TF-IDF와 Naïve Bayes 분류기를 활용한 문서 분류 기법

유종열 · 현상현 · 양동민

대전대학교, 정보통신공학과

Text Document Classification Scheme using TF-IDF and Naïve Bayes Classifier

Jong-Yeol Yoo · Sang-Hyun Hyun · Dong-Min Yang

Dept. of Information & Communications Engineering, Daejeon University, Daejeon, Korea

E-mail : gum10011@naver.com, hsh8023@lycos.co.kr, dmyang@dju.kr

요 약

최근 디지털 경제의 확산으로 대규모의 데이터들이 생성되는 빅데이터 시대가 도래하고 있다. 이러한 빅데이터에서 비정형 데이터 중에서 기술문서, 기밀문서, 허위정보문서 등 유출 시 심각한 문제가 발생하는 텍스트 문서들이 존재한다. 이러한 문제를 방지하기 위해 비정형 텍스트 문서를 분류하고 처리하는 기술의 필요성이 크게 증가하고 있다. 본 논문에서는 TF-IDF와 Naïve Bayes 문서 분류 기법을 이용하여 비정형 텍스트 문서들을 정확하게 분류하는 기법을 제안한다. 제안된 기법의 성능평가를 위해서 파이썬 라이브러리의 TF-IDF와 Naïve Bayes 분류 기능을 활용하여 문서 분류기를 구현한다.

ABSTRACT

Recently due to large-scale data spread in digital economy, the era of big data is coming. Through big data, unstructured text data consisting of technical text document, confidential document, false information documents are experiencing serious problems in the runoff. To prevent this, the need of art to sort and process the document consisting of unstructured text data has increased. In this paper, we propose a novel text classification scheme which learns some data sets and correctly classifies unstructured text data into two different categories, True and False. For the performance evaluation, we implement our proposed scheme using Naïve Bayes document classifier and TF-IDF modules in Python library, and compare it with the existing document classifier.

키워드

TF-IDF, Navie Bayes, Naïve Bayes, 문서 분류기

Ⅰ. 서 론

현대사회는 IT 기술의 발전과 디지털 경제의 확산으로 인해 대규모의 데이터가 생성되고 있는 빅데이터 시대이다. 이러한 빅데이터 시대에서 생성되는 대규모의 데이터들의 특징은 생성 주기가 짧으며, 형태는 수치, 문자, 그림, 영상 등을 포함하는 대규모 데이터들이다. 디지털 환경에서 생성된 여러 형태의 대규모 데이터들은 정형 데이터와 비정형 데이터로 나눌 수 있다. 정형 데이터는 일정한 규격이나 형태를 지닌 숫자 데이터 등을 말하고, 비정형 데이터는 그림이나 영상, 문서처럼 형태와 구조가 다르며, 구조화되지 않은 데이터들을 말한다.

현재 빅데이터 시대에서 생성된 비정형 데이터를 처리하는 기술의 필요성이 크게 증가하고 있다. 가장 대표적인 비정형 데이터는 텍스트 문서로 모든 분야에서 기본이 되고 일반문서, 행정문서, 기술문서, 비밀문서 등 포함하는 규모와 범위가 가장 넓기 때문에 텍스트 문서를 정확히 분류

하고 관리하는 기술이 중요하다. 텍스트 문서는 특정한 구조와 형태가 정해져 있지 않기 때문에 분석과 처리 기술 과정이 매우 복잡하여 많은 시간이 지연되는 문제점이 발생하고 있다. 그리고 여러 형태의 텍스트 문서 중에 기술문서, 기밀문서, 허위정보문서 등 유출 시 심각한 문제가 발생하는 문서들을 분류하여 관리하는 기술의 필요성이 대두하고 있다.

텍스트 문서를 분류하는 기술은 크게 두 가지로 나누는데 첫 번째로 텍스트 문서의 특징점을 추출하는 기술과 두 번째로 텍스트 문서를 분류하는 기술로 나눈다. 첫 번째, 텍스트 문서의 특징점을 추출하는 기술에는 MI(Mutual Information), TF-IDF(Term Frequency Weighting - Inverse Document Frequency Weighting), Apriori[1], FP(Frequent Pattern)-Growth 같은 기술이 있다. 두 번째, 텍스트 문서를 분류하는 기술에는 NB(Naïve Bayes)[2][3][4], SVM(Support Vector Machine)과 같은 기술이 있다.

