대한산업공학회
2011년 추계학술대회

문서분류 성능 향상을 위한 단어 가중치 기법에 대한 연구

김민희, 권영식

동국대학교
산업시스템공학과

목 차

- **인터넷의 사용 증가에 따라 웹 상의 전자 문서의 양 또한 폭발적으로 증가하고 있음**
- **전자 문서는 웹 상에서 접할 수 있는 모든 문서를 말함**
  - ✓ 온라인 문서
  - ✓ 인터넷 뉴스 문서
  - ✓ 전자 메일 문서
  - ✓ 의료 정보 문서
  - ✓ 디지털 도서관 문서 등

- **전자 문서가 양적으로 크게 늘어남에 따라 사람이 이러한 수많은 문서를 일일이 수작업으로 분류하는 작업은 거의 불가능 해졌음**
- **이에 따라 문서를 알맞게 분류하는 것을 도와주는 도구(tool)에 대한 요구가 점차 중요해 지고 있음**
- **따라서 문서분류시스템에 대한 연구가 활발히 이루어 지고 있음**

● **자동 문서분류시스템(Text Classification System)**

- 미리 정의된 두 개 이상의 범주(category)에 대하여, 새로운 문서 집단이 입력 되었을 때 미리 학습된 범주(category)와 입력 문서 간의 유사도 비교를 통해 입력된 문서에 대한 범주 할당을 자동으로 해주는 기법
- 사람이 어떤 글의 내용을 충분히 이해하지 않고도 그 글의 주제를 쉽게 알아내는 능력을 기계에 학습시켜 분류업무를 자동화 할 수 있다는 것에 바탕을 둠
- 자동 문서분류를 이용하여 대량의 문서를 수작업으로 분류하는데 소요되는 시간과 노력 등을 감소시킬 수 있으며, 효율적인 정보의 조직 및 검색을 가능하게 하는 이점이 있음

1.1 연구의 배경 및 필요성

- **적용 분야**
  - 스팸메일 필터링
  - 뉴스기사 필터링
  - 뉴스 기사 등의 주제선정(topic spotting)
  - 웹 문헌 분류
  - 웹 에이전트

[그림 1-1] 문서분류 적용분야 예시

- **적용 분야에 대한 문서분류 성능 향상에 관련된 연구들은 현재까지도 많이 진행되고 있음**
  - 전자메일 분류에 대한 나이브 베이지안 학습과 중심점 기반 분류의 성능 비교(2002, 김국표) – TFIDF 가중치
  - 복합 분류기를 이용한 웹 문서 범주화에 관한 실험적 연구(2003, 이혜원) – $1+\log_2 tf$ 가중치
  - 나이브 베이지안 분류자를 이용한 유해 웹문서 필터링(2007, 정태한) – CHI-square 통계량
  - 변형 나이브 베이즈 분류기를 이용한 자동 문서 분류에 관한 연구(2008, 나윤재) – TFIDF 가중치
  - 상호 정보량을 부가한 나이브 베이지안 분류기를 이용한 관련단어 추천(2010, 임태훈) – Mutual Information

● **문서 분류 기법에서 적용 가능한 통계적 이론과 기계학습 방법**

- 다중회귀모형(multivariate regression models), K-NN(nearest neighbor classifiers), 확률적 베이지안 모형(probabilistic Bayesian models), 결정 트리(decision trees), 신경망(neural networks), SVM(Support Vector Machines) 등이 있음
- 문서 분류를 위한 각각의 알고리즘들은 그 나름대로의 알고리즘의 장단 점 및 모델 형성 과정에서 비롯되는 특징들을 가지고 있음

