토픽모델의 성능 향상을 위한 불용어 자동 생성 기법

이정빈*, 인호*
*고려대학교 컴퓨터학과
e-mail:{jungbini, hoh_in}@korea.ac.kr

Automatic Generating Stopword Methods for Improving Topic Model

Jung-Been Lee*, Hoh Peter In*
*Dept of Computer Science, Korea Universityty

요 약

정보검색(Information retrieval) 및 텍스트 분석을 위해 수집하는 비정형 데이터 즉, 자연어를 전처리하는 과정 중 하나인 불용어(Stopword) 제거는 모델의 품질을 높일 수 있는 쉽고, 효과적인 방법 중에 하나이다. 특히 다양한 텍스트 문서에 잠재된 주제를 추출하는 기법인 토픽모델링의 경우, 너무 오래되거나, 수집된 문서의 도메인이나 성격과 무관한 불용어의 제거로 인해, 해당 토픽 모델에서 학습되어 생성된 주제 관련 단어들의 일관성이 떨어지게 된다. 따라서 분석가가 분류된 주제를 올바르게 해석하는데 있어 많은 어려움이 따르게 된다. 본 논문에서는 이러한 문제점을 해결하기 위해 일반적으로 사용되는 표준 불용어 대신 관련 도메인 문서로부터 추출되는 점별 상호정보량(PMI: Pointwise Mutual Information)을 이용하여 불용어를 자동으로 생성해주는 기법을 제안한다. 생성된 불용어와 표준 불용어를 통해 토픽 모델의 품질을 혼잡도(Perplexity)로써 측정한 결과, 본 논문에서 제안한 기법으로 생성한 30개의 불용어가 421개의 표준 불용어보다 더 높은 모델 성능을 보였다.

1. 서론

정보 검색과 텍스트 분석을 위해서 수집되는 비정형 데이터인 자연어는 어휘나 문법적으로 표현의 형태가 매우 다양하고, 복잡하기 때문에 문장을 자르는 토큰화(Tokenization), 형태소 분석이나 불용어 제거와 같이 다양한 텍스트 마이닝 기법을 이용해 정형 데이터로 정제한다. 이 중, 불용어 제거 과정은 텍스트 분석에 있어 가치가 없거나, 불필요한 어휘들을 제거함으로써 분석 모델의 품질을 향상시킨다. 일반적으로 관사(a, an, the), 전치사(of, in for, through), 대명사(it, their) 등과 같이 자주 사용되는 어휘나 문장에서 큰 역할을 하지 않는 단어들이 불용어로써 제거된다.

특히 다양한 텍스트 문서에 잠재된(Latent) 주제를 추출하는 텍스트 분석 기법인 토픽모델링[1]을 위한 전처리 과정으로써, 일반적으로 널리 사용되는 표준 불용어 리스트(Fox stopword[2])를 이용하여 불용어를 제거한다. 그러나 이러한 표준 불용어 리스트는 다음과 같은 문제점을 가지고 있다. 첫째, 비교적 오래전 문서들로부터 추출된 단어들의 빈도수를 기반으로 작성되었기 때문에 오늘날 수집되는 문서들에서 사용되는 단어들과는 많은 차이가 있다. 둘째, 일반적인 문서들로부터 추출된 단어이기 때문에 특정 주제나 도메인에서 수집된 문서들의 불용어로 사용되기는 적합하지 않다[3]. 이러한 문제점들로 인해 토픽 모델에서 생성된 토픽 관련 단어들의 일관성이 떨어지게고, 이로 인해 분석가들이 토픽 모델의 결과를 해석함에 있어 큰 어려움이 따른다.

따라서 본 논문에서는 이러한 표준 불용어 리스트의 문제점들을 해결하기 위해 점별 상호정보량(PMI: Pointwise Mutual Information)과 토픽 모델링 결과를 활용한 불용어 자동 생성 기법을 제안한다. 본 기법은 표준 불용어와 달리 비교적 최근에 사용되고, 모델링 대상과 관련된 도메인의 문서에서 계산된 점별 상호정보량을 활용하여 반복적인 토픽 모델링 과정에서 주제와 관련성이 떨어지는 불용어를 자동 생성해 준다. 실험 결과, 대표적인 표준 불용어 리스트인 Fox stopword에서 제공하는 단어 개수(421개)의 약 7%밖에 안 되는 30개의 자동 생성된 단어만으로도 8.6% 더 낮은 혼잡도를 보였다. 이 결과를 통해 본 논문에서 제안하는 기법이 기존 표준 불용어 리스트보다 더 높은 토픽 모델의 성능을 보였음을 확인하였다.

