Journal of Biomedical Engineering Research 43: 109-115 (2022)
http://dx.doi.org/10.9718/JBER.2022.43.2.109

학 술 논 문

의무 기록 문서 분류를 위한 자연어 처리에서 최적의 벡터화 방법에 대한 비교 분석

유성림

성균관대학교 SAIHST 의료기기산업학과

Comparative Analysis of Vectorization Techniques in Electronic Medical Records Classification

Sung Lim Yoo

Department of Medical Device Management and Research, SAIHST, Sungkyunkwan University, Seoul 06355, Korea
(Manuscript received 22 February 2022 ; revised 14 April 2022 ; accepted 15 April 2022)

Abstract: Purpose: Medical records classification using vectorization techniques plays an important role in natural language processing. The purpose of this study was to investigate proper vectorization techniques for electronic medical records classification. **Material and methods:** 403 electronic medical documents were extracted retrospectively and classified using the cosine similarity calculated by Scikit-learn (Python module for machine learning) in Jupyter Notebook. Vectors for medical documents were produced by three different vectorization techniques (TF-IDF, latent sematic analysis and Word2Vec) and the classification precisions for three vectorization techniques were evaluated. The Kruskal-Wallis test was used to determine if there was a significant difference among three vectorization techniques. **Results:** 403 medical documents were relevant to 41 different diseases and the average number of documents per diagnosis was 9.83 (standard deviation=3.46). The classification precisions for three vectorization techniques were 0.78 (TF-IDF), 0.87 (LSA) and 0.79 (Word2Vec). There was a statistically significant difference among three vectorization techniques. **Conclusions:** The results suggest that removing irrelevant information (LSA) is more efficient vectorization technique than modifying weights of vectorization models (TF-IDF, Word2Vec) for medical documents classification.

Key words: Natural language processing, Medical records classification, Vectorization techniques, Machine learning, Latent semantic analysis

I. 서 론

의료 기관에 전자 의무 기록(electronic medical record)이 도입되면서, 이를 분석하여 질병의 위험 요인을 찾거나 예후를 예측하기 위한 연구들이 진행되고 있다[1-3]. 이러한 연구들은 정형화된 수치(환자의 연령, 가족력, 생체 징후, 검사 결과)에 기반한 것이 대부분이며, 의무 기록 내에서 환자들이 호소한 자연어(natural language)를 기반으로 한 연구는 소수이다.

자연어로 기술된 문서를 행렬 등의 수치형 자료로 변환하면 수학적으로 분석할 수 있으며, 이 과정을 자연어 처리(natural language processing)라고 한다. 자연어 처리를 통해 방대한 문서들을 비슷한 주제로 묶어 분류할 수 있으며, 문서들의 유사도를 비교하여 사용자에게 비슷한 문서를 추천해 주거나, 특정 정보를 추출하는 것과 같은 유의미한 정보를 제공할 수 있다[4,5]. 자연어 처리를 위해 다양한 알고리즘을 사용할 수 있지만, 최근에는 기계 학습(machine learning)을 적용하려는 시도가 있다[6,7].

기계 학습 알고리즘은 입력층, 은닉층 및 출력층으로 구성된다. 은닉층에서 활성화 함수(activation function)의 가

*Corresponding Author : Sung Lim Yoo
Department of Medical Device Management and Research, SAIHST, Sungkyunkwan University, Seoul 06355, Korea
Tel: +82-2-2148-7799
E-mail: yoosl@catholic.ac.kr

중치(weight) 와 편향(bias)이 알고리즘에 의해 최적화되면 기계 학습이라 정의한다[8-10]. 알고리즘에서 출력되는 값과 실제값의 차이를 비용 함수(cost function)로 정의하고, 이를 최소화하기 위해 옵티마이저(optimizer)를 이용하여 역전파(backpropagation)하면 가중치와 편향을 조절할 수 있다.

최근 기계 학습 알고리즘을 적용하여 환자들의 검사 결과 판독지를 분류하거나, 정신건강의학과에 내원하는 환자들의 전자 의무 기록을 병리 상태에 따라 분류하여 의미 있는 연구 결과를 발표하였다[11,12]. 또한, 방대한 전자 의무 기록에서 환자들의 흡연력을 추출하는데 다양한 기계 학습 알고리즘을 적용하여 0.9 이상의 정확도를 보고하기도 하였다[13].

본 연구는 환자들의 증상을 자연어로 기술한 의무 기록에 대해 세 가지 방법으로 자연어 처리를 하여 분류하고, 이에 대한 정밀도를 비교함으로써 최적의 벡터화 방법을 찾고자 하였다.