SVM

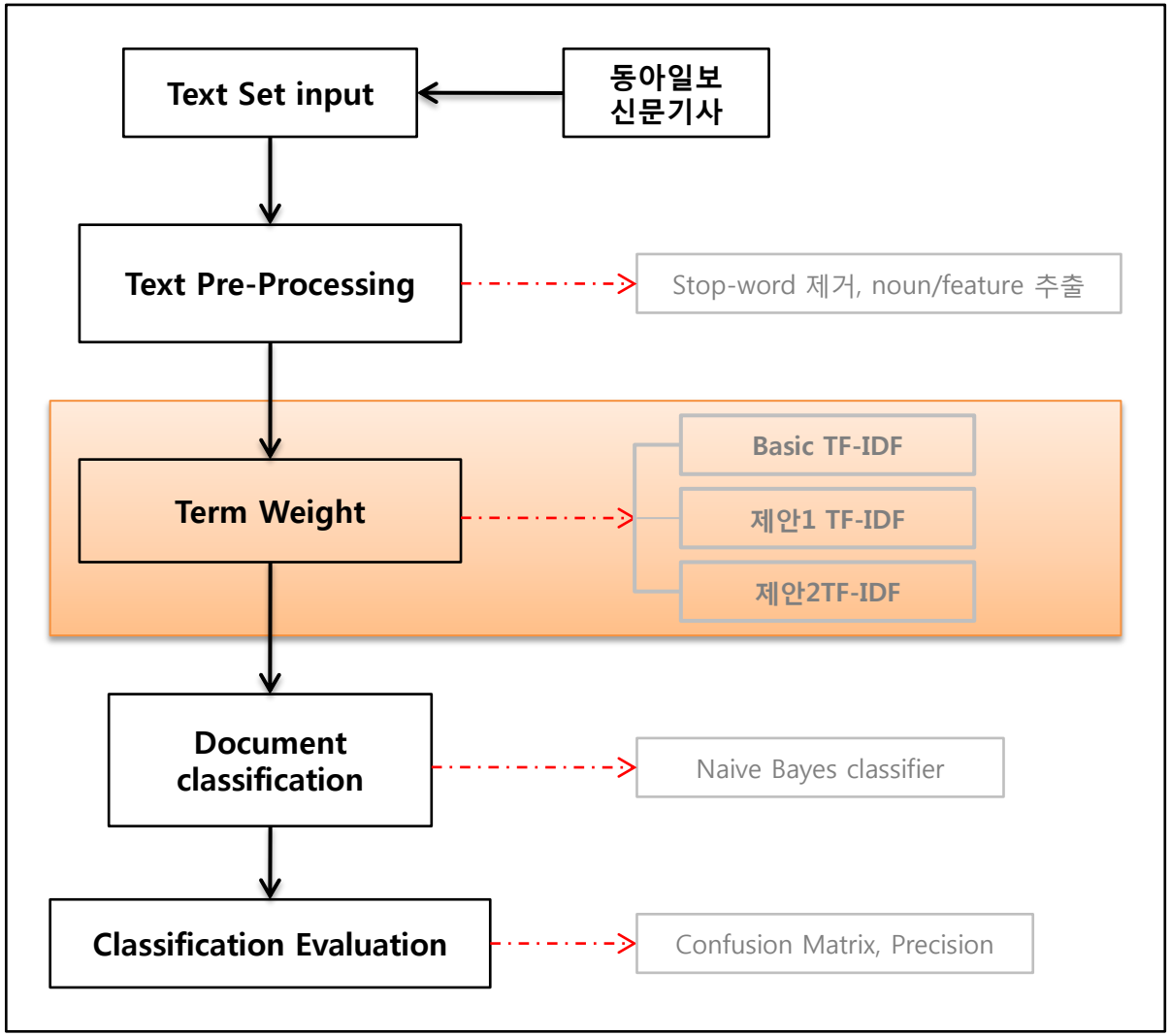
- SVM 분류기를 이용한 문서 분류 실험 연구결과를 보면 다른 학습방법을 적용한 것보다 분류 성능이 우수한 것으로 밝혀진바 있음
- 하지만, 여러 가지 옵션 설정과 파라미터들을 여러 번 설정해야 하는 번거로움 때문에 다른 알고리즘들 보다 이용하기 어려운 점이 있음

NB

- 이 시스템은 일반적으로 다른 알고리즘들에 비해 문서 분류의 정확도가 상대적으로 높다고 알려져 있음
- 또한 Naive Bayes 문서분류 알고리즘은 문서분류함수의 구축이 간단하며, 고 문서 분류의 속도가 상대적으로 빠르기 때문에 문서분류 시스템의 구축에 매우 많이 사용되고 있음

Naive Bayes 알고리즘을 사용하는 문서분류시스템의 문서 정확도를 높이는 작업은 큰 의미가 있다고 할 수 있음

본 연구에서는 나이브 베이즈 분류기를 이용하여 문서분류 실험을 하였고, 문서분류 성능 향상을 위한 단어 가중치 기법을 제안하는 것에 목적을 둔다.


Text Set input
동아일보
신문기사
Text Pre-Processing
Stop-word 제거, noun/feature 추출
Term Weight
Basic TF-IDF
제안1 TF-IDF
제안2TF-IDF
Document
classification
Naive Bayes classifier
Classification Evaluation
Confusion Matrix, Precision

[그림 3-1] 실험 구성

- **단계 1. 실험 데이터 구성**
  - 실험 데이터는 **동아일보 신문기사**를 수집하여 구축하였음
  - 동아일보 신문기사에서 **5개의 카테고리 선정**하여 실험에 사용함

- **단계 2. 데이터 전처리 수행**
  - 형태소 분석기를 이용하여 **불용어 제거 및 어간 추출**
  - 형태소 분석기는 국민대 강승식 교수가 개발한 **HAM version 2.2.0** 을 사용함
  - 형태소 분석된 데이터에서 **명사 및 특정단어 추출** 수행

- **단계 3. 단어 가중치 부여**
  - 분류기 성능 실험을 하기 위해 선정된 단어들에 단어 가중치 부여
  - 기존 가중치 기법 및 제안된 가중치 기법을 적용하여 비교 실험 수행
  - 단어 가중치 부분은 **JAVA를 이용하여 알고리즘 개발**

- **단계 4. 문서 분류기 실험**
  - 학습은 **k-fold cross validation** 방법을 사용하였고, **10-fold**로 실험하였음
  - 가장 보편적으로 많이 사용하고 그 성능이 검증된 Naïve Bayes 분류기로 실험
  - Naïve Bayes 분류기 tool은 **WEKA version 3.6.3**을 사용함

- **단계 5. 실험결과 및 평가**
  - 성능 평가는 **정확률(Precision)**, 그리고 **오분류행렬(confusion matrix)**를 통하여 비교 평가