본 논문의 구성은 다음과 같다. 2장에서는 제안하는 기법을 적용하기 위해 사용된 데이터 셋에 대해 소개하며, 3장에서는 토픽 모델링 및 PMI를 설명하고 이를 이용한 불용어 자동 생성기법에 대해서 설명한다. 4장에서는 제안한 기법에 대한 실험과 토픽 모델의 성능을 비교 평가 한 후, 5장에서 향후 계획과 함께 결론을 맺는다.

2. 데이터 소개

본 논문에서 제안한 기법을 위한 대상 데이터로써 Git 저장소에 공개된 다음 12개의 오픈소스 프로젝트에 대한 커밋 메시지를 수집하여 사용하였다. 일반적인 기사나 문서가 아닌 해당 데이터셋을 선택한 이유는, 소프트웨어 공학 분야에서 소프트웨어 저장소에서 수집되는 다양한 텍스트 정보들을 분석하여 유의미한 정보를 찾는 연구들이 많이 활성화 되어 있기 때문이다. 예를 들어, 소스코드, 커밋로그, 개발자, 버그 정보 등으로부터 추출된 텍스트들이 텍스트마이닝 기법을 통해 버그의 위치를 예측하거나, 잠재 결함의 오탐을 분석[4]하는데 활용된다.

각 프로젝트는 사용자 평가(Stargazer)가 높으며, 최근까지도 활발하게 개발이 진행되어오고 있는 Java 언어 기반의 프로젝트로써, 총 4년간(2012년~2015년)의 커밋 메시지를 추출하였으며, 특정 주제와는 상관없이 수집되었다. 또한, 커밋 메시지에 포함된 소스코드 및 주석과 그 코멘트 모두를 분석 데이터로 활용하였다. 수집된 커밋 메시지는 비교 평가를 위해 불용어 제거를 제외하고, 모두 동일하게 자연어 전처리를 수행하였다.

표 1. 분석 대상 프로젝트의 로그 메시지 데이터 셋

프로젝트	커밋개수	크기	단어 종류
Actor	4,871	185k	17,127
Alluxio	8,251	380k	
Casssandra	7,073	565k	
CoreNLP	12,561	478k	
Druid	4,912	256k	
Gradle	24,396	1,560k	
Graylog	8,354	464k	
Hadoop	19,877	2,310k	
Kotlin	6,801	317k	
Netty	8,320	1,035k	
OrientDB	1,825	89k	
PDE	6,913	252k	

PMI 계산을 위해 모든 프로젝트 커밋 메시지를 통합하여 하나의 텍스트 말뭉치(text corpus)로 만들고, 이에 대한 단어 빈도(Term Frequency)와 동시 발생 빈도(Co-Occurrence Frequency)를 구하였다. 또한, 토픽 모델의 혼잡도의 정확성을 위해서 통합된 커밋 메시지에서 2,000개씩 10개의 샘플을 임의로 추출하여 학습된 토픽 모델의 테스트 데이터로 사용하였다.

3. 불용어 자동 생성 기법

이 장에서는 토픽 모델링과 PMI에 대한 개념을 소개하고 이를 이용한 불용어 자동 생성 프로세스를 제안한다.

3.1. 토픽 모델링

토픽 모델링은 텍스트 마이닝의 기법 중 하나로써 각자의 문서마다 몇 가지의 주제를 가지고 있는데, 그 주제를 바탕으로 문서를 이루는 단어들이 생성된다는 가정을 가지고 출발한다. LDA(Latent Dirichlet allocation)[1]는 문서를 작성하는 과정에 관한 생성 모델이며, 문서들의 잠재된 주제를 분류하는데 사용된다. 이 모델을 통해 학습 데이터의 주제 분석과 동시에 새로운 문서의 주제 역시 분석할 수 있다.

