대한 고려는 없다. 데이터 셋에는

긍정 내용의 글과 부정 내용의 글이 1:1로 존재하는 것이 아니어서 긍정 내용의 글이 다수이면 긍정 빈도수가 자연스레 증가하여 감성 판단을 잘못할 수 있게 된다. 이를 보완하기 위해 긍정 내용 글과 부정 내용 글의 비율을 빈도수에 스케일링 해줌으로써 해결한다. 스케일링 방법은 아래와 같다.

$$\text{긍정빈도수합} \cdot \frac{\text{부정문서개수}}{\text{긍정문서개수}} \quad (3)$$

부정 빈도수에 대한 스케일링은 위 식에서 긍정과 부정을 바꾸어 적용하였고, 위의 방법을 통해 긍정 문서 개수가 많다면 긍정 빈도수 합은 감소하고 부정 문서 개수가 많다면 긍정 빈도수 합은 증가할 것이다.

긍정, 부정 빈도수 값에 대한 고려 역시 필요하다. 가장 기본적인 빈도수를 구하는 방법은 문서에 단어가 등장할 때마다 1을 더해주는 것이지만 이는 각 점수대별로 데이터를 고려하지 않는 방법이다. 최고점이 10점, 최저점이 0점인 평가에서 10점 데이터와 8점 데이터는 긍정 정도에서 분명 차이를 보인다. 부정 내용 데이터 역시 0점과 4점은 그 정도에서 차이를 보인다. 이를 고려하여 빈도수를 더할 때 10점이나 0점의 데이터에서 등장하는 경우에는 가중치를 9점이나 1점에 주는 것보다 더하는 방법을 제안한다. 각 점수대별 가중치 값은 아래와 같고, 가중치 값은 실험을 통해 구하였다. 5점에서 7점 사이의 값은 긍정이나 부정의 감성으로 보기 힘든 애매한 정도를 보이기 때문이다. 애매한 감성의 데이터에서 등장하는 표현들 역시 긍정이나 부정의 감성을 지니기 어렵다고 판단하였기 때문이다.

❖ 긍정

Table 4. Weighting on Positive Reviews

10 Points	1.500
9 Points	1.250
8 Points	1.000

❖ 부정

Table 5. Weighting on Negative Reviews

0 Point	1.500
1 Point	1.375
2 Points	1.250
3 Points	1.125
4 Points	1.000

감성 정도가 가장 강한 10점과 0점, 감성 정도가 가장 약한 8점과 4점 사이의 감성 정도 차이가 2배 이하라고 가정한 후 값을 바꿔가며 실험을 하였더니 감성 정도가 가장 강한 10점과 0점은 1.500을, 감성 정도가 가장 약한 8점과 4점은 1.000을 가중치로 주고 점수간 차등을 같게 하였을 때 결과 값이 높음을 확인하였다.

최종적으로 글의 긍정, 부정을 판단할 때는 글에 등장한 단어들의 긍정, 부정 값을 위의 방법으로 구축한 사전에서 참조하여 합산하게 된다. 각 단어들은 스스로의 긍정 통계치, 부정 통계치를 가지며 이 값들의 조합으로 최종적으로 글의 긍정, 부정을 판단한다.

3.2. 분산·병렬 처리

위에서 제안한 감성 판단 방법은 자연어 처리를 하지 않고 등장한 표현 그대로를 사용하기 때문에 데이터가 많아질수록 판단 결과가 좋아진다고 가정한다. 이를 위해 많은 양의 데이터를 처리해야 하기 때문에 빠른 처리 속도를 위해 분산·병렬 처리가 필요하다. 본 논문은 관련 연구에서 보였던 하둡과 맵리듀스를 이용하여 이를 해결한다. 하둡은 대용량 데이터를 저장하고 관리하는 기능을, 맵리듀스는 수행 작업을 분할하여 빠른 속도로 계산하는 기능을 한다.

하둡 분산 파일 시스템은 일반 파일 시스템을 사용하는 것과 유사하게 사용할 수 있어서 큰 무리 없이 적용 가능하지만 수행할 작업은 분산하여 처리해야 하기 때문에 맵리듀스 프로그래밍 방식에 맞추어 작성해야 한다.

