



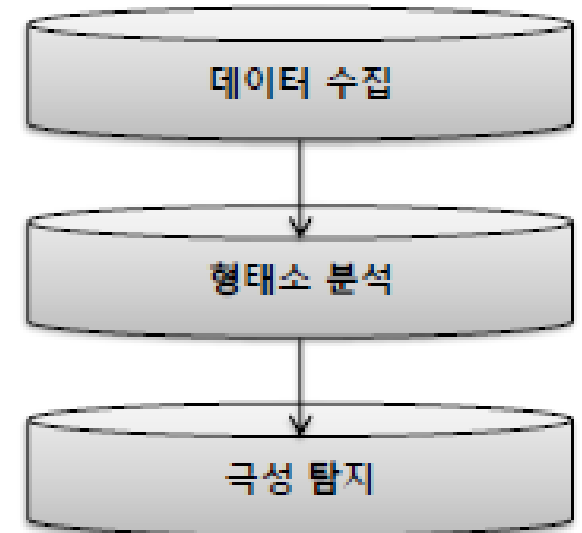
감성분석이란 ?

- 수집된 데이터를 자연어 처리와 텍스트 분석을 이용해서 텍스트 내에서 주관적인 정보를 확인하고 추출하는 기법

1. 데이터 수집(뉴스 파싱)

- 자동으로 시스템에 접속해 데이터를 화면에 나타낸 후 필요한 자료를 추출하는 방법

<감성 분석의 3단계>

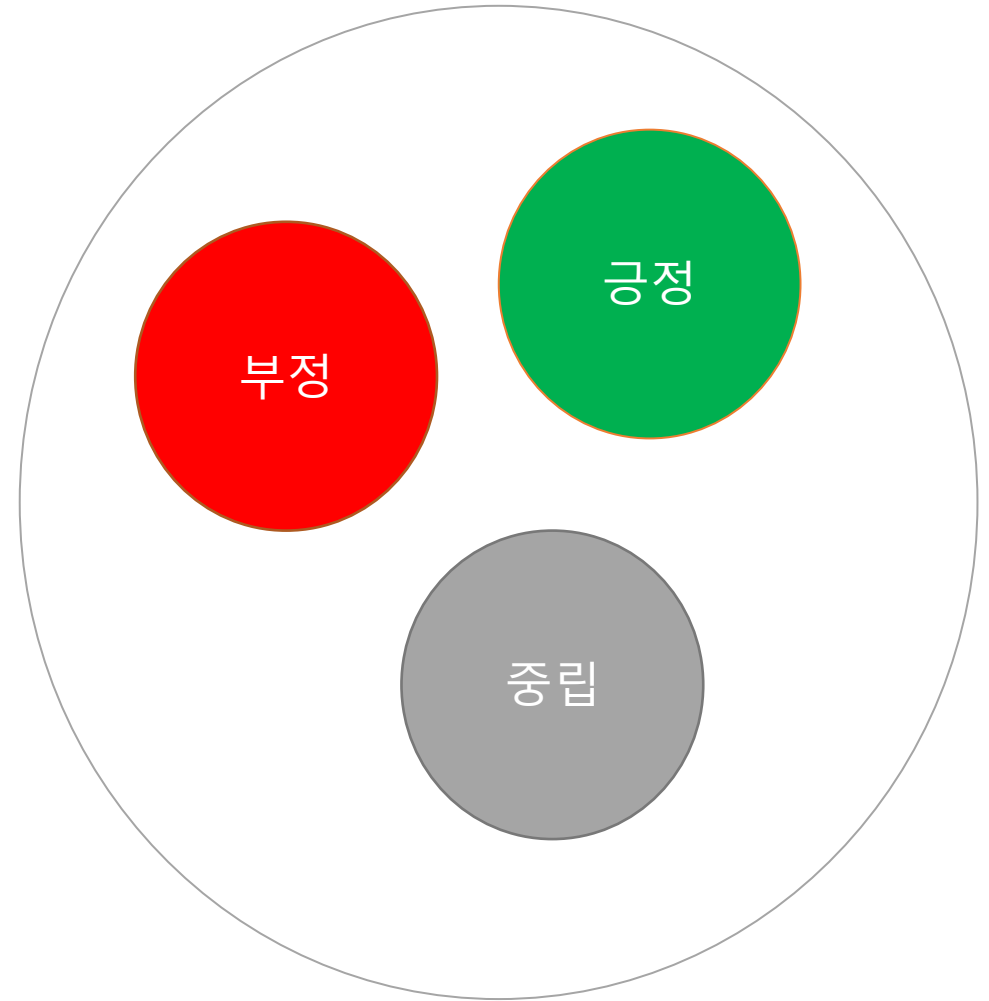


2. 형태소 분석

- 텍스트로부터 작성자의 감정이나 의견을 추출하기 위해 텍스트를 형태소 단위로 분리하여 각 형태소별 극성을 파악한 후 전체 텍스트의 극성을 분류 하는 방식
- 예) 넥슨 던전엔파이터는 9월 16일에 출시한다.
 - 출시(동사) / 한다(어미) / .(마침표)

