ISSN 2288-4866 (Print)
ISSN 2288-4882 (Online)
http://www.jiisonline.org
J Intell Inform Syst 2019 September: 25(3): 19~41 http://dx.doi.org/10.13088/jiis.2019.25.3.019

복합 문서의 의미적 분해를 통한 다중 벡터 문서 임베딩 방법론

박종인
국민대학교 비즈니스IT전문대학원
(*pji4202@kookmin.ac.kr*)

김남규
국민대학교 경영대학 경영정보학부
(*ngkim@kookmin.ac.kr*)

텍스트 데이터에 대한 다양한 분석을 위해 최근 비정형 텍스트 데이터를 구조화하는 방안에 대한 연구가 활발하게 이루어지고 있다. doc2Vec으로 대표되는 기존 문서 임베딩 방법은 문서가 포함한 모든 단어를 사용하여 벡터를 만들기 때문에, 문서 벡터가 핵심 단어뿐 아니라 주변 단어의 영향도 함께 받는다는 한계가 있다. 또한 기존 문서 임베딩 방법은 하나의 문서가 하나의 벡터로 표현되기 때문에, 다양한 주제를 복합적으로 갖는 복합 문서를 정확하게 사상하기 어렵다는 한계를 갖는다. 본 논문에서는 기존의 문서 임베딩이 갖는 이러한 두 가지 한계를 극복하기 위해 다중 벡터 문서 임베딩 방법론을 새롭게 제안한다. 구체적으로 제안 방법론은 전체 단어가 아닌 핵심 단어만 이용하여 문서를 벡터화하고, 문서가 포함하는 다양한 주제를 분해하여 하나의 문서를 여러 벡터의 집합으로 표현한다. KISS에서 수집한 총 3,147개의 논문에 대한 실험을 통해 복합 문서를 단일 벡터로 표현하는 경우의 벡터 왜곡 현상을 확인하였으며, 복합 문서를 의미적으로 분해하여 다중 벡터로 나타내는 제안 방법론에 의해 이러한 왜곡 현상을 보정하고 각 문서를 더욱 정확하게 임베딩할 수 있음을 확인하였다.

주제어 : 문서 임베딩, 다중 벡터 문서 임베딩, 단어 임베딩, 텍스트 마이닝

논문접수일 : 2019년 6월 26일 논문수정일 : 2019년 9월 16일 게재확정일 : 2019년 9월 19일
원고유형 : 일반논문 교신저자 : 김남규

1. 서론

데이터보다 빅데이터라는 용어가 더욱 빈번하게 사용될 정도로, 일상 생활에서 생성, 유통, 활용되는 데이터의 양은 빠르게 증가하고 있다. 한국 IDC는 2025년이면 한 해에 생성되는 데이터 규모가 현재의 10배에 달하는 163ZB에 이를 것으로 예상한 바 있다. 또한 인터넷, 소셜 네트워크 서비스, IoT 등을 통해 사용자가 하루 동안 수행하는 상호작용의 수는 2015년 기준 평균 85건에서 2025년에는 4,785건으로 급증할 것으로 예상하였다. 이와 같은 상호작용은 주로 텍스트를 매개로 이루어진다는 점에서, 향후 유통되는 텍스트 데이터의 양은 현재에 비해 급증할 것으로 충분히 예상할 수 있다. 이처럼 데이터 생태계에서 텍스트 데이터가 차지하는 비중이 높아짐에 따라, 텍스트 데이터에 대한 체계적 관리 및 다양한 분석을 통해 새로운 지식을 창출하고자 하는 시도도 매우 활발히 이루어지고 있다.

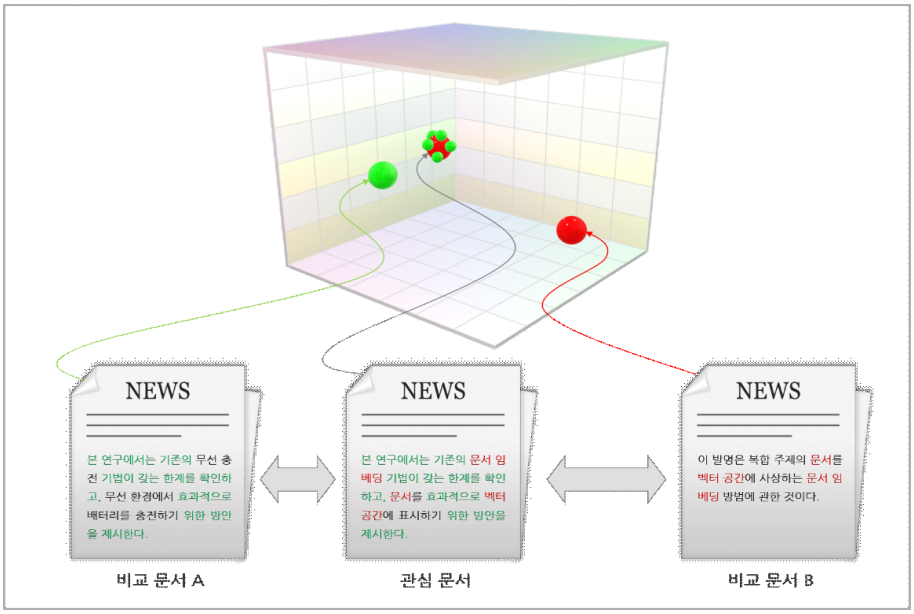
다양한 연산 및 전통적인 분석 기법의 직접 적용이 가능한 정형 데이터와 달리, 모든 비정형 텍스트는 본 분석에 앞서 원본 문서를 컴퓨터가 이해할 수 있는 형태로 변환하는 구조화 작업이 선행되어야 한다. 텍스트 데이터의 구조화를 위

