맵리듀스 기반의 자동 토픽 레이블링 알고리즘

박상민(*), 온병원(**)
(*) 군산대학교 소프트웨어융합공학과, bk1162@kunsan.ac.kr
(**) 군산대학교 소프트웨어융합공학과, bwon@kunsan.ac.kr

MapReduce-based Automatic Topic Labeling Algorithm

Park, Sang-Min(*), On, Byung-Won(**)+
(, **) Kunsan National University, Department of Software Convergence Engineering*

요 약

텍스트 데이터 분석 기법 중 하나인 토픽 모델링 알고리즘은 대량의 텍스트 문서에서 토픽을 찾아 문서 군집을 수행한다. 토픽 모델링을 통해 추출된 토픽은 주요 연관어들로 구성되어있고 이러한 연관어들을 통해 토픽의 의미를 명명하는 것을 토픽 레이블링(Topic Labeling)이라 한다. 기존에는 사람이 직접 연관어들을 통해 의미를 판단하여 토픽 레이블링을 하기 때문에 시간이 오래 걸리고 사람마다 다른 의미로 해석할 수 있는 문제점이 존재한다. 본 연구에서는 이와 같은 문제를 해결하기 위해 빠르고 정확한 토픽 레이블링을 자동으로 수행하는 방안을 제안한다.

1. 서론

텍스트 데이터 분석 기법 중 하나인 토픽 모델링 알고리즘은 대량의 텍스트 문서에서 토픽을 추출하기 위해 사용된다. 토픽 모델링에 의해 추출된 토픽은 주요 연관어들로 구성되어 있고 이러한 연관어들의 의미를 판단하여 토픽에 대해 명명해 주어야 한다. 이것을 토픽 레이블링이라 하며 이와 관련된 다양한 연구가 진행되어 왔다. [1]은 토픽 내 연관어들의 벡터 값을 통해 중앙값을 구하고 중앙값에서 가장 가까운 벡터에 해당하는 단어를 토픽의 레이블로 사용하는 방안을 제안하였고, [2]는 토픽 모델 방법을 통해 추출된 토픽을 어구와 요약문을 통해 자동 레이블링 하여 쉽게 의미를 알 수 있는 방안을 제안하였다.

본 연구에서는 문장과 토픽 간의 관련성의 정도를 나타내는 스코어(Score)를 통해 Top-k개의 문장을 토픽의 레이블로 사용하는 방안을 제안한다. 하지만 제안 방안은 대량의 텍스트 문서에서 스코어를 계산할 경우 상당한 시간이 소요된다는 문제가 존재한다. 이와 같은 문제를 해결하고 빠른 토픽 레이블링을 수행하기 위해 필터링 기법과 블록킹 기법을 제안 한다.

2. 제안방안

제안 알고리즘은 (1) 필터링 (2) 블록킹 (3) 스코어 계산 등으로 구성된다. 실험에 있어서 12,971개의 문장을 사용하였고 이를 입력으로 받아, 통계적인 토픽 추출 기법인 Latent Dirichlet Allocation (LDA)을 통해 주요 토픽을 추출한다. 각 토픽마다 필터링 기법을 통해 후보 문장을 추출한 후 블록킹 기법을 통해 스코어를 계산하여 Top-k개의 문장을 토픽의 레이블로 사용한다.

2.1 문제 정의

본 연구에서는 스코어를 통한 자동 토픽 레이블링 알고리즘을 제안한다. 하지만 본 제안 방안은 100개의 토픽, 500개의 연관어 그리고 100,000,000개의 문장이 존재한다고 가정을 했을 때 5,000,000,000,000번의 스코어 계산을 수행하기 때문에 상당한 시간이 소요된다. 이러한 제안 방안의 문제점을 해결하기위해 본 연구에서는 필터링 기법과 블록킹 기법을 제안한다.

2.2 필터링 기법 (Filtering Method)

(그림 1)은 필터링 기법을 설명한다. 필터링은 전체 문장에서 토픽과 관련성이 높은 문장들을 후보 문장으로 채택하여 제공해주는 제안 방안이다. 이를 통해 불필요한 연산을 최소화 할 수 있다.

2.3 블록킹 기법 (Blocking Method)

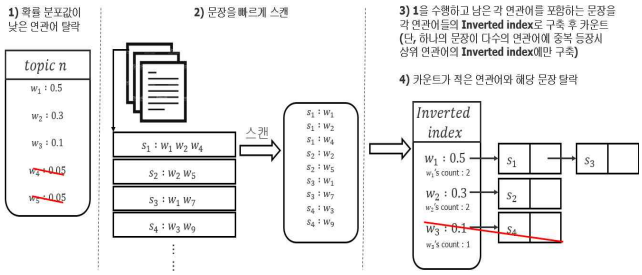
(그림 2)는 블록킹 기법을 설명한다. 블록킹은 필터링에 의해 구축된 역색인(Inverted Index)의 각 레코드를 입력으로 하고 Mapper들을 수행한 후 MapReduce 기반의 병렬 처리

+ 교신저자 : 온병원

1) 이 논문은 2016년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No.NRF-2016R1A2B1014843)
2) 이 논문은 2017년도 정보(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No.NRF-2017M3C4A7068188)

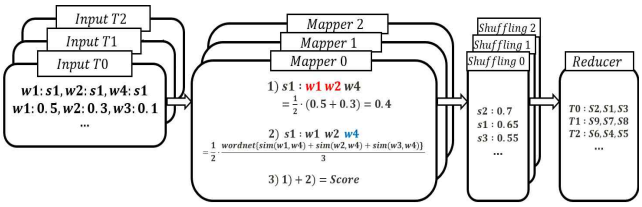
를 통해 스코어 계산을 한다. 이러한 결과들은 Reducer를 통해 취합되고 최종 결과를 수행한다.