본 논문에서는 비정형 데이터에서 가장 대표적인 데이터인 텍스트 문서를 분류하는 문서 분류기에 대해 연구한다. 연구하는 텍스트 문서 분류기의 사용 될 실험 문서는 Enron Email Dataset[5]에서 제공되는 스팸 메일과 햄 메일 텍스트 문서이다. 각 스팸 메일과 햄 메일 텍스트 문서의 형태소를 분석하여 저장한다. 저장되어 있는 분석된 형태소를 통하여 TF-IDF 알고리즘을 이용해 특징점을 추출하고, 추출된 특징점과 Naïve Bayes 분류기를 활용하여 텍스트 문서를 분류하는 텍스트 문서 분류기를 구현한다.

Ⅱ. 관련연구

2.1 TF-IDF [6][7]

TF-IDF는 텍스트 마이닝에서 단어의 특징점을 추출할 때 쓰이는 가장 대표적인 방법 중 하나이며, 여러 문서로 이루어진 문서 집합이 있을 때 문서들에 존재하는 모든 단어들을 특정 문서 내에서 얼마나 중요한지를 나타내주는 통계적 수치이다.

TF(Term Frequency)는 특정 단어가 문서 내에 출현하는 빈도수를 나타내는 값으로, 이 값이 클수록 문서에서 자주 출현하며 중요하다. IDF(Inverse Document Frequency)는 DF의 역으로서, DF(Document Frequency)는 전체 문서 내에 특정 단어를 포함한 문서의 수를 나타내는 값으로 이 값이 클수록 문서에서 자주 출현하며 중요하다고 판단하고, IDF는 값이 클수록 문서의 변별력이 높다는 것을 의미하며, 한 키워드의 가중치인 TF-IDF를 구하는 식은 다음 식(1)과 같다.

$$W = tf \bullet idf \quad (1)$$

- W: 특정 단어의 가중치 값
- tf: 현재 문서의 특정 단어 빈도수
- idf: 특정 단어가 포함된 문서들의 빈도수의 역

2.2 Naïve Bayes 분류기 [8]

Naïve Bayes 분류기는 각 특성들 사이의 독립을 가정하는 베이즈의 정리(Bayes' Theorem)를 적용한 확률 분류기 모델이다. 베이즈의 정리는 매개 변수 x, y가 있을 때, 분류 1에 속할 확률이 $p_1(x,y)$이고, 분류 2에 속할 확률이 $p_{2(x,y)}$일 때, $p_1(x,y) > p_2(x,y)$이면, 이 값은 분류 1에 속하고, $p_1(x,y) < p_2(x,y)$이면, 이 값은 분류 2에 속한다. 이러한 베이즈의 정리를 이용하여, 분리하고자하는 대상의 각 분류별 확률을 계산하고, 그 확률이 더 큰 쪽으로 분류하는 방법이다.

Naïve Bayes 분류기는 학습 문서의 특징들을 분석하여 학습하고, 학습된 정보를 이용하여 새로 입력된 문서가 어떤 범주에 속할 지에 대하여 분류 해주는 방법이다. 문서를 분류하는 방법에 있어서 단순하고 효율적이며 가장 많이 쓰이는 알고리즘 중 하나이다. 이 방법은 카테고리 안의 클래스들을 정의하고 문서가 가지는 특징에 따라 특정 범주에 속할 확률을 계산한다. 그리고 가장 높은 확률을 가지는 범주로 문서를 분류한다. Naïve Bayes는 다음 식(2)과 같다.