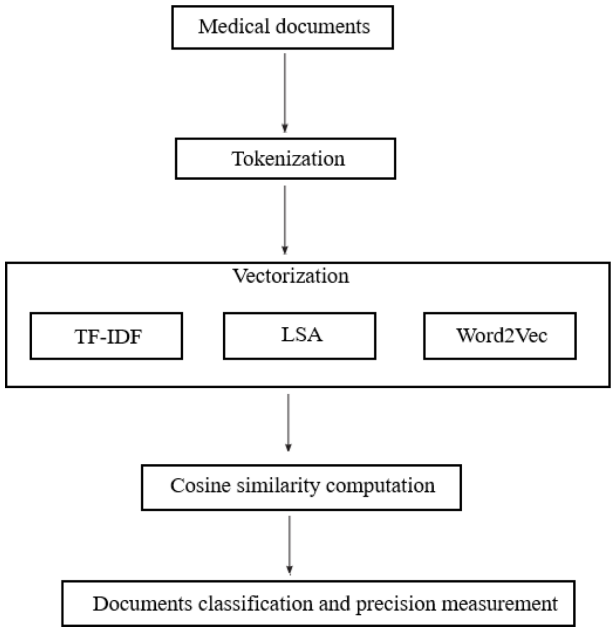

그림 1. 의무 기록 분류 및 정밀도 산출에 대한 개요
Fig. 1. Flow diagram of documents classification and precision measurement

II. 연구 방법

2021년 10 월부터 2022년 2 월까지 단일 의료기관에 내원한 환자들의 전자 의무 기록 403 개를 대상으로 후향적 연구를 시행하였다. 전자 의무 기록 중 초진 기록 만을 대상으로 하여 현병력을 하나의 문단으로 만들어 데이터 세트를 추출하였으며, 초진 기록이 영어로 작성되었거나, 진단명이 감염성 질환 또는 외상인 경우는 연구에서 배제하였다.

의무 기록에서 추출된 403 개의 현병력 및 진단명의 데이터 세트는 진단명에 색인을 하여 식별이 가능하게 하였다. 또한, 검증용 데이터 세트(n=61)는 진단명의 비율에 따라 무작위 배정한 후 별도로 구분하였다. 진단명이 같은 데이터 세트의 수는 15개를 초과하지 않도록 해당 진단명의 의무 기록을 배제하여 편차를 최소화하였다. 총 403개의 의무 기록에 대해 그림 1에서 도시한 것처럼 문서의 토큰화, 벡터화, 유사도 계산, 문서 분류 및 정밀도 산출의 과정으로 연구를 진행하였다.

연구에는 프로그래밍 언어로 파이썬(Python, version 3.8.5), 프레임 워크로 케라스(Keras, version 2.4.3)를 사용하였고, 가상 환경은 아나콘다(Anaconda, version 2020.11, Continuum Analytics, Texas, USA), 구현 환경으로는 Jupyter Notebook(version 6.1.4)을 이용하였다. 본 연구는 본교 생명윤리위원회(IRB)의 승인을 받고 진행되었다(SKKU202112019).

1. 의무 기록의 토큰화(tokenization)

전자 의무 기록 문서에 대해 자연어 처리를 하기 위해 토큰화 과정을 먼저 거쳐야 한다. 토큰화는 문서 안의 단어들을 의미 표현의 기본 단위인 토큰(token)으로 분리하여 추출해내는 과정이다. 한글의 경우에 의미 표현의 기본 단위는 형태소이므로, 의무 기록 문서에서 형태소들을 추출하여야 토큰화를 할 수 있다. 이를 위해 파이썬 기반 한국어 정보 처리 패키지인 Korean natural language processing in python(KoNLPy)의 한글 형태소 분석기를 활용하였다. KoNLPy에 내장된 Open Korean Text 형태소 분석기를 사용하여 의무 기록에서 특수 문자 및 구두점을 제거하고, 형태소들을 모두 추출하였다. 또한, 동사 또는 형용사를 추출할 때 같은 의미를 지닌 단어임에도 어미 변화 때문에 별도의 토큰으로 분석되지 않도록, 어간 추출(stemming)을 하여 토큰화 하였다.

2. 문서의 벡터화(text vectorization)

의무 기록 문서를 토큰화한 후 단어 빈도-역 문서 빈도(Term Frequency-Inverse Document Frequency), 잠재 의미 분석(Latent Semantic Analysis) 및 Word2Vec의 세가지 다른 방법으로 벡터화하여 분석하였다.