(그림 1) 필터링 기법



(Figure 1) Filtering method

(그림 2) 블록킹 기법



(Figure 2) Blocking method

2.3 스코어 계산

스코어는 연관어의 확률 분포 값과 워드넷 유사도를 통해 측정되며 0~1사이의 값을 갖는다. 값이 1에 가까울수록 토픽과 관련성이 높으며 아래의 수식을 따른다. T는 토픽의 집합이며 $t \in T$이다. S는 문장의 집합이며 $s \in S$이고 $w' \in s$ 이다. α는 확률 분포 값에 대한 가중치이다.

$$\text{Score} = \alpha \cdot \frac{\sum_{w' \in tokens(s) \wedge w'=t} P(t)}{|tokens(s)|} + (1-\alpha) \cdot \frac{\sum_{w' \in tokens(s) \wedge w' \neq t} wordnet(w', T)}{|number\ of\ w' \neq t|}$$

3. 실험 환경 및 결과

3.1 실험 환경

보배드림 사용자 후기 게시판(www.bobaedream.co.kr)으로부터 총 12,971개의 K5 자동차 관련 문장을 수집하였으며 이러한 문서들로부터 토픽 추출을 위해 JGibbLDA[3]를 사용하였다. <표 1>은 실험 환경에 대해 나타낸다.

<표 1> 실험 환경

실험 환경	
CPU	Intel® Core i7-4790 CPU@3.60GHZ
Memory	8GB
HDD	900GB
OS	Ubuntu 14.04
Hadoop version	2.7.1
Cluster system	2(namenode:1, datanode:1)

<Table 1> Experimental setup

3.2 실험 결과

<표 2>는 제안방안에 의해 추출된 토픽 0의 Top-3 레이블이다. 해당 레이블들은 K5 자동차의 성능 중에서도 연비, 주행, 브레이크 등을 나타내는 것을 알 수 있으며, 우리가 제안한 레이블을 통해 토픽의 자세한 의미를 파악할 수 있다.

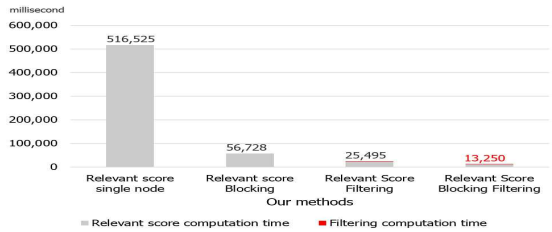
<표 2> 토픽 0의 Top-3 레이블

Rank	Topic 0's label	Score
1	연비는 12.8km/ℓ를 달성.. 여러 준대형 차종들에 비해 한 수 우위의 경제성을 실현 ..	0.791259
2	고성능에 걸맞은 탄탄한 하체.. 다이나믹한 주행을 위해 많이 사용되고있는 시프트 패들은 장착되지 않음	0.763751
3	브레이크 성능 국산차들과 비교해 크게 떨어지지 않음	0.731619

<Table 2> Top-3 labels of topic 0

(그림 3)은 제안방안에 대한 실행 속도를 비교한 결과이다. 스코어를 필터링 기법과 블록킹 기법을 적용하지 않고 단일 노드에서 계산했을 때 516,525ms가 소요되었지만 필터링 기법과 블록킹 기법을 모두 적용하여 스코어를 계산했을 때는 13,250ms이 소요되었다. 이는 기존의 제안 방안보다 39배 빠른 실행 속도를 보여주었으며 이를 통해 필터링 기법과 블록킹 기법의 효용성을 입증할 수 있다.

(그림 3) 제안 방안 실행 시간



(Figure 3) Execution time of our methods

4. 결론 및 향후연구

본 연구에서는 스코어를 이용하여 정확한 토픽 레이블링을 자동으로 수행하였다. 또한 빠른 토픽 레이블링을 위해 필터링 기법과 블록킹 기법을 제안하고 우수성을 입증하였다.

향후 연구로는 1,200만개의 영문 뉴스 기사를 수집하여 대용량 데이터로 실험을 진행할 것이며, 추출된 레이블인 Top-k 문장들을 요약해서 보여주는 opinion summarization 방안을 연구할 계획이다. 또한 관련연구와 비교실험을 통해 제안방안의 우수성을 입증할 것이다.

참고 문헌

[1] Kim, J., Kim, H. and S. Lee, "Monthly issue extracting using topic labeling", Korea Computer Congress (KSC' 16)

[2] Wan, X. and T. Wang, "Automatic labeling of topic models using text summaries", Proceeding of Conference of the Association for Computational Linguistics (ACL' 16)

[3] JGibbLDA - A Java implementation of Latent Dirichlet Allocation (LDA), http://jgibblda.sourceforge.net/ (Accessed in 2016)