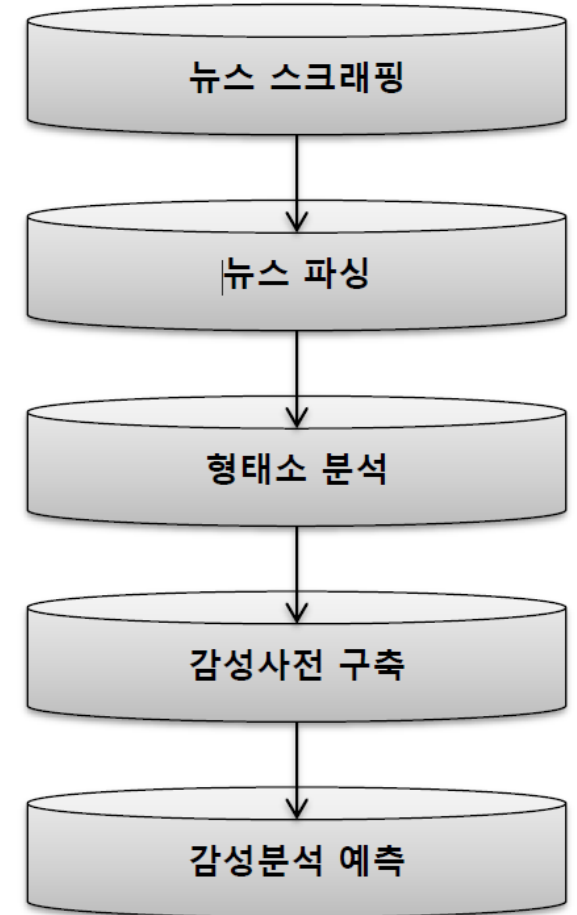
극성 탐지

- '문서' 단위의 극성 분석
- '속성' 단위의 극성 분석
- '사전' 단위의 극성 분석



극성 추출

<뉴스 데이터를 활용한 감성분석 단계>



데이터 수집(뉴스 스크래핑 & 파싱)

- 수집데이터 네이버 증권 뉴스 2019/09 ~2020/09)

형태소 분석

- 수집된 온라인 뉴스 '명사'만 활용
- 결측치 제거 (불필요한 어휘와 기호 단음절)
- 최종 추출된 명사 감성 점수화 통해 감성사전 구축

감성사전 구축

- 각 어휘 감성점수 계산 방법
- 첫번째 방법

긍정적 영향을 갖는 뉴스에서
발생한 i 의 출현 빈도

(1)

$$TermScore(i_p) = \frac{Num(i \in PosDocs)}{TotalNum(i)}$$

부정적 영향을 갖는 뉴스에서
발생한 i 의 출현 빈도

(2)

$$TermScore(i_n) = \frac{Num(i \in NegDocs)}{TotalNum(i)}$$

뉴스 전체에 나온 i 의 출현빈도

어휘 i 의 순 감성 점수

(3)

$$TermScore(i) = TermScore(i_p) - TermScore(i_n)$$

감성분석 예측

<개별 기업의 주가 예측식>

t시점의 기업 j에 대한 오피니언 점수

어휘 i의 극성 점수

$$ComScore(j_t) = \frac{\sum_{i=1}^n Num(i_t) \times TermScore(i)}{\sum_{i=1}^n Num(i_t)}$$

t 시점에 발생한 모든 뉴스에서의 어휘 i의 출현 빈도

- 뉴스 COMSCORE를 구한 뒤
- 실제 다음 거래일의 주가 등락과 일치하는지 확인
- (뉴스 기간: 전일 거래종료일 ~ 다음 거래일 시작 시간)
- 실제 다음 거래일의 주가 등락과 일치하는지 확인
- COMSCORE > 0 : 상승세
- COMSCORE < 0 : 하락세