(1) 단어 빈도-역 문서 빈도(TF-IDF)

단어 빈도-역 문서 빈도는 문서-단어 행렬(Document-Term Matrix)에 단어의 중요도에 따라 가중치를 부여하는 방법이다. 문서-단어 행렬은 문서 내 존재하는 모든 단어에

대해 빈도만을 고려하여 행렬로 표현한 것이며, 단어의 중요도를 반영하지 못 한다는 단점이 있다. 이런 단점을 보완하기 위해 단어의 중요도를 반영할 수 있는 단어 빈도-역 문서 빈도라는 방법을 사용한다[14,15]. 식 (1)과 같이 특정 단어가 등장하는 문서의 빈도[DF(t)]에 반비례하고 전체 문서의 수(n)에 비례하도록 로그 함수를 취하여 IDF(t, D)를 구하고, 식 (2)와 같이 특정 문서에 등장하는 단어의 빈도[TF(t, d)]에 비례하도록 추가 가중치를 주면 TF-IDF가 계산된다.

$$\mathrm{IDF(t,D)} = \log\frac{n}{1+DF(t)} \quad (1)$$

$$\mathrm{TF-IDF} = \mathrm{TF(t,d)} \cdot IDF(t,D) \quad (2)$$

계산된 TF-IDF를 문서-단어 행렬에 곱해 주면 TF-IDF 행렬로 변환된다. 특정 문서에만 자주 등장하는 단어는 중요도가 높게, 여러 문서에 등장하는 단어는 중요도가 낮게 가중치가 부여된 것이 TF-IDF 행렬이다[14,15].

TF-IDF 행렬을 만들면 여러 문서들의 평균 벡터를 계산하고 비교할 수 있다. 본 연구에서는 구현 환경 Jupyter Notebook에서 파이썬 기계 학습 라이브러리 사이킷런(Scikit-learn)의 TfidfVectorizer를 사용하여, 의무 기록 문서를 TF-IDF 행렬로 만들어 벡터화 하였다.

(2) 잠재 의미 분석(Latent Semantic Analysis)

잠재 의미 분석은 문서-단어 행렬에 대해 특이값 분해(Singular Value Decomposition)를 하여 중요도가 낮은 정보는 제거하고, 상대적으로 중요도가 높은 정보에 가중치를 부여하는 방법이다[14,15]. 식 (3)과 같이 행렬을 세 개의 행렬 곱으로 분해하는 것을 특이값 분해라고 한다. 즉, 행렬 A (m × n)에 대해 특이값 분해를 하면, 직교 행렬 U(m × m), 대각 행렬 Σ(m × n), 직교 행렬 V^T(n × n)의 곱셈으로 표현할 수 있다. 식 (3)에서 대각 행렬 Σ의 원소들을 특이값(singular value)이라 하며, 이는 내림차순으로 정렬된다.

$$A(m \times n) = U(m \times m) \cdot \Sigma(m \times n) \cdot VT(n \times n) \quad (3)$$

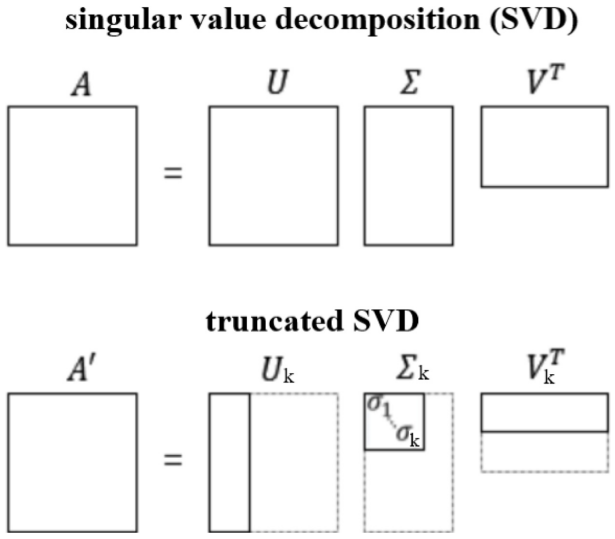

그림 2. 선형대수학의 특이값 분해와 초매개변수 K에 따른 절단
Fig. 2. Singular value decomposition and truncated singular value decomposition

그림 2와 같이 특이값 중 상위 K 개만 남기고 절단한 후, 다시 행렬 곱셈을 하면 행렬 A'(m × n)을 만들 수 있다. 그리고 행렬 A'의 원소들을 비교하여 분석하면 기존의 행렬 A (문서-단어 행렬)에서는 드러나지 않던 잠재된 의미를 보여줄 수 있다[14,15]. 이는 특이값 중 하위값을 절단하면 상대적으로 중요하지 않은 정보들을 제거할 수 있기 때문이며, K는 초매개변수로서 연구자가 결정하는 값이다.