$$P(C|F) = \frac{P(F) \times P(F|C)}{P(C)} \quad (2)$$

C는 카테고리에 속한 클래스를 나타내고 F는 특정 단어를 나타낸다. $P(C|F)$는 특정 단어 F가 C 클래스에 포함될 확률을 말한다. $P(C|F)$는 Naïve Bayes에서 결과적으로 구하고자 하는 목적 값이지만 직접 계산할 수 없으므로 $\frac{P(F) \times P(F|C)}{P(C)}$의 수식으로 변형시켜 값을 계산한다. $P(F)$는 전체 문서를 대상으로 특정 단어가 나타날 확률로서 모든 클래스에 대해 같은 값을 가지게 된다. $P(C)$는 전체 문서 중 C 클래스에 속하는 비율이므로 C 클래스에 속하는 문서들의 수를 전체 문서의 수로 나눈 값이 된다. $P(F|C)$는 C 클래스에서 F라는 특정 단어가 나타날 확률이며 문서 중 F라는 특정 단어가 나타나는 문서의 수를 전체 문서의 수로 나눈 값이 된다.

Ⅲ. 시스템 모델

본 논문의 시스템 모델은 두 가지의 과정으로 나누는데, 학습문서를 학습하는 과정과 입력된 문서를 분류하는 과정으로 나뉜다. 학습문서를 학습하는 과정은 그림 1과 같이 스팸 메일과 햄 메일 문서를 형태소 분석 과정을 통해 각각의 문서마다 단어 집합을 생성하고, 생성된 단어 집합을 통

해 TF-IDF를 활용하여 단어별 가중치 값(TF, DF, IDF, TF-IDF)들을 데이터베이스에 저장한다.

새로 입력되는 문서를 분류하는 과정은 그림 2와 같이 입력된 문서를 형태소 분석 과정을 통해 문서의 단어 집합을 생성하고, 미리 학습한 스팸 메일과 햄 메일 데이터베이스에 있는 단어별 가중치 값을 활용하여 Naïve Bayes 분류기로 스팸 메일 카테고리와 햄 메일 카테고리의 확률을 계산하여 문서를 분류한다.

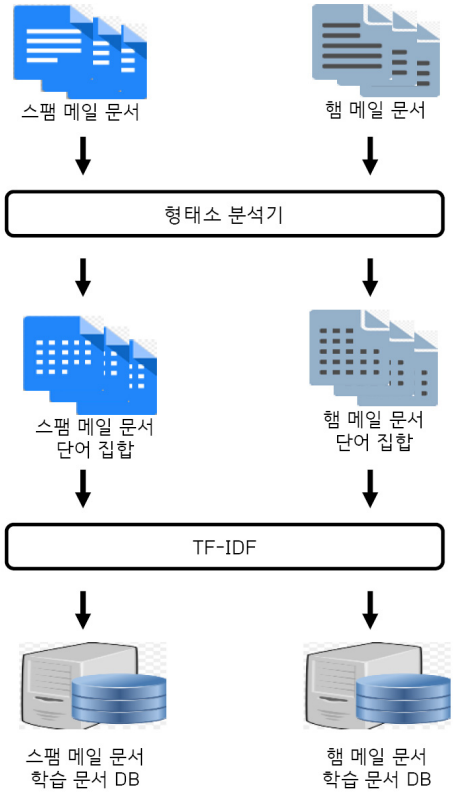

그림 1. 스팸 메일과 햄 메일 문서의 학습과정

3.1 형태소 분석

형태소 분석 단계에서는 학습하는 스팸 메일, 햄 메일과 새로 입력될 메일에 대한 전처리과정인 형태소 분석 작업을 수행한다. 형태소 분석 작업에서는 기본적으로 Python 프로그래밍 언어를 사용한다. 형태소 분석 작업은 Codebox에서 제공하는 Naïve Bayesian Classifier 라이브러리[9]를 활용 한다. Naïve Bayesian Classifier 라이브러리에 포함된 형태소 분석 소스코드를 본 논문의 시스템 모델에 사용하기 위해 학습할 스팸 메일 , 햄 메일과 입력되는 문서의 텍스트 정보를 읽어온다. 읽어온 텍스트 정보를 형태소 분석 소스 코드로 처리하는 과정을 구현한다. 형태소는 의미의 최소 단위로써, 더 이상 분석 불가능한 가장 작은 의미 요소를 뜻하며, 문법적 혹은 관계적인 뜻을 나타내는 단어 또는 단어의 부분이다. 이러한 형태소를 분석하는 이유는 앞서 말한 TF-IDF와 Naïve Bayes 분류기를 활용하기 위한 전처리 작업으로 스팸 메일, 햄 메일, 입력될 메일에 대해서 형태소 단위로 나눈 단어들을 통해 특징점 추출과 문서 분류를 수행한다.