본 논문에서 제안하는 감성 분류를 분산·병렬 처리가 되도록 맵리듀스 프로그래밍으로 작성하여 시스템

을 구성하려 한다. 구성하려는 감성 분류 시스템은 앞서 제안한 사전 구축과 감성 판단 및 평가 모듈을 가진다.

구축한 사전을 이용하여 감성 판단을 내리는 작업은 글 데이터를 이용하기 때문에 스트링(String) 데이터 처리가 필요하다. 본 시스템은 대용량의 데이터를 다루기 때문에 스트링을 처리하는 작업에서 드는 비용은 데이터가 많아질수록 커진다. 이 비용을 낮추기 위해 사전 구축과 감성 판단 및 평가 사이에 전처리 과정을 더한다. 전처리 과정에서는 스트링을 사전의 인덱스(Index)로 치환하는 작업을 수행한다. 이를 통해 감성 판단 및 평가 작업에서는 스트링을 검색하고 비교하는 작업 대신 인덱스를 이용해 데이터에 곧바로 접근할 수 있다.

맵리듀스는 프로그래밍 프레임워크로 프로그램에 쓰이는 데이터 타입 역시 맵리듀스 프레임워크에서 제공하는 데이터 타입을 사용한다. 감성 판단을 위해 구축하는 사전은 맵리듀스에서 제공하는 데이터 타입으로 표현할 수 없기 때문에 새롭게 데이터 타입 클래스를 정의해줄 필요가 있다. 사전은 사전 구축 제안에서 보인 바와 같이 단어와 긍정 빈도수, 부정 빈도수를 저장하는 데이터의 모음이다. 이 데이터 타입을 새로운 클래스로 정의하고 이 클래스의 어레이리스트(ArrayList)로 사전을 표현한다. 정의할 데이터 타입 클래스는 단어, 긍정 빈도수, 부정 빈도수를 멤버로 가지고 필요한 함수들을 정의하거나 오버라이딩(Overriding) 하여야 한다. 기본적으로 정의하여야 하는 함수는 생성자, setter, getter이고 오버라이딩 하여야 하는 함수는 write, readFields, hashCode, equals, toString, compareTo 함수 등이다. 오버라이딩하는 함수들은 이 클래스가 맵리듀스에서 비교 가능하고 읽고 쓰기 가능한 데이터 타입이 되기 위한 조건이다 (Lin & Dyer, 2010).

3.3. 시스템 구성

이번 절에서는 위에서 기술한 내용을 도식화하고 각 모듈별로 좀 더 자세히 설명한다. 각 모듈은 맵리듀스 작업으로 수행되기 때문에 모듈별 설명은 맵과 리듀스 함수를 중심으로 기술한다.

3.3.1. 사전 구축

Map

입력

key − offset | value − 글 데이터

출력

key − 단어 | value − 긍정도 혹은 부정도

함수 내용

▷ 글 데이터를 단어 단위로 분리

▷ 글 데이터의 긍정도 또는 부정도 판단

▷ 단어를 key로 긍정도 또는 부정도를 출력

Reduce

입력

key − 단어 | value − 긍정도 또는 부정도

출력

key − null | value − 단어, 긍정도 합, 부정도 합

함수 내용

▷ 입력으로 온 단어의 긍정도 또는 부정도를 합산

▷ 단어, 긍정도 합, 부정도 합을 출력

프로그램 수행이 모두 끝나면 결과로 사전을 출력한다. 이 작업에서 생성한 사전으로 이후 전처리와 감성 판단 및 평가를 수행한다.

3.3.2. 전처리

Map

입력

key − offset | value − 글 데이터

출력

key − 임의의 값 | value − 인덱스로 치환한 데이터

함수 내용

▷ 데이터의 단어들을 사전의 인덱스로 치환

▷ 치환된 글 데이터를 출력

Reduce

입력

key − 임의의 값 | value − 인덱스로 치환한 데이터

출력

key − null | value − 인덱스로 치환한 데이터

함수 내용

▷ 입력으로 받은 내용을 그대로 출력

맵에서 수행하는 내용 중 단어들을 사전의 인덱스로 치환하기 위해 사전 데이터가 로드(load)되어야 한

다. 사전 데이터 로드는 맵 함수가 실행되는 노드에서 최초 한 번 실행되어 사용한다.