두번째 방법) 감성사전 구축

(1) 단어의 $P(i)$ <긍정지수> 구하기

$$word(i, j) = \begin{cases} 1 & \text{\{기사 } j \text{에 단어 } i \text{가 포함된 경우}\}} \\ 0 & \text{(그 외의 경우)} \end{cases}$$

$$NSP(j) = \begin{cases} 1 & \text{\{기사 } j \text{가 게재된 후 익일}\}} \\ & \text{\{주가가 상승한 경우}\}} \\ 0 & \text{(그 외의 경우)} \end{cases}$$

↑
익일 추가

$$positive(i) = \sum_{j=1}^n \{word(i, j) \times NSP(j)\}$$

↑
Positive(i)는 긍정 값

감성사전 구축

(1) 단어의 $P(i)$ <긍정지수> 구하기

$$frequency(i) = \sum_{j=1}^n word(i, j)$$



학습된 뉴스에서 출현 횟수의 합

$$P(i) = \frac{\sum_{j=1}^n \{word(i, j) \times NSP(j)\}}{frequency(i)}$$



긍정지수는 긍정 값을 빈도수로 나눔

감성분석 예측

(1) 텍스트의 $PT(i)$ <긍정지수> 구하기

$$match(i, j) = \begin{cases} 1 & \left\{ \begin{array}{l} \text{텍스트 } i \text{에 포함된 명사 } j \text{가} \\ \text{감성사전에 존재 할 경우} \end{array} \right. \\ 0 & (\text{그 외의 경우}) \end{cases}$$

$$PT(i) = \frac{\sum_{j=1}^n \{match(i, j) \times P(j)\}}{\sum_{j=1}^n match(i, j)}$$




같은 개념으로 텍스트의 긍정지수 계산

일별 긍정 지수 구하기

$DP(i) > 0.5 \Rightarrow$ 상승

$DP(i) < 0.5 \Rightarrow$ 하락

일별 긍정 지수


$$DP(i) = \frac{\sum_{j=1}^n PT(j)}{n}$$

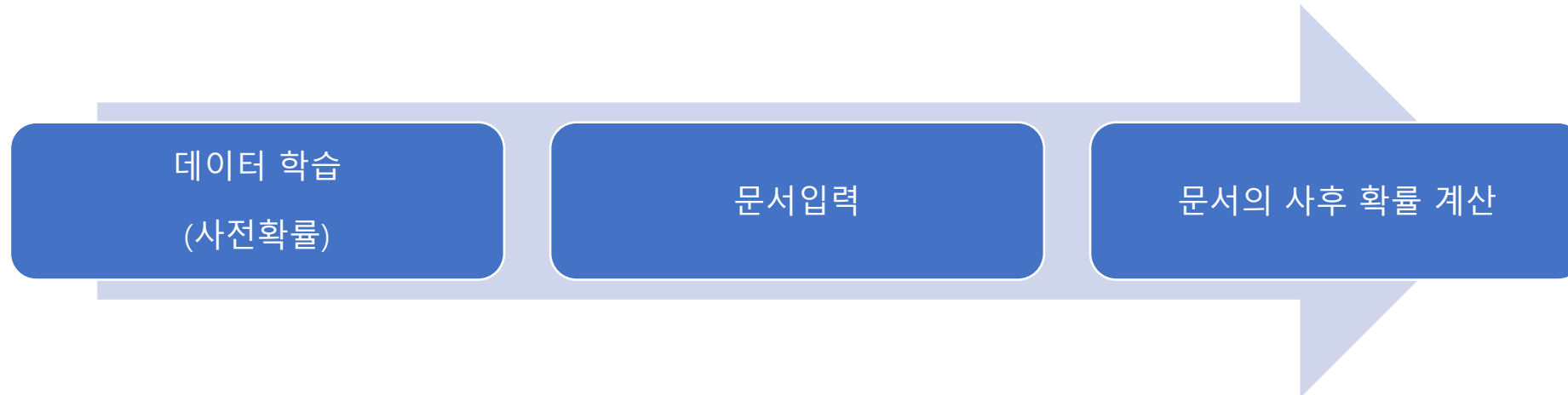
$n = \text{number of text in } i$

첫번째와 차이점

- 상승할 때 기사만을 고려해 상승 예측에 주력
- 반복되는 단어의 횟수를 모두 1로 처리

3번째 방법) Naïve Bayes classifier

- 분류를 위해서 베이즈 룰(Bayes'Rule)을 기본적으로 사용한다.
- 분류에 필요한 파라미터를 추정하기 위한 트레이닝 데이터의 양이 적어도 가능하다.
- 많은 복잡한 실제 상황에서 잘 작동한다.