단계 1. 실험 데이터 수집

성장둔화로 세수차질 우려

올해 성장률이 기대에 미치지 못할 경우 세수에도 차질이 예상돼 경기활성화를 위한 추경편성이 쉽지 않을 것으로 예상된다.
24일 재정경제부에 따르면 올해 세입예산은 5% 성장전망을 토대로 130조원의 국세수입을 예상했으나 지난 1.4분기 성장은 2.7%에 그쳐 하반기 경기가 살아나더라도 올해 5%성장은 사실상 불가능에 가까운 것으로 평가되고 있다. 정부는 올해 수출이 다소 둔화되더라도 내수가 살아나 수출둔화 폭을 보완할 수 있을 것으로 전망했으나 아직까지는 내수회복이 뚜렷하지 않다.
내수가 부진할 경우 지난해 실적이 반영되는 법인세 등 직접세 보다는 부가가치세, 교통세 등 간접세에 타격을 줘 세수차질이 예상된다.
이종규 세제실장은 1.4분기 세수실적 평가결과 정상적인 추세로 들어오고 있다고 밝혔으나 전문가들은 지난해 크게 결손이 난 세수가 올해 어느 정도 보완이 될 지 관심이며 여기서 한 발짝 더 나아가 추경편성 재원확보는 쉽지 않을 것이라고 밝혔다.
한덕수 부 총리 겸 재경부장관은 21일 경제정책조정회의에서 "추경편성은 세입예산을 고려해야 하기 때문에 시간을 두고 생각해 봐야 한다"고 말했다.
지금까지 들어온 세수가 추경편성을 할 만큼 여유가 없고 앞으로도 용이하지 않을 것으로 받아들이는 분위기다.
국세청이 경제에 주름살이 가지 않는 범위 내에서 세수를 늘리기 위한 다각적 대책을 마련하고 있으나 투자활성화와 사회 취약 층 지원을 위한 각종 조세감면 등으로 세수증대가 쉽지 않다.
국세청이 각종 세무조사 등으로 음성.탈루소득에 대한 검증을 강화하고 있지만 세수확보 측면에서 기여도는 크지 않다. 민간에서는 하반기 내수가 살아나지 않으면 다시 적자국채 발행으로 이어질 수 있다고 경고하고 있다.

(서울/연합뉴스)

[그림 3-1] 동아일보 신문기사 본문

- 동아일보 신문기사는 2005.05 ~ 2005.12 까지 8개월 동안의 신문기사를 수집하여 구축하였음

카테고리 이름	Corpus 1	Corpus 2	합 계
경제	60	40	100
정치	60	40	100
사회	60	40	100
국제	60	40	100
문화	60	40	100
총 합	300	200	500

[표 3-1] 실험에 사용된 데이터 정보

형태소 분석

[그림 3-2] 형태소 분석기 HAM 실행 결과

예시 문장

올해 성장률이 기대에 미치지 못할 경우 세수에도 차질이 예상 돼 경기 활성화를 위한 추경편성이 쉽지 않을 것으로 예상된다.

입력

형태소 분석

<형태소 분석기 HMA>

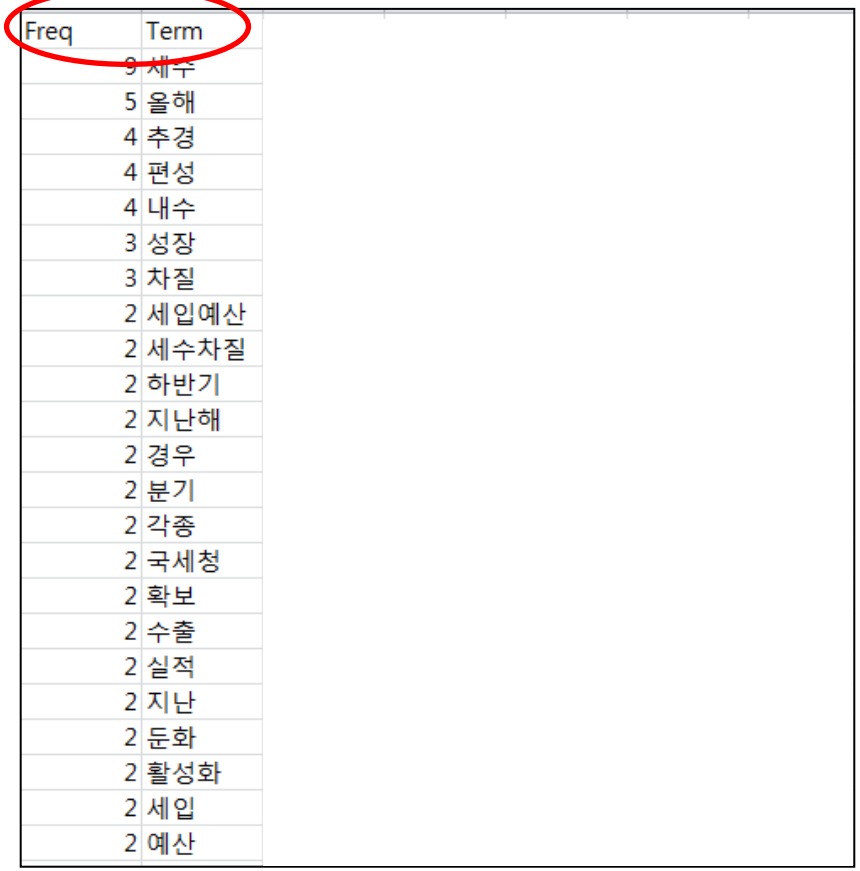
불용어 제거 | **명사 추출** | **단어 빈도수**

올해 성장률이 기대에 미치지 못할 경우 세수에도 차질이 예상돼 경기 활성화를 위한 추경편성이 쉽지 않을 것으로 예상된다.

