텍스트 마이닝

TEXT MINING

텍스트 마이닝

- Data mining Data
  - 수치형, 범주형, bool형, 텍스트 등
- 비정형데이터
  - text, image, sound etc.
  - 어떻게 정형화 데이터로 변환시킬 것인가?
- 텍스트 마이닝
  - 비정형화된 데이터를 처리
  - 블로그, 온라인 포럼, 리뷰사이트, 뉴스기사, 트위터 등
- 활용사례
  - 뉴스를 주제별로 정리
  - 이메일 스팸 분류

텍스트 마이닝

- 텍스트 마이닝
  - 1980년대 급부상하였으나 노동집약적이고 수동적인 방법으로 인해 발전이 더디다가 최근 90년대 말부터 급속하게 발전.(인터넷과 컴퓨터기술의 발달)
  - 텍스트로부터 고품질의 정보를 도출하는 과정으로, 입력된 텍스트를 구조화해 그 데이터에서 패턴을 도출한 후 결과를 평가/해석하는 일련의 과정
  - 텍스트 마이닝은 다양한 포맷의 문서로부터 단어의 매트릭스를 만들어 추가 분석이나 데이터 마이닝 기법을 적용해 통찰(insight)을 얻어 의사결정을 지원하는 방법
  - 감성분석(sentiment analysis), 워드 클라우드(word cloud), 문서의 요약(summarization), 분류(classification), 군집(clustering), 특성 추출(feature extraction) 등에 활용

텍스트 마이닝

- 용어
  - 말뭉치(corpus)
    - 대용량의 정형화된 텍스트 집합(large and structured set of texts)
  - 토큰(token)
    - 분석을 위해 사용하고자 하는 단어와 같은 의미 있는 텍스트 단위
    - 형태소 -> 단어 -> 문장 -> 문단 -> 문서
  - 토큰화(tokenization), 파싱(parsing)
    - 텍스트를 토큰으로 분할하는 과정
  - 형태소 분석(stemming)
    - 단어의 어간을 추출하는 과정
    - learned, learning, learns → learn
  - 불용어(stop words)
    - 의미가 없는 단어들
    - and, the, of, a 등

텍스트 마이닝

- 문서용어행렬(document-term-matrix, DTM)
  - 문서별 특정 문자의 빈도표
  - 이것은 각 문서가 1개 행을 이루고 각 용어가 1개 열을 채우는 구조로 된 문서들의 모음집(즉, 말뭉치)이라고 설명할 수 있는 희소 행렬(sparse matrix)이다. 이 행렬을 구성하는 각 값은 일반적으로 단어 개수 또는 tf-idf 이다.

	it	is	good	Bad
It is good.	1	1	1	0
It is bad.	1	1	0	1

- 용어문서행렬(term-document-matirx, TDM)
  - 단어별 문서의 빈도표 생성
  - DTM의 전치행렬, 문서개수<단어목록

텍스트 마이닝

- tf(term frequency) : 용어빈도
  - 문서에서 단어가 얼마나 자주 나오는가를 나타내는 척도
- idf(inverse document frequency) : 역문서빈도
  - 흔히 사용하는 단어의 중요성은 낮추면서도 문서 모음집에서 많이 사용되지 않는 단어에 대해서는 오히려 중요성을 키운 용어 사용 척도
- tf-idf(용어빈도-역문서빈도) : 두 수량을 서로 곱한 것
  - 빈도는 해당 용어가 얼마나 드물게 사용되는가에 따라 조정된다.

- 단어와 문서의 빈도 분석
  - tf-idf(용어빈도-역문서빈도)
    - 소설집에 포함된 소설이나 웹 사이트 모음집에 들어 있는 1개 웹 사이트와 같은 문서 모음집(즉, 말뭉치)에 속한 1개 문서에 대한 특정 단어의 중요도를 측정하기 위한 통계량.