Naive Bayesian classification

$$P(c|d) = \frac{P(d|c)p(c)}{P(d)}$$

d : 입력 문서

c : 분류할 부류(Class)로 긍정, 부정으로 나뉨

If $P(\text{긍정}|\text{문서}) > P(\text{부정}|\text{문서})$, 문서가 긍정부류에 속함

If $P(\text{긍정}|\text{문서}) < P(\text{부정}|\text{문서})$, 문서가 부정부류에 속함

$$P(\text{긍정}|\text{문서}) = \frac{P(\text{문서}|\text{긍정})P(\text{긍정})}{P(\text{문서})} \quad P(\text{부정}|\text{문서}) = \frac{P(\text{문서}|\text{부정})P(\text{부정})}{P(\text{문서})}$$

동일하므로 긍정, 부정
대소비교에 지장없음

$w = \text{vector of words} = (w_1, w_2, \dots, w_n)$

문서에 속한 단어들의 벡터(모음)

ex) 긍정 or 부정에 상당한 영향을 줄 수 있는 단어벡터

$$\begin{aligned} P(\text{문서}|\text{부정}) &= P(w|\text{부정}) \\ &= P(w_1, w_2, \dots, w_n|\text{부정}) \end{aligned}$$

각각의 단어들이 서로 독립이므로 분리가 가능하다
(베이즈 정리의 기본가정)

$$\begin{aligned} P(w|\text{부정}) &= P(w_1|\text{부정}) P(w_2|\text{부정}) \dots P(w_n|\text{부정}) \\ &= \prod_{i=1}^n P(w_i|\text{부정}) \end{aligned}$$

$$\therefore P(\text{부정}|\text{문서}) \propto P(\text{문서}|\text{부정})P(\text{부정})$$

$$\begin{aligned} &\propto P(w|\text{부정}) P(\text{부정}) \\ &\propto \prod_{i=1}^n P(w_i|\text{부정}) P(\text{부정}) \end{aligned}$$

$$\rightarrow P(w_1|\text{부정}) P(w_2|\text{부정}) \dots P(w_n|\text{부정})P(\text{부정})$$

어려운 점

- 감성사전 이용방식 문제점

- 여러 기사에서 단순히 많이 등장 하는(ex: 기대 고려 생각) 불용어가 높은 점수를 가질수 있다.
- 뉴스에서 단어끼리 서로 의미와 등장에 영향을 주기에 베이지안 처럼 곱하기 연산 혹은 더하기 연산으로 서로 관계를 가지지 않고 독립적인 존재로 생각함
- 주가 예측 정확도가 낮음

- 단순 베이지안 분류기 문제점

- 감성 분류에서 빈도수보다는 그 해당 단어 자체가 있고 없고가 더 중요함
- 즉 단어의 가중치를 생각하지 못함

감성사전 이용방식 문제점 해결

• TF-IDF 이용

- 1) 한 문서 내에서 등장하는 단어의 빈도를 나타내는데 단어와 문서 간의 중요도를 나타내기 위한 것.
 - 문서 내에서 (TF가 높을수록) 상대적으로 더 중요하다는 의미

$$TF_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

$n_{i,j}$ ← 문서 d_j 에서 단어 t_i 가 나오는 횟수
 $\sum_k n_{k,j}$ ← 문서 d_j 에서 나오는 모든 단어 횟수

문서 d_j 에서 단어 t_i 의 중요도.

2) IDF

- DF(Document frequency)는 문서 빈도, 자주 등장하는 단어가 몇 개의 문서에 등장하는지를 나타낸다.
- DF 높다 → 전체 문서에서 많이 등장하는 단어로, 불용어 수준이라 생각함.
- IDF = DF 역수이며 로그를 취해준다. 단어 간의 거리를 일정하게 유지하기 위해 로그를 취해 주는데 자연로그나 상용로그 중 선택하면 된다

$$IDF(t,D) = \log \left(\frac{\text{전체 문서의 갯수}}{\text{단어 } t \text{가 포함된 문서의 수}} \right)$$

TF-IDF 요약

$$TF = \frac{\text{문서 내 단어의 개수}}{\text{문서 내 모든 단어의 수}}$$

$$IDF = \log\left(\frac{\text{문서 전체 갯수}}{\text{단어를 포함한 문서의 수}}\right)$$

$$TF-IDF(t,d,D) = TF(t,d) \times IDF(t,D)$$

- > 특정 문서 내에서 단어 빈도가 높고(TF 높고), 전체 문서에서 그 단어가 포함된 문서가 적다면(IDF높으면) 이 값은 높아진다.
- > 따라서 이 값을 이용하면 불용어를 걸러 낼 수 있으며 단어별 가중치가 된다.