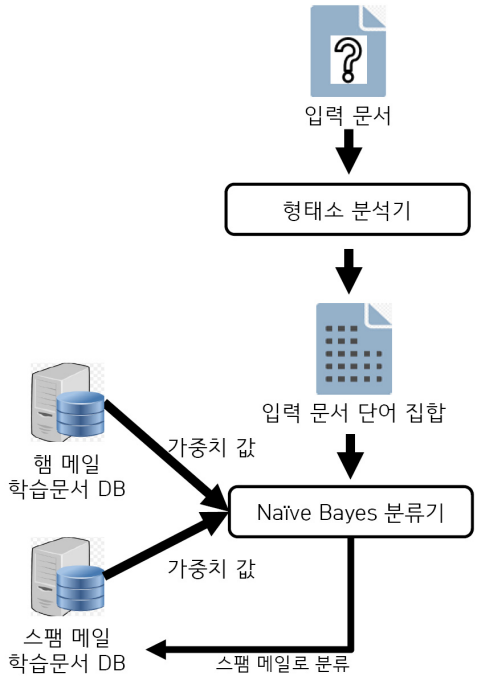

그림 2. 입력된 문서의 분류 과정

3.2 특징점 추출

특징점 추출 단계에서는 형태소 분석된 단어 집합을 활용하여 학습할 문서(스팸 메일, 햄 메일) 및 입력될 문서가 포함하고 있는 단어들의 특징점을 추출하는 과정을 수행한다. 특징점 추출 과정은 특정 문서에 대해 형태소 분석을 수행한 뒤 얻어진 단어 집합들을 활용하여 각 단어의 가중치 값들을 계산하는 과정이다. 특징점 추출 작업에서는 기본적으로 Python 프로그래밍 언어를 사용한다. 특징점 추출 작업은 Hrs에서 제공하는 Python-TF-IDF 라이브러리[10]를 사용한다. 스팸 메일과 햄 메일 텍스트 문서, 입력되는 텍스트 문서들의 각 단어에서 가중치 값들을 추출하기 위해 Python-TF-IDF 라이브러리의 similarities함수를 사용하여 구현한다.

TF는 단어가 그 문서에서 나타난 횟수를 나타낸다. TF 값을 계산하는 식은 다음 식(3)과 같다. $\sum n_{k,j}$는 문서 d_j의 모든 단어의 개수이고, $n_{i,j}$는 문서 d_j에서 단어 t_i가 나타나는 개수이다.

$$tf_i = \log \frac{n_{i,j}}{\sum n_{k,j}} \quad (3)$$

DF는 해당 단어가 나타난 문서의 수를 나타낸다. 각 단어가 그 문서에서 얼마나 나타났는지는 중요하지 않고, 몇 개의 문서에서 나타났는지가 중요하다. DF 값을 계산하는 식은 다음 식(4)와 같다.

$$df_i = \frac{N_i}{N} \quad (4)$$

IDF는 DF를 역수 취한 것이다. DF는 값이 클수록 중요하지 않은 단어를 나타내는 것인데, 이것을 반대로 값이 클수록 중요한 단어로 나타내기 위하여 역수를 취한다. IDF값을 계산하는 식은 다음 식(5)와 같다. N은 전체 문서의 개수이고, N_i은 단어를 포함한 문서의 개수이다.