3.2. 점별 상호정보량 (PMI)

점별 상호정보량은 두 확률변수의 연관성을 나타내는 정보량을 표현하는 지표이다. PMI를 이용하면 사람이 판단하는 것과 매우 흡사하게 토픽의 일관성을 측정할 수 있다[5]. PMI는 의미가 비슷한 단어들이 하나의 문서 안에 나타날 확률이 높다고 가정했을 경우, 두 단어의 연관성을 다음과 같이 측정할 수 있다. 두 단어가 나타날 확률이 서로 독립적이라면 PMI 값이 0이 될 것이고, 양의 방향으로 커질수록 두 단어가 같은 문서 안에 나타날 확률이 크다고 볼 수 있다.

$$PMI(w1, w2) = \log \frac{p(w1,w2)}{p(w1)p(w2)}$$

수식 1. 점별 상호정보량(PMI)

p(w1)과 p(w2)는 문서 안에 해당 단어 w1과 w2가 나타날 확률이라고 할 수 있으며, 이 확률은 다음과 같이 단어 빈도(tf)를 총 문서의 개수 N으로 나누어 표현할 수 있다.

$$p(w) = \frac{tf(w1)}{N}$$

수식 2. 단어(w)가 나타날 확률

p(w1, w2)는 문서 안에서 해당 단어가 동시에 나타날 확률이라고 할 수 있으며, p(w)와 마찬가지로 동시 발생 빈도를 총 문서의 개수 N으로 나누어 표현 가능하다.

본 논문에서는 2장의 커밋 메시지 데이터를 통합하여 만든 텍스트 말뭉치에서 계산된 단어 빈도 및 동시 발생 빈도를 이용하여 PMI를 계산한다.

3.3. 불용어 자동 생성 프로세스

다음은 본 논문에서 제안하는 불용어 생성을 위한 의사코드(Psudo code)를 나타낸다.

표 2. 불용어 자동 생성 의사 코드

```
// ① 입력 매개 변수 할당
X = 불용어 생성 개수
K = 토픽의 개수
DOCs = 수집된 문서들
STOPWORD = []        // 불용어를 담을 배열 변수
```

```
// ② 텍스트 말뭉치(학습 데이터) 만들기 및
// 단어 빈도 및 동시 발생 빈도 계산
trainCorpus = makeTextCorpus(DOCs)
tf, cooccur = calcTermFreq(trainCorpus)

// ③ 반복 토픽 모델링을 통해 불용어 생성
for x = 1 to X do:

  // (ㄱ) 토픽별 단어 리스트 저장
  topicWordList[][] = TopicModeling(trainCorpus, K)

  // (ㄴ) 토픽별 단어들 끼리 PMI 계산
  for k = 1 to K do:
    for t = 1 to 10 do:
      swMatrix = calculatePMI(topicWordList[k][t])
      minSW[k] = min(swMatrix)

  // (ㄷ) K개의 불용어에서 최종 불용어 선택
  finalStopword = calculatePMI(minSW)

  // (ㄹ) 가장 작은 PMI 값의 단어를 불용어로 추가
  STOPWORD.append(finalStopword)
```

① 의사코드를 실행하기 위한 입력 매개변수로는 생성할 불용어의 개수(X), 분류할 토픽의 개수(K) 및 텍스트 문서가 있다.

② 입력 된 텍스트 문서들은 토픽 모델링을 위한 학습데이터로 통합 및 정제되며 이 과정에서 단어 빈도와 동시 발생 빈도를 구한다.

③ 생성할 불용어 개수만큼 토픽 모델링을 반복한다.

(ㄱ) 분류된 K개의 토픽을 각각 대표하는 최상위 단어 10개를 리스트에 저장한다. 그 후, 하나의 토픽에 저장된 단어 리스트들 끼리 PMI 값을 비교하여 10x10 상관 행렬에 저장한다.

(ㄴ) 상관 행렬이기 때문에 각 행 또는 열의 합을 구하여 가장 작은 값을 가지는 단어를 저장한다.

(ㄷ) 각 토픽별로 구해진 불용어 후보가 다른 토픽에서는 높은 상관성을 가질 수 있으므로, 다른 토픽의 단어들과도 PMI를 통해 상관성을 구한다.

(ㄹ) 최종적으로 가장 낮은 PMI를 갖는 단어 1개를 선택하여 불용어 리스트에 추가한다.

4. 실험

본 장에서는 앞서 제안한 불용어 자동 생성 기법을 바탕으로 토픽 모델의 성능을 측정하는 실험을 수행한다. 토픽 모델의 성능을 측정하는 지표 중에 하나인 '혼잡도(Perplexity)'는 학습된 토픽 모델이 실제 관찰 가능한 결과를 생성해 낼 확률을 측정할 수 있다. 즉, 혼잡도가 낮을수록 높은 성능을 가진 모델이 생성되었음을 나타낸다.