개선 방안

1&2. 데이터 수집(뉴스 스크래핑 & 파싱)

-> 수집데이터 : 2014년 6월 ~ 2015년6월 시가총액 상위 30위

3. 형태소 분석

-> KoNLPy(파이썬) 명사만 추출

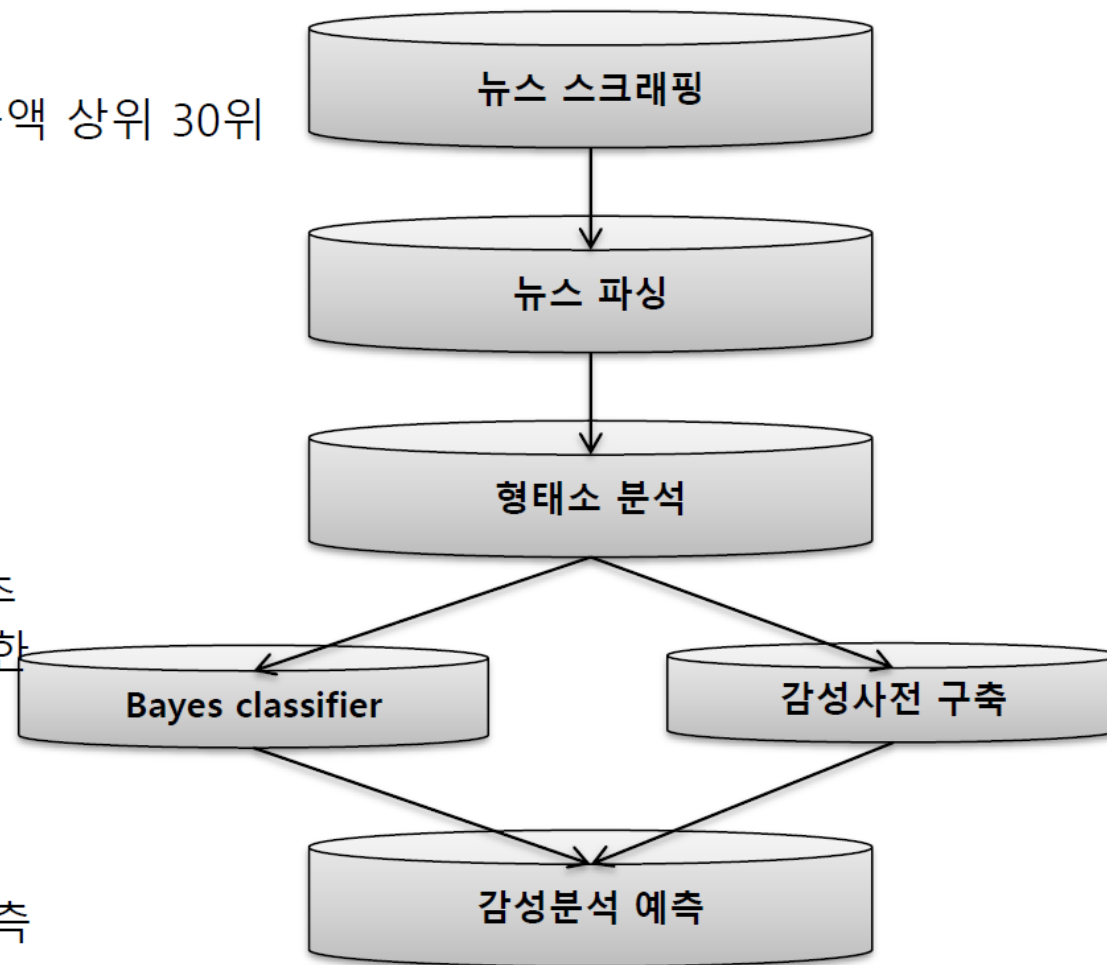
4&5. Bayes classifier & 감성사전 구축

-> Com score 조금 변형한 TF-IDF 이용

-> 문서에 해당 단어가 출현하기만 하면 1로 간주하고 그 출현 빈도에 앞선 TF-IDF 가중치 값을 곱한다.

6. 감성분석 예측

-> 문서의 긍정, 부정확률 비교하여 감성분석 예측



감사합니다