해 임의의 객체를 대수적 성질을 유지하면서 특정 차원의 공간에 사상하는 것을 임베딩(Embedding)이라고 하며, 구체적으로는 단어를 벡터로 나타내는 단어 임베딩(Word Embedding)과 문서를 벡터로 나타내는 문서 임베딩(Document Embedding)으로 실현된다. 단어의 구조화를 위한 가장 고전적인 방식은 원 핫 인코딩(One-hot Encoding) 방식이다. 이는 단어 집합의 크기에 해당하는 차원을 갖는 단어 벡터 공간을 생성하고, 각 단어에 인덱스를 부여한 뒤 해당 단어에 대응되는 인덱스의 차원 값을 1, 그 외의 차원 값을 0으로 설정하는 방식이다. 이러한 방식은 이해가 쉽고 구현도 용이하지만, 단어 수의 증가에 따라 계산 비용도 증가한다는 비효율성과 단어 벡터가 해당 단어의 의미를 충분히 반영하지 못한다는 한계를 갖는다.

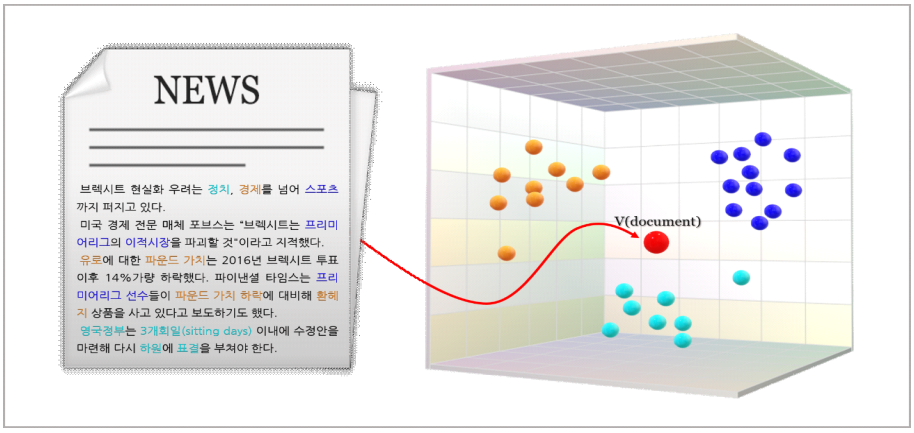
이러한 원 핫 인코딩 방식과 달리 분산 표상(Distributed Representation)(Hinton, 1986) 방식은 대상 데이터를 연속형의 실수 값을 가진 밀집 벡터(Dense Vector)로 표현한다. 이 방식은 계산의 복잡성을 줄여줄 뿐 아니라, 단어 벡터 간 비교를 통한 유사도 분석이 가능하다는 장점을 갖는다. 예를 들어 {PC, NOTEBOOK, RADIO}의 세 단어가 원 핫 인코딩에 의해 각각 PC = (1, 0, 0), NOTEBOOK = (0, 1, 0), 그리고 RADIO = (0, 0, 1) 로 표현되었다고 가정하자. 이 때 PC는 NOTEBOOK과도 다르고 RADIO와도 다름을 알 수는 있지만, PC가 NOTEBOOK과 RADIO 중 어떤 것과 유사한지는 알 수 없다. 이와 달리 위의 세 단어가 분산 표상 방식에 의해 PC = (0.9, 0.6, 0.1), NOTEBOOK = (0.5, 0.8, 0.2), 그리고 RADIO = (0.2, 0.1, 0.9)로 표현되었다면, 벡터 간 거리를 계산하는 다양한 방식에 의해 PC는 RADIO 보다는 NOTEBOOK과 유사함을 알 수 있다. 이러한 장점으로 인해 분산 표상 방식은 텍스트 데이터의 임베딩에 널리 사용되고 있으며, 최근 텍스트 마이닝 분야에서 단어의 의미적 임베딩에 널리 사용되는 word2Vec(Mikolov et al, 2013) 역시 분산 표상 방식에 기반을 두고 있다.