본 연구에서는 의무 기록 문서에 대해 사이킷런의 TfidfVectorizer를 사용하여 문서-단어 행렬을 만들었으며, 이 행렬에 대해 파이썬 라이브러리 SciPy를 이용하여 특이값 분해 후 절단(Truncated Singular Value Decomposition, 초매개변수 K=20)하여 벡터화 하였다.

(3) 기계 학습(Word2Vec)

데이터 세트에 대해 구현 환경 Jupyter Notebook에서 Word2Vec으로 학습시킨 후, 임베딩 벡터의 평균을 계산하였다. Word2Vec은 자연어 처리를 위한 기계 학습 알고리즘 중 하나이며, 입력층, 은닉층 및 출력층으로 구성된다[16,17]. 그림 3에서 도시한 것과 같이 Word2Vec은 입력값으로 원-핫 벡터(one-hot vector)를 사용하며, 가중치(W)와 추가 가중치(W')를 반영하여 Softmax(활성화 함수)를 통해 결과값을 출력한다. Softmax는 다중 분류에 사용되는 함수로서 0과 1 사이의 실수로 구성된 벡터를 확률로서 출력한다[16,17].

Word2Vec 알고리즘은 최적의 가중치를 찾아 내기 위해 비용 함수로서 크로스 엔트로피(cross entropy)를 사용하여 학습한다. 또한, 그림 4에서 도시한 것처럼 가중치(W)에 따라 은닉층에서 계산되는 평균 벡터의 차원이 축소되는데, 이를 임베딩 벡터(embedding vector, v)라고 한다[17]. 알고리즘이 학습을 반복하여 가중치를 조절하면, 은닉층에서 계산되는 임베딩 벡터의 값도 변하게 된다. 이 과정에서 연구자는 알고리즘의 학습 횟수를 정할 수 있고, 학습이 완료되면 임베딩 벡터를 계산하여 비교할 수 있다.

본 연구에서 Word2Vec의 학습은 훈련용 데이터 세트(n=342)를 사용하였으며, 검증용 데이터 세트(n=61)와 분리하여 시행하였다. 초매개변수로서 벡터의 차원(size=100), 분석 단위 수(window=5), 단어 최소 빈도수 제한(min_count=1), 학