명사 추출

올해/ 성장률/ 기대/ 경우/ 세수/ 차질/ 예상 / 경기활성화/ 추경편성/ 예상

특정단어 추출



Freq	Term
9	세수
5	올해
4	추경
4	편성
4	내수
3	성장
3	차질
2	세입예산
2	세수차질
2	하반기
2	지난해
2	경우
2	분기
2	각종
2	국세청
2	확보
2	수출
2	실적
2	지난
2	둔화
2	활성화
2	세입
2	예산

[그림 3-3] 단어 빈도수 정렬 결과

- 형태소 분석기 결과로 나온 단어를 빈도수에 대해 내림차순 정렬하여 상위 20%의 단어들을 특정단어(feature)로 추출하였음
- 빈도수(freq)가 1인 단어들은 제외하고 빈도수가 2 이상인 단어들부터 사용하였음

Freq	term
9	세수
5	올해
4	추경
4	편성
4	내수
3	성장
....	

[표 3-2] 추출된 단어 예

```
// TF-IDF 가중치 계산
    public static void genTFIDF (String dirname, ArrayList<String> columeWord, CalTFIDF idf) {
        String pattern = "#########.#####";
        DecimalFormat dformat = new DecimalFormat(pattern);

        // 문서의 수만큼 반복시행
        for (int i = 1; i <= 100; i++) {
          try {
            HashMap<String, Integer> wList = new HashMap<String, Integer>();
            ArrayList<String> lList = new ArrayList<String>();
            BufferedReader input = new BufferedReader(new FileReader("저장폴더"));
            System.out.print(dirname);
            try {
              String line = null;
              while ((line = input.readLine()) != null) {
                if (line.trim().length() > 0) {
                    lList.add(line);
                    for (StringTokenizer st = new StringTokenizer(line); st.hasMoreTokens();) {
                      String w = st.nextToken();
                      if (wList.get(w) == null)
                        wList.put(w, 1);
                      else
                        wList.put(w, wList.get(w) + 1);
                    }
                }
              }
            int index = 0;
            for (int j = 0; j < columeWord.size(); j++) {
              System.out.print("₩t");
              if (wList.get(columeWord.get(j)) != null) {
          //          System.out.print( tf*IDF 가중치 계산 공식);
                      System.out.print(보정 tf*IDF 가중치 계산 공식);
                      System.out.print(변형tf*IDF 가중치 계산 공식);
               } else {
```

[그림 3-4] 가중치 부여 알고리즘

< 적용된 가중치 공식 >

▪ **TF-IDF 가중치**

$$w_{ij} = tf_{i,j} \times idf_i$$

▪ **제안1 가중치**

$$\left[(1-w)+\frac{w \times n_{i,j}}{\sum n_{k,j}}\right] \times idf_i$$

▪ **제안2 가중치**

$$W_t = (n_{i,j} \times w + df_i \times (1-w)) \times idf_i$$

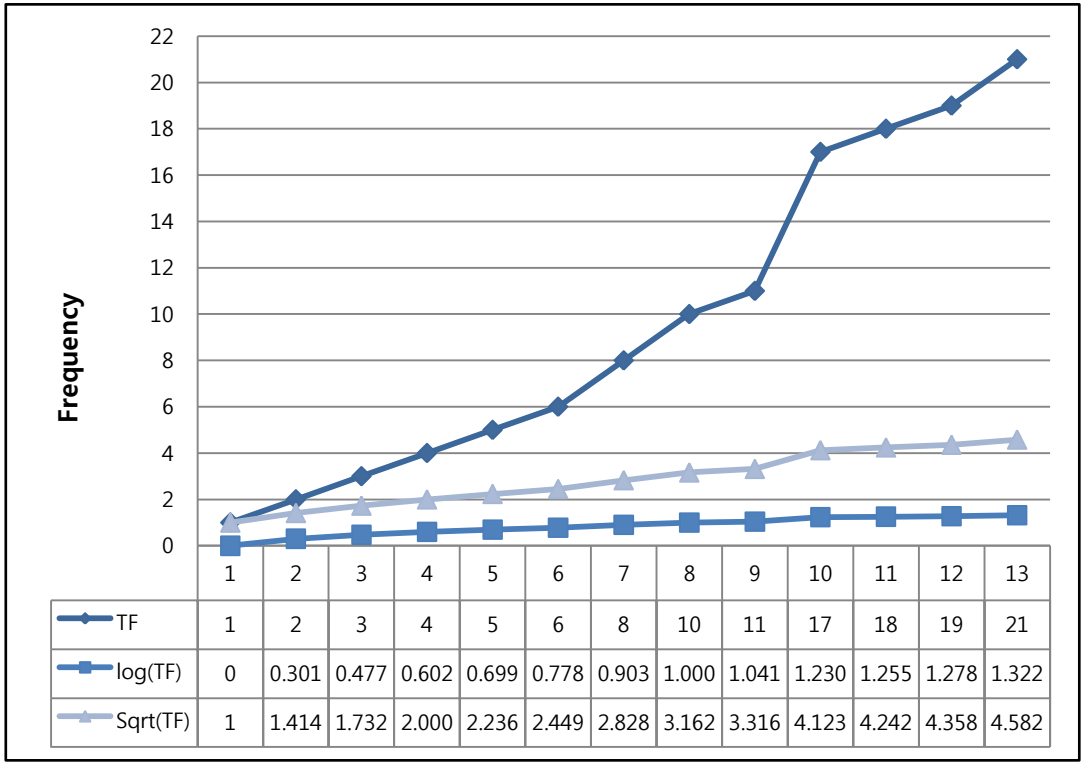
▪ **TF(term frequency)** $tf_{i,j} = \frac{n_{i,j}}{\sum n_{k,j}}$