최근에는 개별 단어의 임베딩뿐 아니라 문장, 단락, 그리고 문서 전체를 임베딩하기 위한 시도도 다양한 측면에서 이루어지고 있다. 특히 문서 임베딩에 대한 분석 수요가 급증함에 따라 이를 지원하기 위한 알고리즘이 다수 고안되고 있으며, 이들 중 word2Vec을 확장하여 각 문서를 하나의 벡터로 임베딩하는 doc2Vec(Quoc and Mikolov, 2014)이 가장 널리 사용되고 있다. doc2Vec은 문서와 단어를 동일한 벡터 공간에서 표현한다는 특징으로 인해 활용도가 매우 높다는 장점을 갖는다. 하지만 doc2Vec은 문서에 나타난 모든 단어를 임베딩에 사용하기 때문에, 핵심 단어뿐 아니라 주변 단어들에 의해 임베딩 결과가 왜곡되어 나타날 수 있다는 한계를 갖는다(Figure 1). <Figure 1>에서 관심 문서의 핵심 단어는 비교 문서 B와 매우 유사하게 나타나고 있다. 하지만 관심 문서에 포함된 주변 단어, 즉 내용에 크게 영향을 미치지 않는 일반적인 단어들은 비교 문서 A와 상당 부분 일치하고 있다. doc2Vec에 의한 문서 임베딩에서 각 문서는 핵심 단어뿐 아니라 주변 단어까지 고려하여 임베딩이 이루어지게 되므로, 주변 단어가 다수 존재하는 경우 <Figure 1>과 같이 관심 문서는 비교 문서 B보다 비교 문서 A와 더욱 유사한 것으로 임베딩될 수 있다.

기존 문서 임베딩의 또 다른 한계는 하나의 문서가 하나의 벡터로 임베딩된다는 본질적 특징에 기인한다. 이러한 특징으로 야기되는 한계는 <Figure 2>를 통해 설명될 수 있다. <Figure 2>의



〈Figure 1〉 Limitations of Traditional Document Embedding: Effect of Non-Core Terms



〈Figure 2〉 Limitations of Traditional Document Embedding: Single Vector Representation

좌측에 나타난 문서는 정치, 경제, 그리고 스포츠 분야를 아우르는 복합 주제를 다루고 있다. 이상적으로 이러한 문서는 정치 주제의 문서와 유사한 것으로 임베딩되어야 할 뿐 아니라, 경제 및 스포츠 주제의 문서와도 유사한 것으로 임베딩되어야 한다. 하지만 doc2Vec에 의한 임베딩은 이들 주제를 각각 구분하여 다루지 않고 전체 문서를 통합하여 하나의 벡터로 표현하기 때문

에, 복합 주제 문서는 <Figure 2>와 같이 정치, 경제, 그리고 스포츠의 어떤 공간에도 가깝지 않은 애매한 공간에 임베딩되는 경향이 있다.

따라서 본 연구에서는 doc2Vec으로 대표되는 기존의 문서 임베딩 방법이 갖는 위의 두 가지 한계를 극복하기 위해 다중 벡터 문서 임베딩 방법론을 제안하고자 한다. 우선 주변 단어의 영향을 받는 한계를 극복하기 위해 제안 방법론은 문서 전체가 아닌 문서의 키워드에 대한 단어 벡터만을 활용하여 문서 벡터를 도출한다. 또한 복합 주제를 단일 벡터로 표현해 온 기존 기법의 한계를 극복하기 위해, 제안 방법론은 키워드에 대한 군집화를 통해 주제별 키워드 그룹을 생성하고 각 그룹별로 주제 벡터를 생성한다. 즉 제안 방법론은 임의의 문서의 키워드가 N개의 주제로 그룹화되는 경우 해당 문서를 N개의 벡터로 임베딩한다.

본 연구의 이후 구성은 다음과 같다. 우선 2장에서는 텍스트 분석의 전반적인 과정과 텍스트 임베딩에 관한 기존 연구들을 요약하고, 3장에서는 본 연구에서 제안하는 방법론을 간단한 예시와 함께 소개한다. 다음으로 4장에서는 제안 방법론을 통해 실제 논문 데이터를 임베딩한 실험 결과를 제시하고, 마지막 5장에서는 본 연구의 기여 및 한계를 요약한다.

2. 관련 연구

2.1 텍스트 분석

급증하는 텍스트 데이터 분석 수요에 부응하여 학계뿐 아니라 다양한 산업 분야에서 텍스트 마이닝(Text Mining)에 대한 연구와 투자가 활발하게 이루어지고 있다. 텍스트 마이닝은 구조화되지 않은 텍스트 데이터로부터 흥미롭고 중요한 패턴이나 지식을 추출하는 프로세스로 정의되며(Tan, 1999), 넓은 의미에서 데이터 마이닝의 한 영역으로 간주되기도 한다. 구체적으로 텍스트 마이닝은 정보 검색, 정보 추출, 자연어 처리, 데이터 마이닝, 그리고 기계학습 등 텍스트 분석을 위한 다양한 방법론 및 알고리즘을 광범위하게 다루고 있다(Hotho et al, 2005).