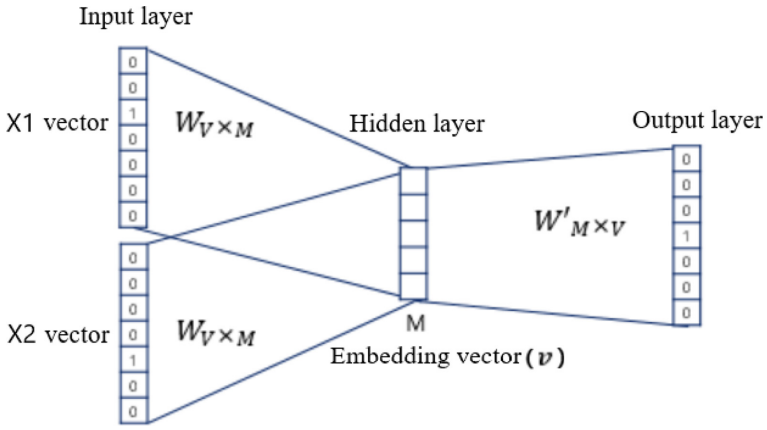

그림 3. Word2Vec의 입력층, 은닉층 및 출력층
Fig. 3. Three layers of Word2Vec

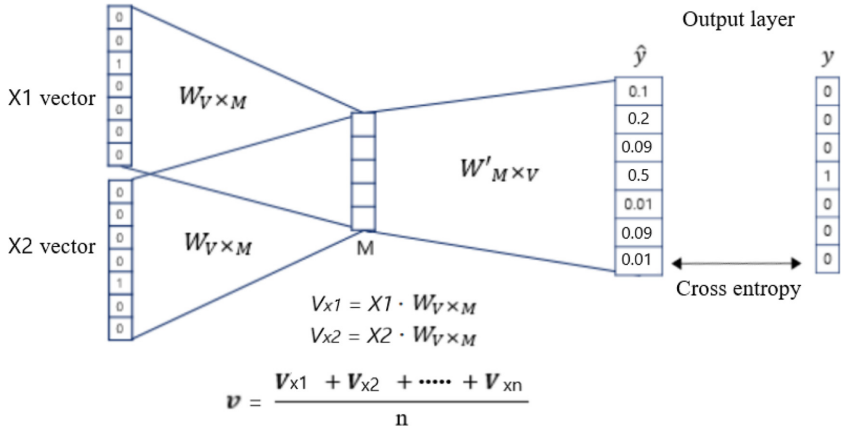

그림 4. Word2Vec에서 임베딩 벡터(v) 계산
Fig. 4. Embedding vector(v) of Word2Vec

습을 위한 중앙처리장치의 수(workers=4)를 설정하여 총 100 에포크(epochs) 학습시켰다.

3. 유사도 계산 및 정밀도 산출

403개의 의무 기록 현병력에 대해 위의 세 가지 방법(TF-IDF, 잠재 의미 분석 및 Word2Vec)으로 벡터화한 후, 파이썬 라이브러리 사이킷런을 이용하여 코사인 유사도를 계산하였다. 코사인 유사도는 식 (4)와 같이 두 벡터(A, B)의 내적을 각 벡터의 노름(norm) 곱으로 나눈 값이다.

$$\text{Cosine similarity} = \frac{\vec{A}\,\vec{B}}{|A|\cdot|B|} \tag{4}$$

계산된 코사인 유사도 데이터를 기반으로 분류 알고리즘을 만들어 실행시키고, 정밀도(precision)를 산출하였다. 위의 알고리즘에 검증용 데이터 세트의 진단명을 입력하면, 코사인 유사도가 높은 순으로 전체 데이터 세트를 정렬시킨 후 상위값만 분리하여 확인할 수 있다. 코사인 유사도 순으로 상위값만 분리하여 정렬시키는 것을 분류로 간주하였고, 여기에 파이썬의 sorted 함수를 사용하였다. 전체 데이터 세트에 대해 입력값(검증용 데이터 세트의 진단명)에 해당하는 진단명의 총 개수만큼 분류하고, 분류된 진단명이 서로 일치하는지를 확인하면 정밀도를 산출할 수 있다. 정밀도는 식 (5)와 같이 산출하였다.

$$\text{Precision} = \frac{\text{True positive}}{\text{True positive} + \text{False positive}} \tag{5}$$

본 연구에서 정확도는 분석에서 배제하였다. 참 음성 값이 참 양성 값에 비해 상대적으로 커서, 분류된 데이터 세트의 진단명이 모두 틀린 경우에도 정확도가 높게 산출되었기 때문이다.

세 가지 방법으로 의무 기록 문서들을 벡터화한 후, 분류된 데이터 세트의 정밀도에 유의한 차이가 있는지 SPSS statistics(IBM SPSS statistics for windows, Armonk, NY)를 이용하여 분석하였다.

III. 결 과

총 403개의 데이터 세트에서 진단명은 41개였으며, 진단명 한 개당 현병력 문서는 평균 9.83개, 표준 편차는 3.46이었다. 진단명의 빈도는 adhesive capsulitis 15개(3.72%), lateral epicondylitis 15개(3.72%), plantar fasciitis 15개(3.72%) 순으로 높았으며, 표 1에 정리하였다.

세 가지 벡터화 방법에 대해 검증용 데이터 세트 61개를 분류 알고리즘에 입력하여, 총 183개의 정밀도 결과를 산출하였다. 이에 대한 정규성 검정 결과가 정규 분포를 따르지 않았기 때문에, 독립표본 Kruskal-Wallis 검정을 이용하여 세 가지 방법에 따른 정밀도에 유의한 차이가 있는 지 분석하였다. 표 2와 같이 TF-IDF를 이용하여 데이터 세트를 분류한 결과 정밀도는 평균 0.789, 표준 편차는 0.16 이었으며, 잠재 의미 분석을 이용한 경우는 정밀도 평균 0.876, 표준 편차는 0.151, Word2Vec을 이용하면 정밀도 평균 0.792, 표준 편차는 0.173 이었다.

독립표본 Kruskal-Wallis 검정을 한 결과, P 값은 0.001 이하로 세가지 벡터화 방법에 따른 정밀도에 유의한 차이가 있었다. 또한, 표 3의 사후 검정 결과를 보면 잠재 의미 분석은 다른 두 가지 방법에 비해 정밀도가 유의하게 높았으며, TF-IDF와 Word2Vec 사이에는 유의한 차이가 없었다.