리듀스의 함수 내용은 아무 작업 없이 받은 내용을 그대로 출력하여서 리듀스 함수가 없어도 되는 것처럼 보인다. 여기서 리듀스 함수를 두는 이유는 맵에서 생성한 결과들을 하나의 파일로 만들어 주기 위함이다. 리듀스 함수 없이 맵으로 끝나는 프로그램은 맵의 개수만큼 결과 파일이 생성되어서 관리가 어렵기 때문이다.

3.3.3. 감성 판단 및 평가

Map

입력

key − offset | value − 인덱스로 치환한 데이터

출력

key − 판단 결과 평가 | value − null

함수 내용

▷ 인덱스로 치환된 데이터를 단어 단위로 분리

▷ 사전에서 각 단어의 긍정, 부정 빈도수 가져오기

▷ 긍정, 부정 빈도수를 수식에 적용하여 긍정, 부정 판단

▷ 판단 결과를 평가하여 출력

Reduce

입력

key − 판단 결과 평가 | value − null

출력

key − 판단 결과 | value − 판단 결과 합산 값

함수 내용

▷ 판단 결과 평가 개수를 합산

맵에서 긍정, 부정을 판단한 후 판단이 맞는지 틀린지의 평가는 글 데이터가 가지고 있는 점수 등을 이용한다. 판단 결과 평가는 리듀스에서 합산하여 F-measure와 accuracy 측정에 사용한다.

3.3.4. 시스템 구성도

위의 각 모듈들을 하나의 시스템으로 표현하면 아래와 같이 도식화 할 수 있다.

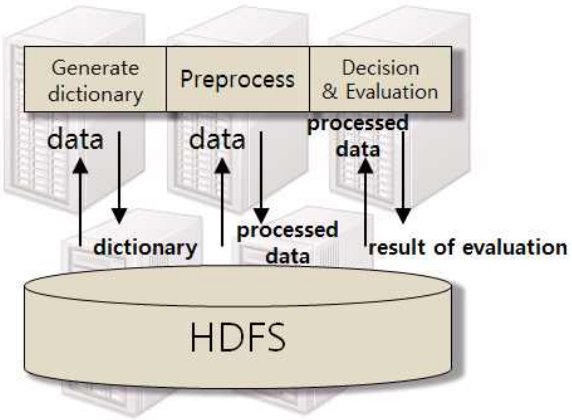


Figure 4. System diagram

데이터 저장 및 관리는 하둡 분산 파일 시스템을 통해 이루어지고 프로그램 수행은 맵리듀스 프레임워크를 통해 이루어진다. 하둡 분산 파일 시스템과 맵리듀스는 둘 다 서버들의 묶음인 클러스터에서 동작한다.

4. 실험 및 분석

이번 장에서는 위의 설계대로 제안한 시스템을 데이터에 실험하고 그 결과를 분석한다.

4.1. 실험 데이터

실험에서 사용할 데이터는 감성을 포함한 글 데이터이다. 실험에서 사용할 감성을 포함한 글은 글 내용에 사용자의 감성이 들어 있고 그 감성을 판단할 수 있는 기준이 있는 데이터를 말한다. 이를테면 영화평이나 상품 리뷰는 영화 혹은 상품에 대한 감성이 들어 있고 대상에 대한 감성 정도를 점수로 부여하기 때문에 실험 데이터로 적절하다.

실험에는 영화평 데이터를 사용하였으며 데이터의 출처는 포털 사이트 네이버와 다음의 영화평이다. 데이터의 통계는 아래와 같다.

Table 6. Statistics of Movie Reviews

	NAVER	DAUM	Sum
Num	3,944,941	564,607	4,509,548

네이버와 다음의 영화평 데이터는 0~10의 점수를 각 영화평에 부여하였다. 0~10점에서 긍정과 부정으로 나누는 점수 기준은 다음과 같이 정하였다.

❖ 긍정 : 8 ~ 10점
❖ 부정 : 0 ~ 4점

위의 점수대를 적용하여 전체 영화평 데이터에서 5~7점대 데이터를 제외한 긍정 데이터와 부정 데이터의 개수는 아래와 같다.

