Document classification

Sang Yup Lee

ML-based approach

- Overall process
  - Text data collection
    - Web scraping 등
    - Preprocessing
      - 불용어가 제거된 특정 품사의 단어들 (i.e., features) 만 선택=> 이러한 단어들은 문서의 특성을 잘 나타내어야 함
    - Representation (vectorization)
      - 전처리 과정의 결과물을 이용
      - Bag of words model
      - TF-IDF
    - Applying a ML algorithms for training data
      - Applying the results of the learning to new data

각 문서를 벡터로 변환

- Example: three documents
  - Doc 1: ‘banana apple apple orange’
  - Doc 2: ‘apple carrot eggplant carrot’
  - Doc 3: ‘banana mango orange orange’
  - all words: ‘apple’, ‘banana’, ‘carrot’, ‘eggplant’, ‘mango’, ‘orange’
- Vectorization
  - 각 문서를 문서에 사용된 단어들로 구성이 된 vector로 표현 가능
  - 같은 corpus에 포함이된 문서들의 vector 크기는 같다 (즉, 전체 단어의 수)
  - Vector의 각 element의 값을 무엇으로 할 것이냐?
    - Frequency (사용빈도): (전처리가 끝난) Corpus에 존재하는 각 단어가 각 문서에서 몇번 사용되었는지에 대한 정보 사용
    - TF-IDF

각 문서를 벡터로 변환

- Example: frequency
  - 각 문서를 단어들의 출현빈도 정보를 가지고 표현
  - 순서
    - 전체 데이터에서 사용된 단어들을 알파벳 순으로 배열
    - 각 단어들이 각 문서에서 사용된 횟수 측정

	apple	banana	carrot	eggplant	mango	orange
Doc 1	2	1	0	0	0	1
Doc 2	1	0	2	1	0	0
Doc 3	0	1	0	0	1	2

이러한 행렬을
DTM, document-term matrix
라고 함

- 단점: 각 단어가 해당 문서에서 갖는 상대적 중요성을 표현하지 못한다.

Vectorization

각 문서를 벡터로 변환

- Example: frequency (cont'd)
  - DTM의 각 행이 각 문서의 벡터
    - Doc1 = (2,1,0,0,0,1)
    - Doc2 = (1,0,2,1,0,0)
    - Doc3 = (0,1,0,0,1,2)
- Exercise
  - 문서들 간의 유사도를 계산해 보세요.
- Frequency 기반 방법의 단점
  - 각 단어가 해당 문서에서 갖는 상대적 중요성을 표현하지 못한다.

TF-IDF

- TF*IDF (inverse document frequency)
  - 단어의 상대적 중요성
  - 예) 아래 두개의 단어 중 어떠한 단어가 문서 2의 특성을 더 잘 반영하는가?
    - 이를 어떻게 수치로 표현할 수 있는가?

	word1	word2
Doc 1	10	0
Doc 2	10	10

TF-IDF

- TF-IDF
  - 특정 단어가 특정 문서의 uniqueness를 얼마나 나타내는가를 계산하기 위해 사용
  - TF-IDF가 높을수록 해당 단어는 다른 문서에서는 적게 사용되고, 해당 문서에서 많이 사용되고 있다는 뜻으로, 해당 단어가 해당 문서의 uniqueness를 많이 나타낸다고 볼 수 있음
- IDF (inverse document frequency)
  - https://en.wikipedia.org/wiki/Tf%E2%80%93idf#Inverse_document_frequency
  - 다른 문서에서 얼마나 사용되지 않았는지를 의미
  - 1/DF 라고 생각할 수 있음

TF-IDF

- IDF 계산
  - 1/DF (실제는 조금 다름)
  - DF의 의미
    - corpus에 존재하는 전체 문서들 중에서 **해당 문서를 제외한 나머지 문서들 중**에서 해당 단어가 몇 개의 문서에서 사용되었는지를 의미
    - 예) 문서 A에서의 단어 1에 대한 DF
      - 데이터셋에 존재하는 전체 문서의 수 = 10
      - 그 중에서 문서 A를 제외한 4개의 문서에서 사용
        - then, 문서 A에서의 단어 1이 갖는 DF = 4
        - 따라서 IDF = ¼

