http://bit.ly/310KMQo

감성 분석에 머신 러닝 적용

박종민

전처리

1. Normalize
2. Tokenize
3. Stem / Lemmatize
4. Remove Stop-words

Normalize

- 영어의 정규화는 보통 Upper case 또는 Lower case로 통일 시키는 것
- 한글은 다른 의미로 사용하기도 함
  - (예) Avengers를 한글로 표기 → 어벤져스 | 어벤저스

We have been studying a machine learning at Kakao office

↓

we have been studying a machine learning at kakao office

Tokenize

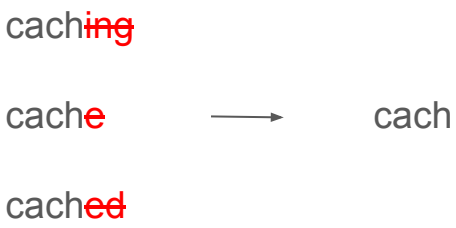
- 문장을 띄어쓰기 간격으로 분리하는 것 (BoW)

we have been studying a machine learning at kakao office

[we, have, been, studying, a, machine, learning, at, kakao, office]

Stem

- 어근 추출 규칙에 따라 어근을 추출
  - Porter, Snowball, ...



Lemmatize

- 단어의 원형이 되도록 변경
  - 명사:

    ones → one

  - 동사:

    is

    was → be

    been

Stem / Lemmatize

- Stem
  - Memory 적게 소요
- Lemmatize
  - Dictionary 필요

[we, have, been, studying, a, machine, learning, at, kakao, office]

[we, have, be, study, machine, learn, kakao, office]

Remove Stop-words

- Stop-words (불용어) 제거
  - 영어: is, the, a, …
  - 한글: 은, 는, 이, 가, ...

[we, have, be, study, machine, learn, kakao, office]

[we, study, machine, learn, kakao, office]

Term Frequency

- 문서 내 특정 단어의 빈도
  - $0 \leq tf$
- 별도의 정규화 방법
  - Boolean: 문서에 단어가 나타나면 1, 없으면 0
  - Logarithmically Scaled: log(tf)
  - Augmented: 단어의 빈도를 문서 내 단어들의 단어 빈도 중 최대 값으로 나눠주는 방법

Inverse Document Frequency

- df
  - 한 단어가 전체 문서 집합 내에서 얼마나 공통적으로 많이 등장하는지를 나타내는 값
  - $0 \leq df \leq 1$
- idf
  - 드문 용어의 idf는 높고, 흔한 용어의 idf는 낮음
  - df의 역수를 취하고 값이 발산하는 것을 방지하기 위해 log를 취함
  - $0 \leq idf$

tf-idf

- 적은 수의 문서에 용어 t가 많이 있으면 가장 높은 값을 가짐
  - 이 문서들에 대한 높은 식별력
- 한 문서나 많은 문서에 그 용어가 적게 있으면 더 적은 값을 가짐
  - 적합성이 뚜렷하지 않음
- 사실상 모든 문헌 안에 그 용어들이 있을 경우 가장 낮은 값을 가짐

HashingVectorizer

- 문제점
  - CounterVectorizer, TfidfVectorizer는 메모리 안에서 처리, 대용량 처리 불가능
  - 문서가 업데이트 될 때마다 훈련해야 함
- 대안
  - 단어에 hash function을 적용해서 **미리 정해진 길이의 vector**를 만들 수 있다.
  - hash value를 index로 사용하여 vector 값을 업데이트

Latent Dirichlet Allocation

- 여러 문서에 걸쳐 자주 등장하는 단어의 그룹을 찾는 확률적 생성 모델
  - 빈도수 기반의 BoW, tf-idf를 입력으로 사용, 단어의 순서를 신경쓰지 않음
  - 검색 엔진, 고객 민원 시스템 등과 같이 문서의 주제를 알아내는 일이 중요할 때 사용
- LDA 동작 개요
  - 문서들은 topic의 혼합으로 구성되고, topic들은 확률 분포에 기반하여 단어를 생성한다고 가정
  - 데이터가 주어지면 문서가 생성되는 과정을 역추적
  - 사용자는 topic의 개수를 지정해줘야 함 (topic 개수가 hyperparameter)

Reference

- NLP Nanodegree, Udacity
- Chapter 6.2, Introduction to Information Retrieval, Christopher D. Manning
- [IR] tf-idf 에 대해 알아봅시다, Blog
- HashingVectorize, scikit-learn
- Topic Modeling, 딥 러닝을 이용한 자연어 처리 입문