표 1. 전체 의무 기록 문서의 진단명에 따른 통계
Table 1. Descriptive statistics of medical documents

Diagnosis	N	Percentage (%)
Adhesive capsulitis	15	3.72
Lateral epicondylitis	15	3.72
Plantar fasciitis	15	3.72
Knee osteoarthritis	15	3.72
Trigger finger	15	3.72
Hand osteoarthritis	15	3.72
Lumbar spinal stenosis	14	3.47
Carpal tunnel syndrome	14	3.47
Gouty arthritis	14	3.47
Cervical spinal stenosis	13	3.26
Etc.(31)	258	64.01
Total (41)	403	100

표 2. 세 가지 방법에 따른 정밀도 비교 분석
Table 2. The precision comparison between three groups

	Precision		
	TF-IDF	LSA	Word2Vec
Average	0.789	0.876	0.792
Standard deviation	0.160	0.151	0.173
P-value		0.001	

표 3. 세 가지 방법의 정밀도 차이에 대한 사후 검정
Table 3. The precision differences between three groups (post-hoc analysis)

Analysis methods		P-value
TF-IDF	LSA	0.001
	Word2Vec	0.736
LSA	TF-IDF	0.001
	Word2Vec	0.003
Word2Vec	TF-IDF	0.736
	LSA	0.003

IV. 고찰 및 결론

파이썬 라이브러리 SciPy를 이용하여 문서-단어 행렬에 대해 특이값 분해할 때, 초매개변수인 K 값을 어떻게 정하느냐에 따라 정밀도 결과가 달라졌다. 본 연구에서는 K 값을 20으로 할 때, 잠재 의미 분석을 이용한 의무 기록 문서 분류의 정밀도가 가장 높게 산출되었다. 그림 2에서 도시한 것처럼 문서-단어 행렬에 대해 특이값 분해한 후 절단을 할 때, K 값의 크기를 설정하여 특이값이 제거되는 정도를 결정할 수 있다[14,15]. K 값을 작게 하면 특이값이 상대적으로 많이 제거된다. 따라서 새로운 행렬은 원래 행렬의 벡터와 비교하여 많이 변화하게 되며, 이는 정보의 손실 또는 제거를 의미한다. 반대로 K 값을 크게 하면, 특이값을 많이 남기게 되므로 새로운 행렬은 이전 행렬의 벡터와 비슷하게 된다. 이 경우에 정보의 손실은 적지만, 관련성이 적거나 불필요한 정보까지 포함되어 유의미한 분석이 안 될 수 있다[18,19].

최근 PubMed에 존재하는 논문들에 대해 잠재 의미 분석으로 유사도를 계산하여 발표한 연구에 따르면, 저자들은 특이값 분해를 할 때 K 값을 6, 7, 8, 9, 10으로 하여도 결과에 큰 차이가 없음을 확인하였다[20]. 이에 대해 위 논문의 저자들은 K 값에 따라 절단하면 정보의 손실은 있지만, 새롭게 만들어진 행렬이 기존 행렬의 벡터에 근사하기 때문에 결과에 차이가 없다고 분석하였다[20]. 즉, 대각행렬 Σ의 원소인 특이값은 내림 차순으로 정렬되므로, K 값에 따라 절단할 지라도 상대적으로 중요도가 낮은 정보들이 먼저 제거되는 것이다. 따라서 K 값을 적절히 조절하면 관련성이 낮은 정보들을 제거하고, 중요한 정보만을 남긴 후 분석할 수 있어 정밀도를 올릴 수 있다. 본 연구에서는 K 값이 10보다 작으면, 중요도가 높은 정보까지 제거되어 정밀도가 0.6

이하로 계산되었다. 또한, K 값을 20 보다 크게 설정하면, 단어 빈도-역 문서 빈도를 이용하여 산출된 정밀도와 차이가 없게 되었다.

기계 학습(Word2Vec)을 이용하여 의무 기록 문서들의 평균 벡터를 계산할 때, 학습의 횟수(에포크)에 따라 산출된 정밀도 결과에 차이가 있었다. 본 연구에서 Word2Vec 알고리즘의 학습을 100 에포크까지 증가시키면 정밀도가 높아졌지만, 그 이상으로 에포크를 높게 설정하여도 정밀도에 차이가 없었다. 특히, 50 에포크 이하에서는 정밀도가 0.5 이하로 산출되었으며, 100 에포크까지 학습의 횟수를 증가시키면 정밀도가 점차 증가하였다. 그러나 잠재 의미 분석의 정밀도를 능가하지는 못 하였다.