텍스트 마이닝은 일반적으로 문서 수집, 형태소 분석, 구조화까지의 단계와 분석 목적에 따라 문서 분류, 군집화, 토픽 모델링 등의 작업을 수행하는 분석 및 활용의 단계로 구분되어 수행된다. 문서 분류에는 주로 의사결정나무, 규칙 기반 분류, SVM(Support Vector Machine), 신경망, 그리고 베이지안 분류와 같은 방법이 사용되어 왔으며(Aggarwal and Zhai, 2012), 최근에는 CNN (Convolutional Neural Network), RNN(Recurrent Neural Network)과 같은 딥러닝 기법을 텍스트 데이터 분류에 활용하는 연구가 활발히 이루어지고 있다(Kim, 2014; Lai et al, 2015; Liu et al, 2017). 텍스트 데이터에 대한 군집 분석의 경우 정형 데이터에 대한 분석과 마찬가지로 K-평균(K-means) 알고리즘이 널리 사용되며(Kim et al, 2017), 연구 목적에 따라 주제별 군집화를 위해 대표적인 토픽 모델링 기법인 LSA(Latent Semantic Analysis) 또는 LDA(Latent Dirichlet Allocation) 기법이 사용되기도 한다(Yu et al, 2019).

최근까지 텍스트 마이닝 관련 연구들은 구조화 이후의 단계, 즉 문서 분류, 군집화, 토픽 모델링 등을 적용한 활용에 초점을 맞추고 진행되어 왔다. 하지만 텍스트 구조화 과정이 분석 결과의 품질을 실질적으로 좌우한다는 발견에 따라, 최근에는 용어 사전 및 불용어 사전의 구축 과정,

가중 빈도 산출 방식, 그리고 차원 축소 기법 등에 대한 중요성이 더욱 강조되고 있다. 특히 텍스트 데이터를 벡터로 표현하는 과정에서 단어 및 문서가 갖는 의미를 최대한 보존하여 분석 결과의 품질을 향상시키 위한 다양한 임베딩 방법들이 활발하게 연구되고 있다.

2.2 텍스트 임베딩

비정형 데이터인 텍스트에 대해 여러 분석 기법들을 적용하기 위해, 모든 비정형 텍스트는 본 분석에 앞서 원본 문서를 컴퓨터가 이해할 수 있는 형태로 변환하는 구조화 작업을 거쳐야 한다. 문서의 구조화를 위해 고안된 다양한 방법 중 가장 대표적인 방법으로는 벡터 공간 모델(Vector Space Model) (Salton et al, 1975)을 들 수 있다. 벡터 공간 모델에서 각 문서는 하나의 벡터로 표현되며, 차원의 수는 모든 문서에 쓰인 단어 집합의 크기와 같다. 각 문서와 단어가 교차하는 셀은 해당 문서와 단어에 대응되는 가중 빈도를 값으로 갖게 되며, 가중 빈도로는 일반적으로 TF-IDF(Term Frequency-Inverse Document Frequency) 값이 사용된다. 이러한 방식을 통해 구조화된 문서 벡터 간 코사인 유사도 계산을 통해, 공통 단어를 다수 포함한 문장들을 서로 유사한 문장들로 식별하는 작업을 수행할 수 있다. 하지만 벡터 공간 모델은 서로 유사한 의미를 갖지만 상이하게 표현된 두 단어, 즉 이음 동의어 및 유사어를 전혀 반영하지 못한다는 한계를 갖는다. 따라서 최근에는 단순히 단어의 출현 빈도뿐 아니라 단어 자체가 갖는 고유의 의미를 표현하기 위한 다양한 연구들이 수행되었다.

단어의 의미를 표현하기 위해 고안된 다음 방법들의 기본적인 가정은 “같은 맥락에서 쓰인 단어는 비슷한 의미를 가지는 경향이 있다.”는 분포 가설(Distributional Hypothesis) (Firth, 1957)에 기반을 둔다. 분산 표상을 이용한 방식은 상기 가정에 따라 대상 단어의 주변 단어를 학습하는 방식을 통해, 고차원의 벡터를 가진 단어를 저차원의 연속적인 실수 값을 가진 벡터로 표현한다. 이처럼 실수 값으로 표현된 벡터가 분산 표상이며, 단어를 저차원 공간에 사상시키는 과정을 단어 임베딩이라고 한다. 단어 임베딩을 통해 도출된 벡터의 각 차원은 단어의 잠재적인 특징을 나타내며(Turian et al, 2010), 단어의 분산 표상을 학습하는 초기의 대표적인 방법으로는 Neural Network Language Model(NNLM)(Bengio et al, 2003)을 들 수 있다. NNLM은 신경망을 활용한 방법으로, n-1번째의 단어로부터 n번째 단어가 출현할 확률을 극대화시키는 방식을 통해 분산 표상을 학습한다. 이 모델은 이전 모델들에 비해 상대적으로 높은 정확도를 보였으나, 계산 복잡도가 크기 때문에 학습 시간이 매우 길다는 단점을 가지고 있다(Mikolov et al, 2011).