Table 7. Statistics of Movie Reviews to be used

Positive	Negative	Sum
3,101,364	771,241	3,872,605

실험은 위의 데이터를 사용한다.

4.2. 실험 환경

실험에 쓰인 클러스터는 아래와 같은 환경이다.

Table 8. Experiment Conditions

	Condition
Hadoop Version	0.20.2+737 Cloudera's dist.
HDFS Capacity	12.44 TB
Num of HDFS Masters	2
Num of HDFS Slaves	14
Num of MapReduce Masters	1
Num of MapReduce Slaves	11

하둡은 클라우데라 사이트에서 자체 배포하는 클라우데라 하둡 배포 버전이다. HDFS 마스터는 2대로 하나는 네임 노드 역할을 하고 다른 하나는 세컨더리 네임 노드(Secondary name node)로서 네임 노드의 정보를 백업하는 역할을 수행한다. HDFS와 맵리듀스 프로세스는 한 서버에서 동작할 수도, 각기 다른 서버에서 동작할 수도 있으며 총 사용하는 서버는 16대이다.

4.3. 평가 방법

감성 판단의 평가는 F-measure(F1 score)를 사용한다. F-measure는 테스트의 정확도를 구하는 측정법이며 정확도(precision)과 재현율(recall)을 통해 값을 구한다. 정확도와 재현율은 분류의 측면에서 아래와 같이 정의한다.

Table 9. Evaluation Criteria of Classification

		Actual Class	
		positive	**negative**
Classified Class	**positive**	true positive	false positive
	negative	false negative	true negative

위 정의에서 True Positive ~ False Negative는 아래의 의미를 가진다.

1) True positive : positive 판단이 맞음
2) True negative : negative 판단이 맞음
3) False positive : positive 판단이 틀림
4) False negative : negative 판단이 틀림

위 4가지 판단 결과를 통해 정확도 재현율 식을 아래와 같이 쓸 수 있다.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (4)$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (5)$$

위의 정확도와 재현율은 positive를 제대로 분류하였는지 판단할 때 쓰이는 식이다. 본 논문의 감성 분류 시스템은 긍정 분류와 부정 분류 모두를 고려하기 때문에 부정 분류에 대한 판단 식도 구하여야 한다. Negative의 정확도와 재현율 식은 다음과 같다.

$$Precision = \frac{True\ Negative}{True\ Negative + False\ Negative} \quad (6)$$

$$Recall = \frac{True\ Negative}{True\ Negative + False\ Positive} \quad (7)$$

Positive와 negative의 정확도, 재현율 값은 감성 판단 및 평가 작업의 결과로 출력되는 True Positive ~ False Negative를 통해 구할 수 있다. 이 결과를 이용해 정확도와 재현율을 계산하고 이 값을 이용해 F-measure를 구한다. F-measure 식은 아래와 같다.

$$F = 2 \cdot \frac{\mathrm{Precision} \cdot Recall}{\mathrm{Preciision} + Recall} \quad (8)$$

정확도와 재현율이 positive와 negative 각각 도출되기 때문에 F-measure 값 역시 positive, negative가 따로 계산한다.

True Positive ~ False Negative를 이용해 F-measure와 함께 accuracy 값도 구한다. Accuracy는 전체 판단 결과 중 얼마나 제대로 판단하였는지를 보여준다. Accuracy 식은 아래와 같다.

$$Accuracy = \frac{(1)+(2)}{(1)+(2)+(3)+(4)} \quad (9)$$

판단은 위와 같이 정확도, 재현율, f-measure, accuracy로 평가한다.

4.4. 실험

실험은 영화평 데이터를 통해 사전을 구축하고 이 사전을 통해 데이터의 전처리 작업을 수행한다. 전처리 작업을 거친 영화평 데이터를 감성 판단 및 평가 작업의 입력으로 주어 판단 결과를 평가한다.

실험은 총 3가지로 크로스 확인(Cross validation), 나이브 베이지언 모델과 비교 실험, 기존 연구와의 비교 실험이다.

4.4.1. 크로스 확인

성능 검증을 위해 영화평 데이터에 대해 크로스 확인을 실시하였다. 데이터를 중복 없이 20개로 나누어 테스트로 1개, 트레이닝으로 19개를 사용한다. 총 20번의 실험을 수행하였고 아래의 결과는 20번 실험의 평균이다.