문서-단어 행렬은 희소 표현(sparse presentation)의 일종으로서, 단어의 개수가 증가하면 벡터의 차원도 무한정 높아지게 되는 단점이 있다[14,15,17]. 본 연구에서 문서-단어 행렬의 차원은 403x527 이었고, 이 행렬을 기반으로 두 가지 벡터화 방법(단어 빈도-역 문서 빈도 및 잠재 의미 분석)을 적용하여 데이터 세트를 분류하였다. Word2Vec 알고리즘을 이용한 벡터화 방법에서는, 초매개변수인 임베딩 벡터의 차원을 100으로 축소하여 데이터 세트를 분류하였다. 임베딩 벡터는 희소 표현과 대비되는 개념인 밀집 표현(dense presentation)으로서, 의미가 비슷한 단어를 유사한 벡터로 조밀하게 표현하여 차원을 줄일 수 있다[16,17]. 본 연구에서 임베딩 벡터의 크기를 60, 70, 80, 90, 100, 110, 120으로 다르게 설정하여도, Word2Vec 알고리즘으로 산출된 정밀도에 차이가 없었다. 따라서 기계 학습의 경우에 다른 초매개변수보다 학습의 횟수가 결과에 가장 많은 영향을 주는 요소임을 확인하였다.

문서 안의 단어들에 대해 벡터화 방법을 어떻게 설정하는지에 따라, 문서 분류의 정밀도에 유의한 차이가 있었다. 잠재 의미 분석은 초매개변수인 K 값에 따라 중요도가 낮은 정보들을 제거하여 벡터화하는 것으로, 다른 방법에 비해 의무 기록 문서 분류의 정밀도가 높았다. 이는 의무 기록 문서들을 분류할 때 중요도가 낮은 정보들을 제거하는 것이, 가중치를 조절하는 것보다 우월한 결과를 얻을 수 있음을 의미한다. 최근에는 문서들을 분류할 때 데이터 필터를 적용하여, 관련성이 낮은 정보들을 더 섬세하게 제거함으로써 정밀도를 높이는 방법도 연구되고 있다[21].

최근 연구에서 서포트 벡터 머신(Support vector machine)이나 합성곱 신경망(Convolutional neural network)을 적용한 기계 학습으로 문서들을 분류하는 알고리즘을 새롭게 제안하고 있다[22-24]. 본 연구에서 분류 알고리즘은 코사인 유사도 계산 후 상위값을 정렬시키는 것으로, 비교적 단순하게 설계하였다. 왜냐하면 연구의 목적이 새로운 분류 알고리즘을 제안하는 것이 아니라, 의무 기록 문서에 대한 최적의 벡터화 방법을 찾는 것이기 때문이다. 이를 위해 문서 안의 단어들에 대해 어미 변화는 배제하고, 불용어 및 구두점을 제거하는 등 최대한 정제한 후 토큰화 하였다.

정교한 토큰화의 중요성은 영어보다 한국어에서 더 크다. 왜냐하면 영어로 기술된 문서는 파이썬 자연어 처리 패키지 Natural Language Toolkit 등을 이용하여 띄어쓰기 단위로 토큰화 가능하지만, 한국어로 된 문서는 그럴 수 없기 때문이다. 한국어는 조사 및 어미 변화가 있는 교착어이므로, 띄어쓰기 단위로 토큰화 하면 같은 의미임에도 다른 토큰으로 구분되는 단어들이 생기게 된다[25,26]. 또한, 영어에서는 언어 구조상 띄어쓰기를 지키지 않으면, 의미가 전달되지 못 하는 경우가 많아서 비교적 잘 지켜지는 편이지만, 한국어의 경우는 그렇지 않다. 따라서 한국어로 기술된 문서에 대해 띄어쓰기를 기준으로 토큰화하는 것은 부적절하다[25,26]. 정교한 토큰화가 되어야 최적의 벡터로 표현할 수 있으므로, 한국어에서 정교한 토큰화 및 최적의 벡터화를 위한 연구가 더 필요하다. 의무 기록 문서 분류 알고리즘으로 별도의 심층 신경망을 적용하지 않았지만, 적절한 벡터화만으로도 정밀도 0.876의 결과를 얻을 수 있었다. 이는 자연어 처리에서 벡터화 방법이 분류 알고리즘만큼 중요하다는 것을 의미한다.

의무 기록 문서를 분류할 때 잠재 의미 분석은 정밀도를 높일 수 있는 벡터화 방법이다. 추후 심층 신경망을 포함한 기계학습 알고리즘을 추가하여 의무 기록 문서 분류의 신뢰도를 높일 수 있을 것으로 기대한다.