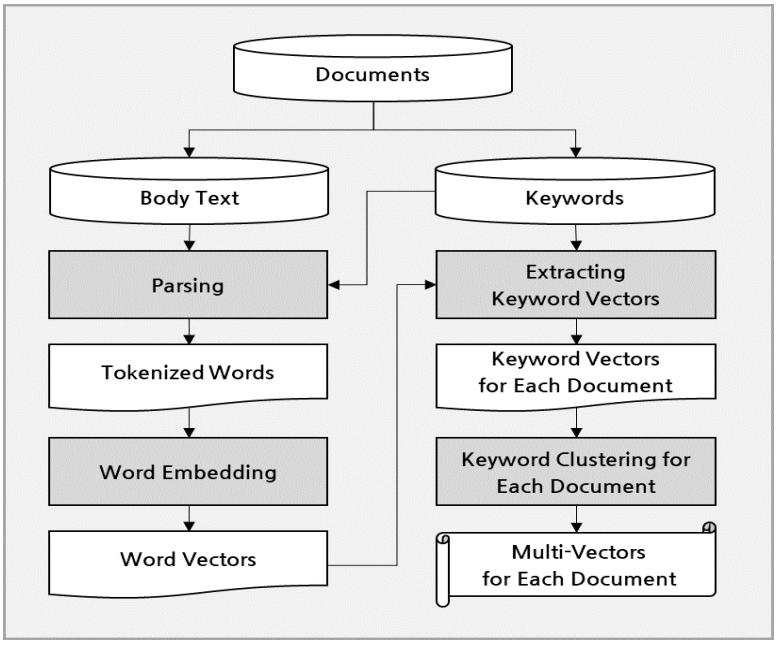
NNLM의 장점을 유지하면서 단점을 보완한 단어 임베딩 방법이 최근 널리 사용되고 있는 word2Vec이다. word2Vec은 자주 출현하는 단어에 대해 Subsampling을 하고, 전체 단어에 대해 Negative Sampling을 수행하는 방식을 통해 계산 복잡성을 비약적으로 줄임으로써 학습 속도를 향상시켰다. word2Vec은 CBOW와 Skip-Gram의 2개의 서로 다른 모델로 구성되어 있으며, 빠른 학습 속도와 높은 정확도로 인해 현재 많은 분야에서 활용되고 있다. 또한 최근에는 word2Vec을 확장하여 문장, 단락, 그리고 문서 단위의 임베딩을 수행하기 위한 연구도 활발하게 진행되고 있다(Quoc and Mikolov, 2014; Kiros et al, 2015; Kenter and Rijke, 2015).

문서 임베딩 기법 중 대표적인 doc2Vec은 word2Vec이 확장된 형태로, 단어 벡터의 학습 과정에서 문서에 대응하는 임의의 벡터를 추가하고 이를 해당 문서 내 사용된 단어 벡터들과 가깝게 위치시키도록 조정하는 과정을 통해 문서를 표현한다. 단어 벡터의 공간이 곧 문서 벡터의 공간이므로 서로 다른 길이의 문서를 하나의 고정된 길이의 저차원 벡터로 표현할 수 있을 뿐 아니라, 문서 벡터와 단어 벡터 간 유사도 비교를 통해 문서 내 단어를 예측할 수 있다는 장점을 갖는다. 하지만 전술한 바와 같이 doc2Vec에 의한 문서 임베딩은 각 문서가 항상 하나의 벡터로 표현될 수 있다는 다소 경직된 가정 하에 수행된다는 한계를 갖는다. 또한 문서 내 모든 용어를 학습에 사용하므로, 문서의 핵심 단어는 상이하지만 주변 단어, 즉 문서의 내용에 크게 영향을 미치지 않는 비 핵심 단어들을 다수 공유하는 두 문서의 벡터가 서로 가깝게 위치하는 경향을 나타낸다. 최근 doc2Vec의 특성을 계승한 채 다른 측면에서의 성능 개선을 시도한 연구는 다수 수행되고 있지만, doc2Vec이 갖는 위의 두 가지 한계를 극복하기 위한 시도는 상대적으로 찾아보기 어렵다.