Table 10. Result of Cross Validation Experiments

False Negative		7,810.05
False Positive		10,468.5
True Negative		25,350.05
True Positive		137,268.4
ambiguous		53.3
sum		180,897
precision	**positive**	0.929141
	negative	0.764476
recall	**positive**	0.946166
	negative	0.707734
f-measure	**positive**	0.937576
	negative	0.735009
accuracy		0.898691

정확도는 평균 약 89.87%를 보였다. 이 수치는 모호한(ambiguous) 수치를 false로 처리하였을 때의 값이다.

4.4.2. 나이브 베이지언 모델과 비교

분류에서 가장 대표적인 모델인 나이브 베이지언(Naive Bayesian)과의 실험 결과 비교를 통해 본 논문에서 제안하는 감성 판단 방법의 성능과 타당성을 입증하려 한다. 먼저 나이브 베이지언을 본 감성 분류에 어떻게 적용하는지 설명한 후 실험 결과를 비교한다.

나이브 베이지언은 베이지언 모델에 특정한 가정을 적용하였을 때를 말한다. 아래는 베이지언 모델의 식이다.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (10)$$

이 때 A를 클래스, B를 데이터라고 할 때, 데이터 B는 단어의 모음이기 때문에 다음과 같이 치환할 수 있다.

$$P(A|W_1, W_2, W_3, ..., W_n) = \frac{P(W_1, W_2, W_3, ..., W_n|A)P(A)}{P(W_1, W_2, W_3, ..., W_n)} \quad (11)$$

$W_1 \sim W_n$은 단어를 표현한다. 이 수식에서 각 단어들의 출현이 서로 독립이라고 가정하는 것이 나이브 베이지언 모델이다. 독립을 가정한 후 식을 표현하면 다음과 같다.

$$P(A|W_1, W_2, W_3, ..., W_n) = \frac{P(W_1|A)P(W_2|A)\cdots P(W_n|A)P(A)}{P(W_1, W_2, W_3, ..., W_n)} \quad (12)$$

데이터를 긍정 또는 부정으로 분류하는 일은 수식에서 클래스인 A가 긍정일 때와 부정일 때의 값을 비교하는 것이므로 수식의 우변 분모는 같은 값이기 때문에 생략할 수 있다. 결과적으로 식은 아래와 같이 쓸 수 있다.

$$P(A|W_1, W_2, W_3, ..., W_n) = \prod_{i=1}^{n} P(W_i|A)\cdot P(A) \quad (13)$$

$P(A)$, $P(W_i|A)$ 각각은 아래와 같이 구할 수 있다.

$$P(A) = \frac{N_A}{N_A + N_{A'}} \quad (14)$$

$$P(W_i|A) = \frac{T_{AW}}{\sum_{W' \in V} T_{AW'}} \quad (15)$$

식16에서 N_A는 A 클래스의 문서 수이고 식17에서 T_{AW}는 A클래스에서 단어 W의 출현 빈도 수, $\sum_{W' \in V} T_{AW'}$는 A클래스에 등장하는 모든 단어들의 출현 빈도 수 합이다(Manning 등, 2007).

위와 같이 정리한 나이브 베이지언 모델을 감성 판단 방법으로 크로스 확인을 적용하여 다음의 실험 결과를 얻었다.

Table 11. Result of Naive Bayesian Experiments

False Negative	3008.43
False Positive	21897.6
True Negative	12771.14
True Positive	140533.9
ambiguous	3028.14
sum	178211
precision	**positive** 0.865189
	negative 0.809349
recall	**positive** 0.979041
	negative 0.368381
f-measure	**positive** 0.918601
	negative 0.506304
accuracy	0.845871

실험 결과 약 84.60%의 정확도를 보였다. 4.4.1 항에서 제안한 방법으로 보인 크로스 확인 평균 정확도 값과 비교하면 다음과 같다.