4.1 실험 설계

3.3장에서 제안한 프로세스에 불용어의 개수 X는 30개, 토픽의 개수 K는 10개로 일정하게 제한하였다. 토픽 모델링을 위해 LDA를 구현한 자바 기반의 MALLET[1] 도구을 활용하였으며, 본 논문에서 제안하는 방법으로 생성되는 stopword와 비교하기 위해 Fox stoplist[2]를 도구의 stoplist로 사용하였다. 또한, 토픽 모델링을 위한 파라미터 값들은 도구에서 제공하는 디폴트 값을 사용하였다. 기타 자연어 처리 및 PMI 구현은 파이썬으로 구현하였다. 마지막으로, PMI 계산을 위해 필요한 단어 빈도와 동시 발생 빈도 정보를 텍스트 파일로 저장하여 검색하지 않고, MongoDB에 저장한 후 인덱싱을 생성하여 검색 속도를 높였다.

통계적 검정을 수행하기 위해 3.3의 프로세스를 30번 반복하여 평균을 내었다. 따라서 10개의 임의의 샘플에 대한 30개의 불용어를 생성하는 과정을 30번 반복하여 총 900번의 실험을 진행하였고, 그 평균 혼잡도를 구했다.

4.2. 결과 분석 및 평가

총 30번의 반복된 프로세스에서 대부분 거의 비슷한 불용어를 생성하였으며, 표 3은 임의의 프로세스에서 생성된 불용어 리스트 30개이다. PMI의 특성으로 인해 독립적으로는 발생 빈도가 높지만, 토픽 모델에서 생성된 주제 관련 단어들과는 동시에 발생하는 빈도가 낮은 단어들이 불용어로 생성된다.

표 3. 불용어 자동 생성을 통해 리스트에 추가된 단어

jbelli, model, review, repo, cassandra, channel, this, spec, motiv, block, plugin, patch, yarn, contribut, result, read, byte, feat, changes, tachyon, maven, now, output, system, unit, apach, https, releas, that, rule

정성적인 분석 결과, jbelli, cassandra, tachyon, apach와 같은 단어들은 특정 프로젝트의 커밋 메시지에서 빈도가 높을 뿐, 다른 프로젝트에서는 거의 등장하지 않았기 때문에 불용어로 생성되었다고 볼 수 있다. 또한, this, now, that 과 같이 Fox stoplist에 포함된 단어들도 등장하였으나, 90% 이상의 단어들이 해당 도메인과 관련된 불용어로 포함되어 표준 불용어 리스트가 본 데이터 셋에는 크게 적합하지 않음을 알 수 있었다.

그림 1은 Fox stopword를 적용했을 때와 제안한 자동 생성 불용어를 적용했을 때 토픽 모델의 평균 혼잡도를 비교한 결과 그래프이다. 각각 10개의 샘플 테스트 데이터에서 구한 혼잡도 30개를 평균한 값을 비교하였다.

1) http://mallet.cs.umass.edu/

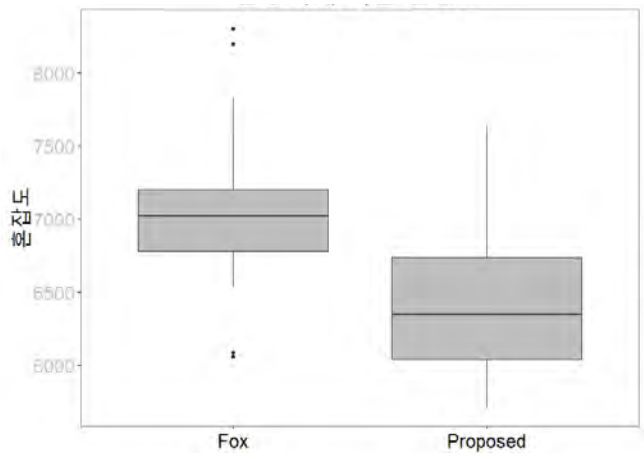

그림 1. 적용 불용어에 따른 평균 혼잡도

그림1에서 보는 것과 같이 Fox stopword를 적용했을 때보다 제안한 자동 생성 불용어 리스트를 적용했을 때, 8.6% 더 낮은 혼잡도를 보였으며, t 검정 수행 결과 역시 p값이 0.05 미만(p-value = 6.75e-06)으로 유의한 차이를 보였다. 이는 토픽모델의 성능이 전반적으로 높아졌음을 나타낸다. 또한, Fox stopword가 포함하고 있는 단어의 개수 421개의 약 7% 수준인 30개의 단어만으로도 토픽 모델의 혼잡도를 유의미하게 낮췄다고 해석할 수 있다.