3. 제안 방법론

3.1 연구 모형

본 장에서는 제안하는 문서 임베딩 방법론, 즉 문서의 핵심 내용을 담고 있는 키워드를 활용하여 하나의 문서를 다중 벡터로 표현하는 방안을 예시와 함께 소개한다. 제안 방법론의 전체 과정은 <Figure 3>와 같다.



〈Figure 3〉 Research Overview

본 방법론은 본문 내용과 키워드가 명시적으로 구분되어 있는 문서를 적용 대상으로 한다. 물론 키워드가 명시적으로 제공되지 않는 문서의 경우 다양한 분석 방법을 통해 본문의 핵심 단어를 도출하여 이를 키워드로 정의한 후 본 방법론을 적용할 수 있다. 하지만 이는 제안 방법론에서 핵심적으로 다루는 내용이 아니므로, 본 절에서는 본문과 키워드가 명시적으로 구분된 문서에 대해 제안 방법론을 적용하는 과정을 소개한다.

제안 방법론은 (1) 파싱(Parsing), (2) 단어 임베딩, (3) 키워드 벡터 도출, (4) 키워드 군집화, 그리고 (5) 다중 벡터 생성의 주요 모듈로 구성된다. 우선 파싱 단계에서는 대상 문서에 대한 형태소 분석을 통해 모든 용어들을 토큰(Token)으로 분리한다. 다음으로 단어 임베딩을 통해 각 토큰을 N차원의 실수 값을 가진 벡터로 변환한 후, 이들 벡터 중 각 문서별 키워드로 지정된 토큰의 단어 벡터만을 추출하여 문서별 키워드 벡터 집합을 구성한다. 다음으로 문서에 포함된 복합 주제를 식별하기 위해 각 문서별 키워드 집합에 대해 군집 분석을 수행하고, 마지막으로 각 군집을 구성하는 키워드들의 벡터로부터 군집별 벡터를 도출한다. 각 단계별 구체적 설명은 본 장의 이후 절에서 가상의 예를 통해 소개한다.

3.2 단어의 벡터화

본 절에서는 <Figure 3>의 (1) 파싱 및 (2) 단어 임베딩에 해당하는 과정, 즉 문서의 모든 용어를 토큰으로 분리하고 이들을 벡터화하는 과정을 다룬다. 전술한 바와 같이 분석 대상 문서는 본문과 키워드가 명시적으로 구분되어 있는 것으로 가정한다. 우선 파싱을 통해 대상 문서를 구성하는 모든 용어를 토큰 단위로 분리하는데, 여기서 토큰이란 의미를 가진 최소 단위인 형태소를 의미한다. 일반적인 텍스트 분석의 경우 형태소 분석 결과로 도출된 토큰을 그대로 사용하여 구조화를 수행하지만, 제안 방법론에서는 다음의 이유로 형태소 분석 과정에서 각 문서별 키워드 집합을 참조한다.

제안 방법론의 핵심 중 하나는 문서 내 단어 중 키워드에 해당하는 단어를 벡터로 표현하는 것인데, 이러한 키워드에는 단순 명사뿐 아니라 복합 명사도 다수 포함되어 있다. 하지만 일반적으로 사용되는 한국어 형태소 분석기는 일부 관용어나 고유 명사를 제외한 복합 명사는 토큰으로 분리해내지 못한다는 한계가 있다. 예를 들어 문서의 키워드에는 “문서 임베딩” 이라는 어휘가 포함되어 있지만, 본문에 대한 형태소 분석을 통해 획득한 토큰 집합에는 “문서”와 “임베딩”이 존재할 뿐 “문서 임베딩”이라는 어휘는 존재하지 않는다. 보다 구체적인 예는 <Figure 4>에서 확인할 수 있다.

<Figure 4(a)>는 각 문서별 키워드 집합을 나타내며, <Figure 4(b)>는 키워드 집합을 고려하지 않은 일반 형태소 분석 결과를 나타낸다. 한편 <Figure 4(c)>는 키워드 집합에 포함된 복합 명사를 고려한 형태소 분석 결과를 나타낸다. <Figure 4(c)>에 나타난 “관절 안정화”, “장거리 달리기”, 그리고 “인터벌 트레이닝” 등의 어휘는 일반적으로 형태소 분석에 사용되는 범용 사전에는 수록되어 있지 않은 복합 명사이다. 따라서 이러한 어휘를 토큰으로 관리하기 위해서는 형태소 분석 단계에서 이들 어휘를 사전에 추가할 필요가 있다. 예를 들어 Python 패키지 중 한글 형태소 분석에 가장 널리 사용되는 Komoran 형태소 분석기는 토큰을 분리하는 과정에서 사용자가 지

정한 단어를 지정한 품사로 태깅(Tagging)하여 분리할 수 있는 기능을 지원하며, 실제로 본 연구의 실험에서도 이를 활용하여 키워드에 사용된 복합 명사를 토큰으로 식별하였다.