Table 12. Comparison of Proposed Method and Naive Bayesian

	Proposed method	**Naive Bayesian**
Accuracy	0.898691	0.845871

본 논문에서 제안하는 감성 판단 방법은 나이브 베이지언 방법과 비교하여 정확도에서 약 5.3% 정도 높은 결과를 보인다. 정확도, 재현율, f-score 값에서도 나이브 베이지언의 결과는 positive와 negative의 차이가 심한데 비해 본 논문의 방법은 비교적 차이 적다. 판단하지 못하는 모호한 수치에서 역시 본 논문은 평균 53.3개지만 나이브 베이지언 방법은 3,028개를 판단하지 못한 것으로 나타났다. 정확도를 포함한 여러 수치로 나이브 베이지언 모델보다 높은 성능임을 알 수 있다.

4.4.3. 기존 연구와 비교

이번 실험은 관련 연구에서 보인 자연어 처리를 통한 규칙 기반의 감성 분류 연구와 본 논문의 시스템을 비교한다. 비교할 규칙 기반의 감성 분류 연구는 김정호 등(2010) 연구의 실험 데이터와 성능 결과이다. 비교 연구의 실험 데이터는 쇼핑 사이트에서 추출

한 MP3P 상품평이며 데이터 구성은 아래와 같다.

Table 13. MP3P Reviews from Comparative Study

	Positive	Negative
Training	3000	3000
Test	260	260

실험을 위해 비교 연구의 데이터를 요청하였고 아래와 같이 데이터를 받아 실험에 사용하였다.

Table 14. MP3P Reviews to be used

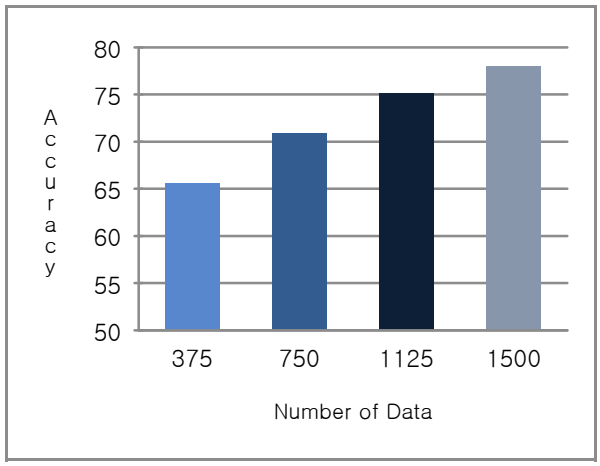
	Positive	Negative
Training	1500	1500
Test	209	209

위 데이터로 본 논문의 시스템을 이용해 실험을 하였고 실험 결과는 아래와 같다.

Table 15. Result of Experiments on MP3P Reviews

False Negative		16
False Positive		75
True Negative		133
True Positive		189
ambiguous		0
sum		413
precision	**positive**	0.715909
	negative	0.892617
recall	**positive**	0.921951
	negative	0.639423
f-measure	**positive**	0.80597
	negative	0.745098
accuracy		0.779661

테스트 데이터 긍정, 부정 각 209개씩 418개 중 내용이 없는 상품평이 제외되어 413개가 실험되었고 위와 같은 결과를 얻었다. 정확도를 비교하여 보면, 비교 연구의 실험 정확도는 82.83%를 보이고 본 논문의 시스템에 데이터를 적용하였을 때의 정확도는 약 77.97%이다. 비교 연구에 비해 4.86% 낮은 정확도를 보인다. 이는 비교 연구 실험 데이터에 비해 트레이닝 데이터 개수가 50%이어서 사전에 등록되는 단어의 개수와 통계치가 부족함이 원인이다. 트레이닝 데이터 개수에 따른 정확도 향상은 실험을 통해 아래 결과를 얻었다.



김정호 등(2010)의 연구를 포함한 기존의 연구들과 달리 트레이닝 데이터 개수가 증가하면서 성능이 향상됨을 알 수 있다.

다음은 영화평 데이터에 김정호 등(2010)에서 사용한 형태소 분석을 실시한 후 이를 본 논문의 시스템에 입력으로 주었을 때의 실험이다. 영화평 데이터 949,960개 데이터에 형태소 분석을 실시하였고 크로스 확인 방법을 시행하여 데이터를 20개로 나누어 테스트에 1개, 트레이닝에 19개를 사용하였다.