텍스트 마이닝

- 기존 텍스트 마이닝에서 텍스트를 저장하는 방식
  - 문자열(string)
    - 텍스트를 R 언어의 문자열(문자 벡터 또는 리스트) 형식으로도 저장할 수 있으며, 종종 텍스트 데이터는 이 형식에 맞춰 메모리로 읽혀진다.
  - 말뭉치(corpus)
    - 이러한 객체 유형에는 일반적으로 추가 메타데이터(metadata) 및 세부 사항(details)을 사용해 주석 처리한 원시 문자열(raw strings)도 들어 있다.
  - 단어주머니(bag-of-words, BOW)
    - 단어들의 배치 순서를 고려해서 얻는 이득보다 불편함이 더 크기 때문에 어순을 무시한 텍스트 데이터를 정형화된 데이터로 간주(단어주머니 접근법)
    - 어근을 중시하는 입장에서는 어근을 통합하여 처리한 텍스트 데이터를 정형화된 데이터로 간주
    - 단어들은 물론 단어들의 순서가 포함된 텍스트 데이터가 정형화된 데이터로 간주.

- R에서 text 분석
  - 텍스트 데이터 분석 R 함수들과 결과물들은 리스트(list) 형식을 따른다.
  - lapply() : list apply
  - sapply() : simplified apply
  - tapply() : table apply

- 텍스트 데이터 or DTM을 분석하는 방법
  - 사전기반 접근방법(dictionary-based approach)
    - 사전(事前,a priori)에 규정된 단어의 의미를 기반으로 텍스트의 의미를 추정.
  - 지도 기계학습(supervised machine learning)
    - 텍스트 데이터 중 일부에서 텍스트의 의미가 알려져 있는 경우
  - 비지도 기계학습(unsupervised machine learning)
    - 텍스트 데이터의 의미가 전혀 알려져 있지 않고, 기계학습을 통해 텍스트 데이터의 의미를 추정하는 경우

- 텍스트 사전처리(preprocessing)
  1. 공란 처리
  2. 대소문자 통일
  3. 숫자표현 제거
  4. 문장부호 및 특수문자 제거
  5. 불용단어 제거
  6. 어근 동일화 처리

- Text mining R Packages
  - tm : 방대한 양의 텍스트를 효과적이고 효율적으로 처리하기에 적합한 패키지.
  - wordcloud : 텍스트 데이터 시각화
  - SnowballC : 불용어처리

- tm_map
  - 말뭉치 사전처리 과정. 데이터 정제, 문자열 표준화
    - 구두점과 불필요한 글자들(공백, 숫자, 특수기호, 대소문자 구분)을 제거.
    - 메시지를 개별 단어로 나눈다.
  - VCorpus : 메모리에서만 유지하는 corpus, 휘발성
  - Pcorpus : R외부의 데이터베이스나 파일로 관리되는 corpus.
  - tm_map 함수의 인자들
    - As.plainTextDocument : XML문서를 text로 변환
    - stripWhitespace : 빈칸 제거, 2회 이상 연이어 등장하는 공란이나 탭 공란 등을 1개의 공란으로 치환.
    - tolower : 대문자를 소문자로 변환
    - stopwords : 띄워쓰기, 시제 변환
    - removeNumbers : 숫자 제거
    - removeWords : 불용어 제거
    - stopwords() : 불용어 벡터 생성
    - removePunctuation : 특수문자 제거
  - Vectorsource : vector source 생성.

문자열 처리 방식

- 문자 처리 방식
  - 한글, 영어, 중국어, 일본어 등 수많은 언어가 존재.

- 영문의 경우
  - 대문자 A ~ Z : 26문자
  - 소문자 a ~ z : 26문자
  - 숫자 0 ~ 9 : 10개 (62개)
  - 특수기호 !, @, $ … (66개)
  - 62 + 66 = 128(2^7) : ASCII code (128개(0~127)의 문자와 정수가 1:1로 매칭)
  - 128+ 128(예비문자) = 256(2^8) : byte (문자 하나를 표현하는 단위, 0 ~ 256개 경우의 수)

- 한글, 한자, 중국어, 일본어 등 다양한 문자열을 표현

- Unicode (Multibyte를 이용하여 한 문자를 표현)
  - UTF-8 (Universal Transformation Format)
    - Unicode의 한 종류, 대부분 사용
    - 가변길이 1 ~ 4 byte를 사용한다. 1,112,064자까지 표현이 가능.
    - ASCII 호환 U+0000 ~ U+007F (1 ~ 127)
    - 한글 set이 존재

- 인코딩(encode) ←→ 디코딩(decode)
  - Encode : 문자열 → 0 or 1(byte)
  - Decode : 0 or 1 → 문자열 (string)