References

[1] Chicco D, Lovejoy CA, Oneto L. A machine learning analysis of health records of patients with chronic kidney disease at risk of cardiovascular disease. IEEE Access. 2021;9(3): 165132-44.

[2] Blakey JD, Price DB, Pizzichini E. Identifying risk of future asthma attacks using UK medical record data: A respiratory effectiveness group initiative. J Allergy Clin Immunol. 2017; 5(4):1015-24.

[3] Tomasallo CD, Hanrahan LP, Tandias A. Estimating Wisconsin asthma prevalence using clinical electronic health records and public health data. Am J Public Health. 2014;104(1):65-73.

[4] Spasic I, Livsey J, Keane JA. Text mining of cancer-related information: Review of current status and future directions. Int J Med Informatics. 2014;83(9):605-23.

[5] Jonnalagadda SR, Adupa AK, Garg RP. Text mining of the electronic health record: An information extraction approach for automated identification and subphenotyping of HFpEF patients for clinical trials. J Cardiovasc Transl Res. 2017;10(3): 313-21.

[6] Rahaman T. Discovering new trend and connections: Current application of biomedical text mining. Med Ref Services

Quarterly. 2021;40(3):329-36.
[7] Le Glaz A, Haralambous Y, Kim D. Machine learning and natural language processing in mental health: Systemic review. J Med Internet Res. 2021;23(5):15708.
[8] Shai SS, Shai BD. Understanding machine learning: from theory to algorithms. New York: Cambridge University Press; 2014.
[9] Peter F. Machine learning: the art and science of algorithms that make sense of data. Cambridge: Cambridge University Press; 2012.
[10] Mehryar M, Afshin R, Ameet T. Foundations of machine learning. Cambridge: MIT press; 2012.
[11] Chen MC, Ball RL, Yang L. Deep learning to classify radiology free-text reports. Radiology. 2018;286(3):845-2.
[12] Pak DH, Hwang MG, Hwang JU. Application of text classification based machine learning in prediction psychiatric diagnosis. Korean J Biol Psychiatry. 2020;27(1):18-26.
[13] Andrea C, Leif J, Hercules D. Natural language processing and machine learning to enable automatic extraction and classification of patients' smoking status from electronic medical records. Upsala J Med Sci. 2020;125(4):316-24.
[14] Yuli V. Natural language processing with Python and spaCy: a practical introduction. San Francisco: No Starch Press; 2020.
[15] Hobson L, Cole H, Arwen G. Natural language processing in action: understanding, analyzing and generating text with Python. Shelter Island, NY: Manning Publications Co.; 2019.
[16] Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. In Proceedings of Workshop at ICLR. 2013;1301-13.
[17] Gastaldi JL. Why can computers understand natural language? The structuralist image of language behind word embeddings. Phil Tech. 2021;34(1):149-214.
[18] Guillermo JB, Ricardo O, Jose AL. Using latent semantic analysis and the predication algorithm to improve extraction of meanings from a diagnostic corpus. Span J Psychol. 2009; 12(2):424-40.
[19] Zhou Y. An introduction to text classification with applications to medical records. 2nd international conference on informational technology and computer application. 2020;471-75.
[20] Kherwa P, Bansal P. Latent semantic analysis: an approach to understanding sematic of text. International conference on current trends in computer, electrical, electronics and communication. 2017;870-4.
[21] Almas J, Qamar U. Affect of data filter on performance of latent semantic analysis based research paper recommender system. 5th International conference on computational intelligence and application. 2020;50-54.
[22] Weng WH, Wagholikar KB, McCray A., Szolovits P. Medical subdomain classification of clinical notes using a machine learning based natural language processing approach. BMC med inform Decis Mak. 2017;17(1):1-13.
[23] Jamaluddin M, Wibawa AD. Patient diagnosis classification based on electronic medical record using text mining and support vector machine. International seminar on application for technology of information and communication. 2021; 243-8.
[24] Wang Y, Sohn SH, Liu S, Shen F. A clinical text classification paradigm using weak supervision and deep representation. BMC med inform Decis Mak. 2019;19(1).
[25] Park KB, Lee JH, Jang SB, Jung DW. An empirical study of tokenization strategies for various Korean NLP tasks. Computer Science. 2020.
[26] Cho DB, Lee HY, Kang SS. An empirical study of Korean sentence representation with various tokenization. Electronics. 2021;10(7):845-57.