Table 16. Result of Experiments on Morpheme Analyzed Movie Reviews

False Negative		2066.4
False Positive		3375.2
True Negative		4374.4
True Positive		27870.2
ambiguous		4504
sum		37686.2
precision	**positive**	0.891984
	negative	0.679202
recall	**positive**	0.930977
	negative	0.564514
f-measure	**positive**	0.911061
	negative	0.616545
accuracy		0.764265

Table 17. Result of Experiments on Not Morpheme Analyzed Movie Reviews

False Negative		1740.2
False Positive		2346.2
True Negative		5441.2
True Positive		29019.2
ambiguous		23.6
sum		38546.8
precision	**positive**	0.925199
	negative	0.757679
recall	**positive**	0.943431
	negative	0.698725
f-measure	**positive**	0.934224
	negative	0.726986
accuracy		0.893443

실험 결과, 형태소 분석을 실시한 데이터 실험에서 정확도가 약 76.43%를 보이고, 형태소 분석을 하지 않은 데이터의 실험 결과는 89.34%의 정확도를 보였다. 형태소 분석한 데이터의 정확도가 12.91% 낮은 결과이다. 이는 탈문법, 탈철자, 이모티콘 등의 사용이 잦은 영화평 데이터에 형태소 분석을 하는 일이 의미 단위를 추출하는 일이 어렵기 때문이다. 실험 결과에서도 형태소 분석을 실시할 때에 모호한 수치가 높은데, 이는 처리하지 못하는 표현들로 인해 단어를 잃게 되어 판단할 수 없게 된 결과이다. 처리 시간에서도 기존연구와 본 논문의 시스템은 차이를 보인다. 949,960개의 영화평 중 테스트 데이터인 약 47,000여개를 판단할 때, 기존 연구는 형태소 분석이 반드시 실시되어야 한다. 이 때 형태소 분석에 소요되는 시간은 김정호 등(2010)의 형태소 분석기로 47,000개를 판단하기 위해서 형태소 분석에 약 4,500초를 소비한다. 이에 비해 본 논문의 시스템은 47,000개 데이터의 감성 판단에 본 논문의 실험 환경으로 평균 56초를 소비한다. 본 논문에서 제안하는 시스템은 형태소 분석을 거치지 않기 때문에 매우 높은 처리율을 보인다.

5. 결론 및 향후 연구

감성 분류는 인터넷 사용자들이 생성하는 데이터에서 감성 요소를 추출하고 이를 분류하는 일로 기업이나 제품의 이미지, 사람들의 성향 등을 알 수 있는 근거가 되기 때문에 의미 있는 연구로 볼 수 있다. 이런 이유로 현재까지도 감성 분류 연구는 활발히 진행되고 있다.

현재까지 감성 분류의 주된 연구 주제는 형태소 분석을 포함한 자연어 처리를 통해 데이터 글을 정제하고 이를 바탕으로 규칙 기반의 사전을 구축하는 형태이다. 규칙 기반의 감성 분류 연구들은 자연어 처리 결과에 따라 성능이 좌우되는 특성을 가지고 있는데 현재까지 형태소 분석을 포함하는 자연어 처리 연구의 성능은 기대만큼 나오지 않았다. 이에 본 논문에서는 사용자들이 생성하는 감성 표현들을 형태소 분석과 같은 자연어 처리를 거치지 않고 그대로 사용하는 방법을 제안하였다. 이 정제되지 않은 표현들은 사전에 그대로 등록되며 감성 판단은 사전에 등록된 표현들의 빈도수 값의 조합으로 결정한다. 이에 따라 사전에 등록되어 있는 표현이 다양하고 빈도수가 증가할

수록 점점 더 좋은 결과를 가져올 것이라고 가정하였다. 이 가정은 4장 실험을 통해 참임을 입증하였다.

본 논문은 현재까지 이루어진 주류의 감성 분류 연구들과 다른 방향의 접근을 보였다. 이 접근으로 기존 문제점들을 해결할 수 있었지만 접근법의 특성으로 인해 몇 가지 보완해야 할 점을 지닌다.