이렇게 식별된 모든 토큰들은 단어 임베딩을 통해 실수 값을 가진 벡터로 구조화된다. 본 연구에서는 현재 사용되고 있는 단어 임베딩 방법 중 단어의 특질(Features)을 가장 잘 반영하는 것으로 알려진 word2Vec 알고리즘을 통해 토큰을 벡터로 변환한다. 앞에서 도출한 토큰을 N차원 벡터로 변환한 예는 <Figure 5>에서 확인할 수 있다.

Doc.	Keywords
1	관절 안정화, 균형감각, 반사신경, 생체역학, 어플리케이션, 자이로센서
2	공간정보, 네비게이션, 마라톤, 스마트폰, 장거리 달리기
3	대사증후군, 비만, 사이클, 유전자, 인터벌 트레이닝
…	…

(a) List of Keywords

Doc.	Token Set
1	어플리케이션, 활용, 관절, 안정화, 운동, 개발, 검증, …
2	공간정보, 장거리, 달리기, 네비게이션, 트레이닝, …
3	사이클, 유전자, 인터벌, 트레이닝, 질병, 운동, …
…	…

(b) Initial Tokenization

Doc.	Token Set with Keywords
1	어플리케이션, 활용, 관절 안정화, 운동, 개발, 검증, …
2	공간정보, 장거리 달리기, 네비게이션, 트레이닝, …
3	사이클, 유전자, 인터벌 트레이닝, 질병, 운동, …
…	…

(c) Token Set with Keywords

〈Figure 4〉 Tokenization with Keywords

Doc.	Token Set with Keywords
1	어플리케이션, 활용, 관절 안정화, 운동, 개발, 검증, …
2	공간정보, 장거리 달리기, 네비게이션, 트레이닝, …
3	사이클, 유전자, 인터벌 트레이닝, 질병, 운동, …
…	…

(a) Token Set with Keywords

Word	D_1	D_2	…	D_N
어플리케이션	0.0171	-0.0746	…	0.0137
활용	-0.0154	-0.0026		-0.0054
관절 안정화	-0.0127	0.0216	…	0.0700
운동	0.0520	0.0724	…	-0.0414
개발	0.0896	0.0594		-0.0470
검증	0.1338	0.0131		-0.0898
…		…		

(b) Word Vector of Each Token

〈Figure 5〉 Word Embedding with word2Vec

3.3 다중 벡터 임베딩

본 절에서는 벡터화된 용어들을 활용하여 문서를 다중 벡터로 임베딩하는 과정, 즉 <Figure 3>의 (3) 키워드 벡터 도출, (4) 키워드 군집화, 그리고 (5) 다중 벡터 생성에 해당하는 과정을 다룬다. 기존의 문서 임베딩이 경우에 따라 핵심 단어보다 오히려 주변 단어의 영향을 크게 받을 수 있다는 한계를 극복하기 위해, 제안 방법론은 전체 단어가 아닌 키워드 단어의 벡터만을 사용하여 문서를 임베딩한다. <Figure 5(b)>에 나타난 토큰 벡터의 경우 키워드뿐 아니라 주변 단어의 벡터도 함께 포함하고 있다. 키워드 벡터 도출 과정은 이들 토큰 벡터 중 각 문서별 키워드로 명시된 어휘의 벡터만을 추출하는 과정이며, 그 예가 <Figure 6>에 제시되어 있다. <Figure 5(b)>의 “활용”, “운동”, “개발”, 그리고 “검증” 등의 토큰은 <Figure 6(a)>의 키워드 집합에 포함되어 있지 않기 때문에 <Figure 6(b)>의 키워드 벡터 집합에는 나타나지 않음을 확인할 수 있다. 제안 방법론의 이후 과정에서는 <Figure 6(b)>에 나타난 키워드 벡터만을 문서 임베딩에 활용한다.

기존 문서 임베딩 방법의 또 다른 한계는 하나의 문서를 항상 하나의 벡터로 표현한다는 점에서 찾을 수 있다. 따라서 다양한 주제를 복합적으로 포함하고 있는 문서의 경우, 임베딩의 결과로 나타난 벡터가 이들 주제 각각을 정확하게 나타낼 것을 기대하기란 매우 어렵다. 예를 들어 <Figure 6>에 나타난 문서 Doc_1는 “IT” 주제의 키워드인 “어플리케이션”과 “자이로센서”를 포함하고 있으며, 이와 동시에 “Medical” 주제의 키워드인 “관절 안정화, “균형감각”, “반사신경”, 그리고 “생체역학”을 포함하고 있음을 알 수 있다. 하지만 이들 키워드 전체를 아우르는 문서를 하나의 벡터로 표현하는 경우, 예를 들어 전체 키워드 벡터의 평균을 문서 벡터로 사용하는 경