▪ **IDF(Inverse Document Freq.)** $idf_i = \log\left(\frac{N}{n_{i,j}}\right)$

- **TF (Term Frequency Weighting)**
  - 단어 빈도 가중치 기법에는 다음과 같은 것들이 있음

이 름	공 식
단순	$TF = tf$
루트직선	$TF = \frac{tf+3}{4}$
루트	$TF = \sqrt{tf}$
로그	$TF = 1 + \log(tf)$
더블로그	$TF = 1 + \log(1 + \log(tf))$
더블로그2	$TF = 1 + \log_2(1 + \log_2(tf))$
보정	$TF = (1-w) + w \times \frac{tf}{\max\ tf}$

[표 3-3] 단어빈도 가중치 공식



	1	2	3	4	5	6	7	8	9	10	11	12	13
TF	1	2	3	4	5	6	8	10	11	17	18	19	21
log(TF)	0	0.301	0.477	0.602	0.699	0.778	0.903	1.000	1.041	1.230	1.255	1.278	1.322
Sqrt(TF)	1	1.414	1.732	2.000	2.236	2.449	2.828	3.162	3.316	4.123	4.242	4.358	4.582

[그림 3-5] TF, 로그TF, 루트TF 적용 결과 비교

● **IDF (Inverse Document Frequency Weighting)**

▪ Sparck Jones에 의해서 제안된 **단어빈도를 문헌빈도로 나누어 빈도 값을 표준화시킨 상대 빈도 값**으로, 보통 이 역 문헌빈도(Inverse Document Frequency)에 용어 빈도($freq_{ij}$)를 곱하여 단어에 가중치를 부여함

▪ 단어 W_i의 문서 j에서의 빈도수를 $freq_{ij}$라 하고, 총 문서의 개수를 N, 단어 k_i가 있는 문서의 개수를 n_i라고 할 때, 특정 카테고리에서 TF-IDF에 의한 가중치는 다음과 같이 계산됨

$$idf_i = \log \frac{N}{n_i}$$ → 역 문헌(IDF) 가중치

$$W_{ij} = \frac{freq_{ij}}{\max_k freq_{kj}} \times idf_i = \frac{freq_{ij}}{\max_k freq_{kj}} \times \left(\log \frac{N}{n_i} \right)$$ → 역 문헌 빈도(TF-IDF) 가중치

▪ 문서 d_j에서의 단어 k_i의 가중치는 이고 따라서 문서 내에 한번도 출현하지 않은 단어의 가중치는 0이 되고 문서 d_j는 단어 벡터로 표현됨

▪ **단어의 중요도는 그 단어가 출현하는 전체 문헌수에 대해 반비례한다**고 보고 출현빈도가 낮은 저빈도 단어일수록 높은 가중치를 주고, 출현빈도가 높은 고빈도 단어일수록 낮은 가중치 부여함

● **제안1 가중치**

- TF-IDF 가중치의 변형공식으로 **정보검색분야에서 단어추출 기법으로 사용되고 있는 단어 가중치** 중 보정 TF 가중치를 IDF (역 문헌 빈도)에 곱하여 보정TF-IDF 가중치 공식을 적용하였음
- 기존 TF-IDF 공식보다 분류기의 성능 향상에 더 좋은 결과를 나타낼 수 있도록 변형된 공식임
- w값은 여러 쌍을 실험한 결과 0.5를 적용하였을 때 가장 좋은 성능을 나타내었음

$$\left[(1-w)+\frac{w\times n_{i,j}}{\sum n_{k,j}}\right]\times idf_i \longrightarrow idf_i=\log\left(\frac{N}{n_{i,j}}\right)$$

● **제안2 가중치**

- 변형 TF-IDF 가중치는 본래 TF-IDF 가중치 공식에서 **TF(Term Frequency)와 DF(Document Frequency)에 일정 가중치를 더해 줌으로써 DF의 중요도를 반영**한 변형된 가중치 공식임
- 단어 빈도수에 0.5의 가중치를 주고 해당 단어가 출현한 문헌 빈도수에 0.5의 가중치를 준 뒤 두 가중치를 더하여 그 합이 1이 되도록 하였으며 그 후에 역 문헌 빈도수를 다시 곱하여 계산함

$$W_t=\left(n_{i,j}\times w+df_i\times(1-w)\right)\times idf_i$$

TF(Term Frequency) *DF(Document Frequency)*

단계 3. 가중치 부여 cont.

클래스 5개에 모두 적용

단어 / 문서	하반기	하루	하락폭	하락	하나은행	하나	필지	필요	피해자	피해배상	피해
문서1	0	0	0	0	0	0	0	0	0	0	0
문서2	0	0	0	0	0	0	0	0	0	0	0
문서3	18.03651	0	0	0	0	0	0	0	0	0	0
문서4	0	0	0	0	0	0	0	0	0	0	0
문서5	0	0	0	0	0	0	0	0	0	0	0
문서6	0	0	0	0	0	0	0	0	0	0	0
문서7	0	0	0	0	0	0	0	0	11.55038	3.17876	28.65834
문서8	0	0	0	0	0	0	0	0	0	0	0
문서9	0	0	0	0	13.20272	24.83898	0	0	0	0	0
문서10	18.01757	0	0	0	0	0	0	0	0	0	0
문서11	0	0	0	0	0	0	0	0	0	0	0
문서12	0	0	0	0	0	0	0	0	0	0	0
문서13	0	0	5.99673	21.48722	0	0	0	0	0	0	0
문서14	0	0	0	0	0	0	0	0	0	0	0
문서15	0	0	0	0	0	0	0	0	0	0	0
문서16	0	0	0	0	0	0	5.61979	0	0	0	0
문서17	0	0	0	0	0	0	0	0	0	0	0
문서18	0	0	0	0	0	0	0	0	0	0	0
문서19	0	0	0	0	0	0	0	0	0	0	0
문서20	0	0	0	0	13.24903	24.8877	0	0	0	0	0
문서21	0	0	0	0	0	0	0	0	0	0	0
문서22	17.98346	0	0	0	0	0	0	14.88945	0	0	0
문서23	0	0	0	0	0	0	0	0	0	0	0
문서24	0	0	0	0	0	0	0	0	0	0	0
문서25	0	0	0	0	0	0	0	0	0	0	0
문서26	0	0	0	0	0	0	0	0	0	0	0
문서27	0	0	0	0	0	24.86169	0	0	0	0	0
문서28	0	0	0	0	0	0	0	0	0	0	0
문서29	0	0	0	20.95875	0	0	0	0	0	0	0
문서30	18.00999	0	0	0	0	0	0	0	0	0	0