다음은 자동 생성된 불용어의 개수에 따른 평균 혼잡도의 평균 변화 값이다. 1개의 불용어로 시작하여 30개가 생성될 때까지 평균 혼잡도가 점점 떨어지는 추세를 보이고 있다. 이러한 추세에 따르면, 불용어 1개당 평균적으로 토픽모델의 혼잡도를 약 20씩 낮출 수 있다고 볼 수 있다.

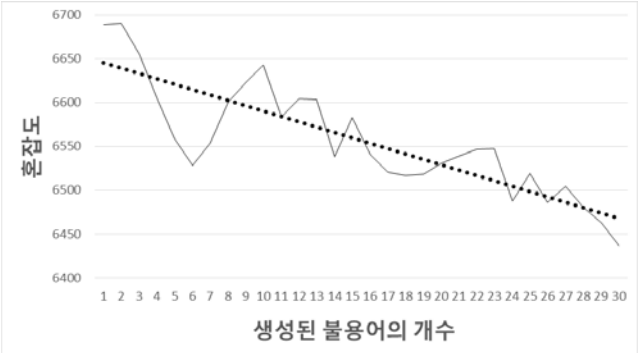

그림 2. 생성된 불용의 개수에 따른 혼잡도 변화

추가적으로 더 많은 불용어의 생성 개수와 혼잡도의 관계를 확인하기 위해, 30개 이상의 불용어를 생성하는 실험 수행하였으며 그 결과는 표 4와 같다.

표 4. 30개 이상의 불용어 생성에 따른 혼잡도 변화

불용어 개수	30개	50개	100개	200개	400개	Fox
혼잡도	6516	6229	5893	6044	6806	7224

표 4에서 보는 것과 같이 불용어 개수에 따른 혼잡도가 100개를 기점으로 다시 상승하는 것을 확인할 수 있었다. 따라서 불용어의 개수가 혼잡도와 지속적으로 반비례하지는 않기 때문에 적절한 불용어의 선택이 필요할 것으로 보인다.

5. 결론 및 향후 연구

다양한 텍스트 문서에 잠재된 주제를 추출하는 텍스트 분석 기법인 토픽모델링의 전처리 과정인 불용어 제거는 데이터 모델의 크기를 줄여 분석 속도를 높이고 모델의 품질을 향상시킬 수 있는 효과적이고 간단한 기법 중에 하나이다. 그러나 도메인의 특성이 반영되지 않고 오래된 표준 불용어 리스트는, 토픽 모델에서 생성된 토픽 관련 단어들의 일관성을 떨어뜨려 전체적으로 모델을 해석하기 어렵게 만든다.

따라서, 본 논문에서는 도메인과 관련 된 최근 문서로부터 추출한 점별 상호정보량과 토픽 모델링 결과를 활용한 불용어 자동 생성 기법을 제안하였다. 표준 불용어 리스트와 비표 평가 실험을 수행 한 결과, 표준 불용어 리스트인 Fox stopword 약 7% 수준인 30개의 단어만으로도 모델의 혼잡도를 8.6% 낮추어 더 높은 토픽 모델의 성능을 보임을 검증하였다.

향후에는 더 다양한 종류와 도메인의 데이터 셋에 적용하고, 혼잡도 이외에 모델을 평가할 수 있는 추가적인 지표를 활용하여 본 기법의 효과성을 검증하고자 한다. 현재는 알고리즘을 개선을 통해 생성 속도 및 불용어 개수를 최적화 하는 연구를 진행하고 있다.

사사

이 논문은 2016년도 정부(미래창조과학부)의 재원으로 한국연구재단-차세대정보·컴퓨팅기술개발사업의 지원을 받아 수행된 연구임(2012M3C4A7033345)

참고문헌

[1] D.M. Blei, A.Y.Ng, and M.I. Jordan. “Latent Dirichlet allocation” JMLR, 3:993-1022, 2003.

[2] Fox, C. “A stop list for general text.” ACM-SIGIR Forum, 24, 19-35. 1990.

[3] Baradad, Vicenç Parisi, and Alexis-Michel Mugabushaka. "Corpus Specific Stop Words to Improve the Textual Analysis in Scientometrics." ISSI. 2015.

[4] 이정빈, 이택, 인호, “소스파일 주제 기반 잠재결함 오탐 분석 기법”, 한국소프트웨어공학 학술대회 논문집 (KCSE 2015), 17권 1호, p.190-191, 2015.

[5] D.Newman, J.H. Llau, K. Grieser, and T. Baldwin, “Automatic evaluation of topic coherence.”, In NAACLHLT, 2010.