$$idf_i = \log\frac{N}{N_i} \quad (5)$$

3.3 문서 분류

문서 분류 단계에서는 새로 입력되는 문서를 구별하기 위해 기존의 Naïve Bayes 분류기를 활용한다. 문서 분류 작업에서는 기본적으로 Python 프로그래밍 언어를 사용한다. 문서 분류 작업은 Codebox에서 제공하는 Naïve Bayesian Classifier 라이브러리[9]를 사용한다. Naïve Bayesian Classifier 라이브러리에서 문서를 분류하기 위해 계산을 하는 classify함수의 소스코드를 수정하여 구현한다. 기존의 Naïve Bayes 분류기는 간단하지만 본 논문에서 제안하는 TF-IDF 가중치 값을 활용한 Naïve Bayes 분류기는 문서의 각 단어에서 추출한 가중치를 통해 문서를 분류하는데 정확도를 향상시켜주며, 계산식은 다음 식(6)과 같다.

$$P(C|F) = \frac{P(F)\times P(F|C)}{P(C)} + TF-IDF \quad (6)$$

Ⅳ. 결론

본 논문에서 제안하는 문서 분류 시스템 모델은 Enron Email Dataset을 활용하여 스팸 메일과 햄 메일 문서의 학습을 수행하며, 학습 과정에서 TF-IDF를 활용하여 가중치 값들을 데이터베이스에 저장하고, 기존의 Naïve Bayes 분류기 모델에 TF-IDF 가중치 값을 활용하여 더 정확한 문서 분류를 하는 문서 분류기를 구현하였다. 구현한 문서 분류기는 새로 입력되는 문서를 분류하여 분류한 카테고리에 학습하는 과정을 추가하여 입력되는 문서가 늘어날수록 정확도가 향상된다.

현재는 특징점 추출 단계에서 TF-IDF 알고리즘을 사용하였고, 문서 분류 단계에서는 Naïve Bayes 분류기를 사용하였는데, 차후 더 정확한 문서 분류를 위해 특징점 추출 단계에서는 Apriori 알고리즘, FP-Growth 알고리즘 등을 활용하고, 문서 분류 단계에서는 SVM을 활용하여 문서 분류기를 구현할 계획이다.

Acknowledgement

본 논문은 미래창조과학부의 2015년 고용계약형 SW석사과정 지원사업을 지원받아 수행한 결과입니다. (H0116-15-1007)

참고문헌

[1] 고수정, 이정현, “Apriori 알고리즘에 의한 연관 단어 지식 베이스에 기반한 가중치가 부여된 베이지안 자동 문서 분류”, 멀티미디어학회논문지 제 4권 제 2호, 4월, 2001년

[2] 김현준, 정재은, 조근식, “가중치가 부여된 베이지안 분류자를 이용한 스팸 메일 필터링 시스템”, 정보과학회논문지, 소프트웨어 및 응용 제 31권 제 8호, 8월, 2004년

[3] 김남원, 박진수, “Naïve Bayes 방법론을 이용한 개인정보 분류”, 지능정보연구 제 18권 제 1호, 3월, 2012년

[4] 송철환, 유성준, “문서 분류 알고리즘을 이용한 한국어 스팸 문서 분류 성능 비교”, 한국정보과학회 논문지 제 33권 제 2호, 10월, 2006년

[5] Enron Email Dataset, Available : http://www.aueb.gr/users/ion/data/enron-spam

[6] 고수정, 최준혁, 이정현, “연관 단어 마이닝을 사용한 웹문서의 특징 추출”, 정보과학회논문지, 데이터베이스 제 30권 제 4호, 8월 2003년

[7] 최우식, 김성범, “대칭 조건부 확률과 TF-IDF 기반 텍스트 분류를 위한 N-gram 특질 선택”, 대한산업공학회지 제 41권 제 4호, 8월 2015년

[8] 염하늘, 방재근, 황명권, 황미녕, 정한민, “나이브 베이즈 모델에 기반한 웹 뉴스 기사의 제목 추출” 한국정보과학회 제 41회 정기총회 및 동계학술발표회, 12월, 2014년

[9] Codebox, Naive Bayesian Classifier, Available :https://github.com/codebox/bayesian-classifier

[10] Hrs, Python-tf-idf, Available : https://github.com/hrs/python-tf-idf