[그림 3-6] 문서표현 결과 예시

- 총 100개의 문서에서 한 문서에 총 30개의 단어가 있고 그 중 상위 20%로 추출된 단어가 아래의 표에 있는 단어라고 가정

TF Freq	term
9	세수
5	올해
4	추경
4	편성
4	내수
3	성장

[표 3-4] 단어의 빈도수

DF Freq	term
5	세수
3	올해
2	추경
6	편성
3	내수
8	성장

[표 3-5] 단어의 문서 빈도수

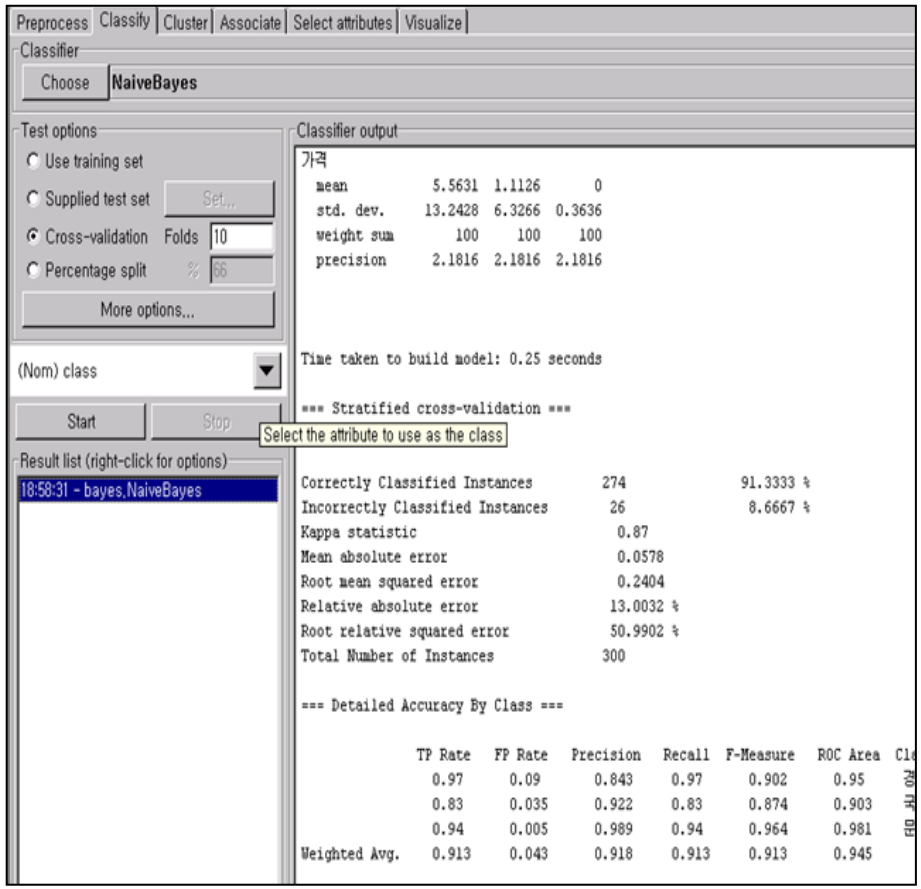
가중치 공식 적용

TF-IDF	TF	IDF
$w_{ij} = tf_{i,j} \times idf_i$	$tf_{i,j} = \frac{n_{i,j}}{\sum n_{k,j}}$	$idf_i = \log\left(\frac{N}{n_{i,j}}\right)$