1) 통계치 노이즈(Noise)

2) 사전 데이터 관리

제안한 시스템에서 사용하는 가중치를 적용한 빈도수 통계치와 이를 적용한 수식은 데이터가 점차 늘어가면서 사용자가 부여한 감성 정도와 표현한 단어를 전혀 다르게 작성한 데이터로 인한 노이즈가 쌓일 수 있다. 본 논문의 감성 판단 수식에는 이러한 노이즈 처리 부분이 없어 판단 결과에 영향을 줄 가능성이 있다. 이를 위해 데이터와 판단 결과에 대한 심도 있는 분석이 필요하다.

사전에는 사용자가 생성한 모든 감성 표현들이 등록되기 때문에 데이터의 양이 매우 많다. 앞으로 데이터가 점차 쌓일수록 사전의 크기 역시 커지리라 예상할 수 있다. 현재는 사전 데이터를 메모리에서 처리하고 있지만 데이터가 커지면 구글의 빅테이블(BigTable)에 대응하는 HBase와 같은 데이터베이스를 이용할 필요가 있으리라 생각한다. HBase는 하둡과 함께 동작하는 데이터베이스로 비교적 빠른 검색과 업데이트 속도를 보인다. 앞으로 HBase의 안정화와 성능 향상으로 사전 데이터 관리 데이터베이스로서의 적합성은 더해질 것이다(Chang 등, 2008).

위의 보완 연구들로 본 논문에서 제안한 '맵리듀스를 이용한 통계적 접근의 감성 분류'는 훌륭한 감성 분류 시스템이 되리라 기대한다.

REFERENCES

Bracewell, D.B. (2010). Semi-Automatic WordNet Based Emotion Dictionary Construction, *Machine Learning and Applications (ICMLA), 2010 Ninth International Conference.*

Christopher D. Manning., Prabhakar Raghavan., & Hinrich Schütze. (2007). Introduction to Information Retrieval, *Cambridge University Press, Text classification and Naive Bayes.*

Fay Chang, Jeffrey Dean., Sanjay Ghemawat., Wilson C. Hsieh., Deborah A. Wallach., Mike Burrows., Tushar Chandra., Andrew Fikes., & Robert E. Gruber. (2008). Bigtable: A Distributed Storage System for Structured Data, *ACM Transactions on Computer Systems, Vol 26, Issue 2.*

Hong, J. P. & Cha, J. W. (2008) A New Korean Morphological Analyzer using the Eojeol Pattern Dictionary(어절패턴 사전을 이용한 새로운 한국어 형태소 분석기), *Korea Computer Congress Vol.35, No.1.*

Jeffrey Dean. & Sanjay Ghemawat. (2008) MapReduce: simplified data processing on large clusters, *Communications of the ACM - 50th anniversary issue: 1958 - 2008, Volume 51 Issue 1, January.*

Jimmy Lin. & Chris Dyer. (2010) Data-Intensive Text Processing with MapReduce, *University of Maryland, College Park.*

Kang, S. S. (2003) Korean Morpheme Analysis and Information Retrieval(한국어 형태소 분석과 정보검색), *HONGRUNG PUBLISHING*, 106p.

Kim, M. K. (2010) A polarity classification system on internet emotional texts(인터넷 감성 텍스트에 대한 극성 분류 시스템), *Graduate School of Korea Aerospace University.*

Kim, J. H., Kim, M. G., Cha, M. H., In, J. H., & Cha, S. H. (2010) Sentiment Classification considering Korean Features(한국어 특성을 고려한 감성 분류), *Korean Society for Emotion & Sensibility,* 449-458

Myung, J. S., Lee, D. J., & Lee, S. G. (2008) A Korean Product Review Analysis System Using a Semi-Automatically Constructed Semantic Dictionary (반자동으로 구축된 의미 사전을 이용한 한국어 상품평 분석 시스템), *Korean Institute of Information Scientists and Engineers, 35(6).*

Shvachko, K., Hairong Kuang, Radia, S., & Chansler, R. (2010) The Hadoop Distributed File System, *Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium*, pp 1-10.

Sim, J. H., Kim, J. S., Cha, J. W., & Lee, G. B. (2000) Robust Part-of-Speech Tagger using Statistical and Rule-based Approach(통계와 규칙을 이용한 강인한 품사 태거), *POSTECH.*

Tom White. (2010) Hadoop: The Definitive Guide,

Second Edition, *O'REILLY*.

원고접수: 2012.01.04
수정접수: 2012.09.08
게재확정: 2012.11.15