TF-IDF

TF

	word1	word2
Doc 1	10	0
Doc 2	10	10

DF (해당 단어가 사용된 다른 문서의수)

	word1	word2
Doc 1	1	1
Doc 2	1	0

IDF (1/(DF+1)로 계산)

	word1	word2
Doc 1	1/2	1/2
Doc 2	1/2	1

TF-IDF

	word1	word2
Doc 1	10*1/2=5	0*1/2=0
Doc 2	10*1/2=5	10*1=10

Vector

Vectorization

- sklearn을 이용한 문서 vectorization
  - TF: CountVectorizer 클래스 사용
  - TF-IDF: TfidfVectorizer 클래스 사용

$$\mathrm{idf}(t) = \log \frac{1+n}{1+\mathrm{df}(t)} + 1$$

- Examples
  - For toy data, see 'doc_vectorization_example.ipynb'

CountVectorizer 사용하기

- CountVectorizer class
  - 객체 생성
    - vectorizer = CountVectorizer()

Sentiment analysis

Vectorization

Sentiment analysis (감성분석)

- What is it?
  - 글에 담긴 특정 주제에 대한 논조 (or sentiment) (긍/부정성)를 파악하는 것
  - 영화평의 예
    - 영화평1: "너무 재밌어서 또 보고 싶어요"
    - 영화평2: "돈도 아깝고 시간도 아깝습니다. 영화보고 여자친구랑 싸웠어요"
- Two different approaches
  - Machine learning (기계학습)
    - Supervised learning
  - Lexicon based (감성어 사전 기반)

ML-based approach

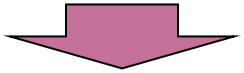
- 기계학습 기반의 감성 분석
  - Supervised learning
    - 정답이 있는 데이터 필요
      - Data with labels
      - e.g., 영화평 with 긍/부정 labels
      - 학습데이터와 평가데이터로 구분
    - 풀고자 하는 문제에 대한 데이터
      - Data with no labels
      - e.g., 영화평 without such labels

ML-based

- 영화평 분석의 예
  - 학습 데이터

힌트정보: 어떠한 단어들이 얼마만큼 사용되었는가?

영화평	Label
This movie is so fun.	Positive
This movie is disappointing.	Negative



알고리즘

  - 새로운 데이터

영화평	Label
This movie was boring.	?
The main actor is so attractive.	?

Labeling data

- 추가 정보
  - 문서 마다의 label (or class) 정보
  - 예)
    - Doc 1 -> 긍정
    - Doc 2 -> 부정
    - Doc 3 -> 긍정
  - ML algorithm이 하는 일
    - Using the words and labels information -> 어떤 단어들이 나왔을 때 문서가 긍정 혹은 부정일 확률이 높은지를 계산, 이를 label이 없는 문서에 적용해서 label을 추정

ML-based approach

- Overall process
  - Text data collection
    - Web scraping
    - 정답 데이터와 문제 데이터
  - Preprocessing
    - 특정 품사의 단어들 (i.e., features) 만 선택
  - Representation (vectorization)
    - TF
    - TF-IDF
  - Applying a ML algorithms for training data
    - Applying the results of the learning to new data

ML algorithms

- Algorithms used for classification
  - 전통적인 기계학습 알고리즘
    - **Logistic regression**
    - Support vector machine
    - Decision tree
      - Random forest
      - XGBoost
      - Light GBM
    - Naive Bayes
  - 신경망 기반 (딥러닝 알고리즘)

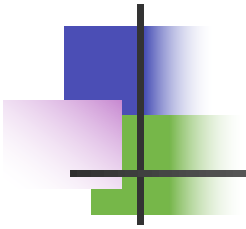
영화평 데이터 감성분석 해보기

영화평 감성분석

- Example data
  - 2016년도에 상영된 영화 중 상위 300 개 영화에 대한 네이버 리뷰 데이터
  - Training & test data
    - 이중 일부를 training에 사용하고 나머지를 모델의 결과를 evaluate하는데 사용
    - You can also have a validation set

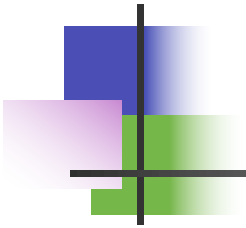
Example 1 (cont'd)

- Overall process
  - 1) Collect review data
    - Both training and test data
    - 예) 네이버 평점 데이터
      - http://movie.naver.com/movie/bi/mi/basic.nhn?code=155716
      - 네트워크 검사 방법 사용
  - 2) Text preprocessing
    - 이를 통해 특정 품사의 단어들만 저장
  - 3) DTM 로 표현 (Frequency or TFIDF)
  - 4) ML 알고리즘 적용
    - Logistic regression 사용
    - 학습 -> Evaluation



Example 1 (cont'd)

- Example codes
  - see 'LR_sentiment.ipynb'



Q & A