TFIDF 계산결과 예

$$w_{세수,문서1} = \frac{9}{30} \times \log\left(\frac{100}{5}\right) = 0.3903$$

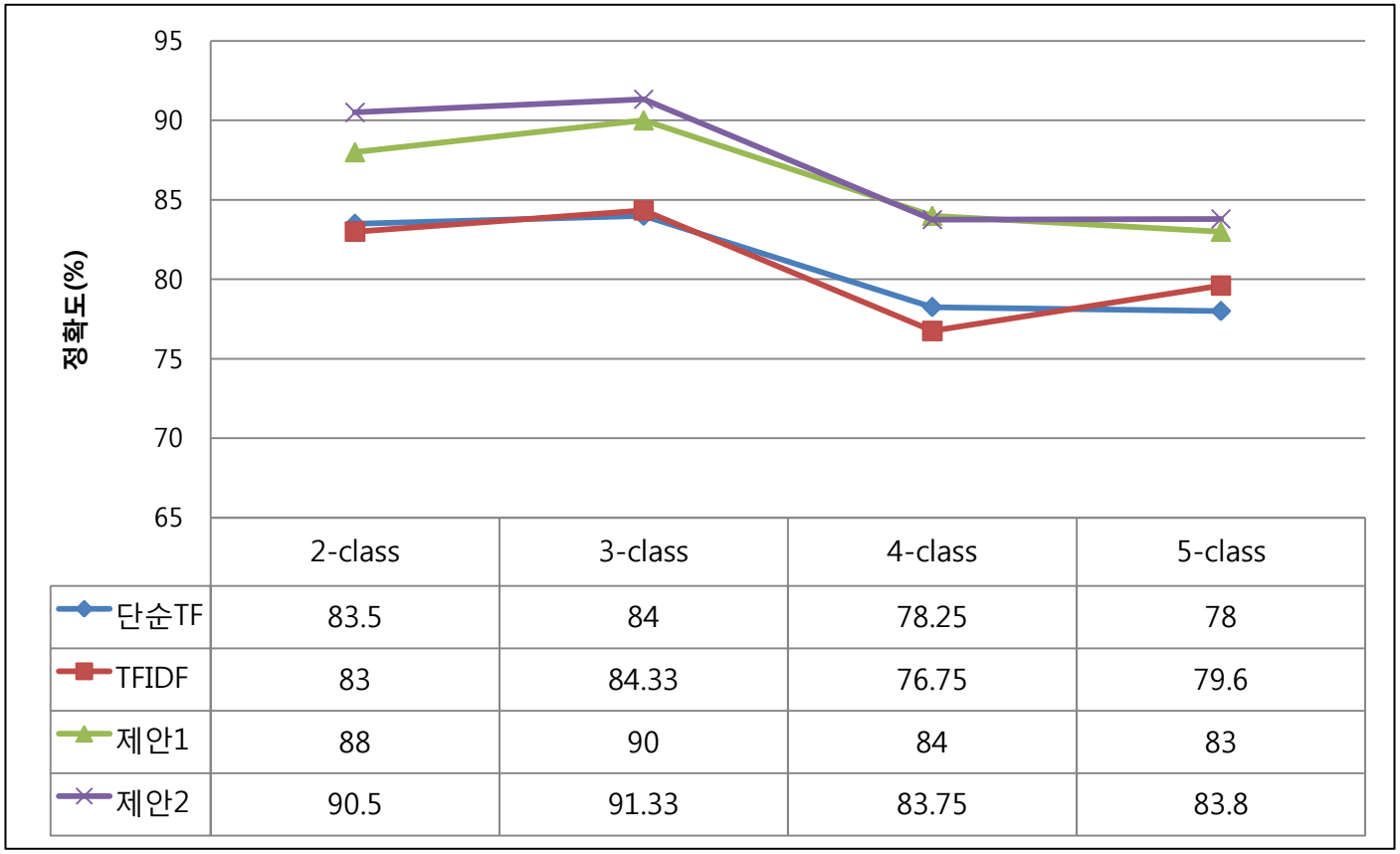
Naïve Bayes Classifier



[그림 3-7] WEKA 문서분류 실행 결과

- WEKA version 3.6.3에서 여러 가지 알고리즘 들 중 Naïve Bayes classifier를 이용하여 문서분류 실험을 하였음
- 실험 평가는 정확도(Precision), 그리고 오분류행렬(confusion matrix)를 이용하여 카테고리 별로 실험 결과를 비교하였음
- 학습방법은 k-fold cross validation을 이용하였으며 10-fold로 실험 하였음
  - k-fold cross validation은 수집된 sample들의 검증을 위한 통계적 분석 방법
  - 전체 집합을 k개로 나눈 뒤 하나를 다른 것들과 비교하여 전체적으로 특이한 집합이 없는지 확인하는 방식이며, 주로 인공지능의 데이터마이닝 연구에서 주로 사용됨
    - ➢ 수집된 최초의 오리지널 샘플을 k개의 서브 sample로 나눔
    - ➢ 하나의 서브 샘플은 모델의 테스트를 위한 validation 데이터로 두고 남은 k-1개의 서브 sample들은 트레이닝 데이터로 사용됨.
    - ➢ 모든 서브 sample들이 validation 데이터로 정확히 한번씩 사용될 때까지 k개의 서브 sample들은 Cross-Validation 프로세스 동안 k 번 반복됨
    - ➢ 프로세스의 각 스텝마다 각 부분으로부터 나온 k개의 결과는 하나의 평가 지표로 만들기 위해 평균을 구하며, 이를 이용해 검증을 수행할 수 있음.

● **4개 가중치 적용 결과 비교**



	2-class	3-class	4-class	5-class
단순TF	83.5	84	78.25	78
TFIDF	83	84.33	76.75	79.6
제안1	88	90	84	83
제안2	90.5	91.33	83.75	83.8

[그림 4-1] 클래스 별 문서분류 정확도 비교 그래프

● 2-class 실험

▪ 10-fold Cross Validation

<Basic TF-IDF >

정분류율	83 %
오분류율	17 %

$$W_t = tf_i \times \log\left(\frac{N}{df_i}\right)$$

class	경제	국제
경제	**89**	11
국제	23	**77**

<confusion matrix>

<제안1 >

정분류율	88.0 %
오분류율	12.0 %

$$W_t = \left[(1-w) + w \times \frac{tf_i}{\sum tf_k}\right] \times \log\left(\frac{N}{df_i}\right)$$

class	경제	국제
경제	**96**	4
국제	19	**81**

<confusion matrix>

<제안2>

정분류율	90.5 %
오분류율	9.5 %

$$W_t = (tf_i \times n + df_i \times (1-n)) \times \log\left(\frac{df_i}{N}\right)$$

class	경제	국제
경제	**97**	3
국제	16	**84**

<confusion matrix>

● 3-class 실험

▪ 10-fold Cross Validation

<Basic TF-IDF >

정분류율	84.33 %
오분류율	15.67 %

$$W_t = tf_i \times \log\left(\frac{N}{df_i}\right)$$

class	경제	국제	문화
경제	**86**	11	3
국제	21	**73**	6
문화	2	4	**94**

<confusion matrix>

<제안1 >

정분류율	90 %
오분류율	10 %

$$W_t = \left[(1-w) + w \times \frac{tf_i}{\sum tf_k}\right] \times \log\left(\frac{N}{df_i}\right)$$

class	경제	국제	문화
경제	**95**	5	0
국제	18	**81**	1
문화	1	5	**94**

<confusion matrix>

<제안2>

정분류율	91.33 %
오분류율	8.67 %

$$W_t = \left(tf_i \times n + df_i \times (1-n)\right) \times \log\left(\frac{df_i}{N}\right)$$

class	경제	국제	문화
경제	**97**	3	0
국제	16	**83**	1
문화	2	4	**94**

<confusion matrix>

● 4-class 실험

▪ 10-fold Cross Validation

<Basic TF-IDF >

정분류율	76.75 %
오분류율	23.25 %

$$W_t = tf_i \times \log\left(\frac{N}{df_i}\right)$$

class	경제	국제	문화	사회
경제	**83**	10	1	6
국제	19	**66**	5	10
문화	1	3	**94**	2
사회	14	15	7	**64**

<confusion matrix>

<제안1 >

정분류율	84 %
오분류율	16 %

$$W_t = \left[(1-w) + w \times \frac{tf_i}{\sum tf_k}\right] \times \log\left(\frac{N}{df_i}\right)$$

class	경제	국제	문화	사회
경제	**94**	3	0	3
국제	14	**73**	2	11
문화	2	3	**94**	1
사회	12	11	2	**75**

<confusion matrix>

<제안2>

정분류율	83.75 %
오분류율	16.25 %

$$W_t = \left(tf_i \times n + df_i \times (1-n)\right) \times \log\left(\frac{df_i}{N}\right)$$

class	경제	국제	문화	사회
경제	**94**	3	0	3
국제	14	**75**	2	9
문화	2	3	**95**	0
사회	15	13	1	**71**

<confusion matrix>

● 5-class 실험

▪ 10-fold Cross Validation

<Basic TF-IDF >

정분류율	79.6 %
오분류율	20.4 %

$$W_t = tf_i \times \log\left(\frac{N}{df_i}\right)$$

class	경제	국제	문화	사회	정치
경제	**87**	7	0	5	1
국제	20	**69**	0	10	1
문화	3	3	**79**	1	14
사회	13	17	0	**63**	7
정치	0	0	0	0	**100**

<confusion matrix>

<제안1 >

정분류율	83 %
오분류율	17 %

$$W_t = \left[(1-w) + w \times \frac{tf_i}{\sum tf_k}\right] \times \log\left(\frac{N}{df_i}\right)$$

class	경제	국제	문화	사회	정치
경제	**92**	4	0	3	1
국제	14	**72**	3	10	1
문화	2	3	**94**	1	0
사회	8	10	1	**74**	7
정치	11	3	2	1	**83**

<confusion matrix>

<제안2>

정분류율	83.8 %
오분류율	16.2 %

$$W_t = \left(tf_i \times n + df_i \times (1-n)\right) \times \log\left(\frac{df_i}{N}\right)$$

class	경제	국제	문화	사회	정치
경제	**93**	2	0	4	1
국제	14	**74**	1	8	3
문화	2	4	**94**	0	0
사회	7	9	2	**72**	10
정치	9	3	2	0	**86**

<confusion matrix>

5 결 론

- **대부분의 문서분류 연구는 자질선택 및 분류기 알고리즘의 특성에 따른 분류기 선택문제로 연구되고 있음**

- **본 연구에서는 문서분류를 하기 위해서 가장 근본적인 문제라고 할 수 있는 단어 가중치 기법에 대하여 연구하였음**
  - 본 연구에서 제안한 가중치 기법으로 실험한 결과 보편적으로 성능이 좋다고 알려지고 많이 사용되고 있는 TF-IDF 기법보다 더 우수한 성능 결과를 보여주었음
  - 본 실험에서는 한글 문서집단인 동아일보 신문기사에 대한 문서분류 실험을 수행하였는데, 각 클래스별 마다 문서의 주제에 특징이 있고, 서로 비슷한 내용의 문서들도 많이 있었음
  - 특히, 국제 카테고리와 사회 카테고리는 황우석 줄기세포 사건 등과 같이 비슷한 주제의 내용을 포함한 문서들이 많이 있어서 문서분류 실험에서 분류성능이 다른 카테고리에 비해 낮았음 – 4-class 실험의 성능이 떨어진 이유

- **향후 과제**
  - 본 실험에서 적용된 가중치 기법들을 이용하여 다른 분류 알고리즘에 적용해보는 실험을 통하여 어떤 분류성능결과가 나오는지 알아볼 수 있음
  - 많은 데이터 문서를 사용하여 증명하는 것도 필요할 것이며, 한글 데이터 외에 영어 데이터를 적용했을 경우 어떤 결과가 나오는지 알아볼 필요가 있음
  - 한글 문서집단에 대한 실험 데이터 구축도 빨리 이루어져서 한글 문서분류의 연구가 앞으로도 많은 부분에서 발전이 되어야 할 것임

감사합니다.