Doc.	Keywords
1	관절 안정화, 균형감각, 반사신경, 생체역학, 어플리케이션, 자이로센서
2	공간정보, 네비게이션, 마라톤, 스마트폰, 장거리 달리기
3	대사증후군, 비만, 사이클, 유전자, 인터벌 트레이닝
…	…

(a) List of Keywords

Doc.	Keywords	D_1	D_2	…	D_N
Doc_1	관절 안정화	-0.0127	0.0216	…	0.0700
	균형감각	-0.0216	0.0243	…	0.0570
	반사신경	-0.0116	0.0290	…	0.0600
	생체역학	-0.0250	0.0152	…	0.0758
	어플리케이션	0.0171	-0.0746	…	0.0137
	자이로센서	0.0187	-0.0658	…	0.0216
Doc_2	공간정보	0.0253	-0.0879	…	0.0265
	네비게이션	0.0123	-0.0878	…	0.0152
	마라톤	0.1005	0.0230	…	-0.0360
	스마트폰	0.0093	-0.0641	…	0.0166
	장거리 달리기	0.0930	0.0150	…	-0.0281
Doc_3	대사증후군	-0.0061	0.0135	…	0.0627
	비만	-0.0185	0.0078	…	0.0758
	사이클	0.1047	0.0090	…	-0.0381
	유전자	-0.0176	0.0193	…	0.0621
	인터벌 트레이닝	0.1134	0.0194	…	-0.0470
…	…	…			

(b) Vector of Keywords

〈Figure 6〉 Set of Keyword Vectors for each Document

우, 이 벡터는 “IT” 주제와도 거리가 멀고 “Medical” 주제와도 거리가 먼 곳에 사상된다. 이러한 현상은 다른 문서들에서도 유사하게 나타나며, 그 결과 대부분의 문서는 각자 포함하고 있는 세부 주제의 상이함에도 불구하고 서로 유사한 공간에 사상되는 경향을 갖는다.

따라서 본 연구에서는 이와 같이 복합 문서가 하나의 벡터로 표현되는 한계를 극복하기 위하여, 각 문서를 구성하고 있는 주제의 수에 따라 복수개의 벡터로 표현한다. 예를 들어 Doc_1의 경우 키워드 집합이 두 개의 그룹으로 분리된다면 제안 방법론은 해당 문서를 두 개의 멤버 벡터(Member Vector)로 표현할 수 있다. 키워드 벡터를 그룹화하는 과정은 일반적인 군집화 알고리즘을 통해 수행되므로 이 과정에 대한 자세한 설명은 생략하며, 제안 방법론은 가장 대표적인 군집화 알고리즘 중 하나인 K-means 기법을 이용한다.

<Figure 6(b)>의 키워드 벡터에 대해 문서별 군집화를 수행하여 각 문서를 구성하고 있는 키워드 집합을 각각 두 개의 그룹으로 분리하고, 각 문서의 멤버 벡터를 도출한 예가 <Table 1>에

〈Table 1〉 Multi-Vectors for Each Document

Doc.	Subject	Keywords	D_1	D_2	…	D_N
Doc_1	IT	어플리케이션	0.0171	-0.0746	…	0.0137
		자이로센서	0.0187	-0.0658	…	0.0216
		IT 평균	0.0179	-0.0702	…	0.0177
	Medical	균형감각	-0.0216	0.0243	…	0.0570
		관절 안정화	-0.0127	0.0216	…	0.0700
		반사신경	-0.0116	0.0290	…	0.0600
		생체역학	-0.0250	0.0152	…	0.0758
		Medical 평균	-0.0177	0.0225	…	0.0657
	Doc_1 전체 평균		**-0.0058**	**-0.0084**	**…**	**0.0497**
Doc_2	IT	공간정보	0.0253	-0.0879	…	0.0265
		네비게이션	0.0123	-0.0878	…	0.0152
		스마트폰	0.0093	-0.0641	…	0.0166
		IT 평균	0.0156	-0.0799	…	0.0194
	Sports	마라톤	0.1005	0.0230	…	-0.0360
		장거리 달리기	0.0930	0.0150	…	-0.0281
		Sports 평균	0.0968	0.0190	…	-0.0320
	Doc_2 전체 평균		**0.0481**	**-0.0404**	**…**	**-0.0012**
Doc_3	Medical	대사증후군	-0.0061	0.0135	…	0.0627
		비만	-0.0185	0.0078	…	0.0758
		유전자	-0.0176	0.0193	…	0.0621
		Medical 평균	-0.0141	0.0135	…	0.0669
	Sports	사이클	0.1047	0.0090	…	-0.0381
		인터벌 트레이닝	0.1134	0.0194	…	-0.0470
		Sports 평균	0.1091	0.0142	…	-0.0426
	Doc_3 전체 평균		**0.0352**	**0.0138**	**…**	**0.0231**

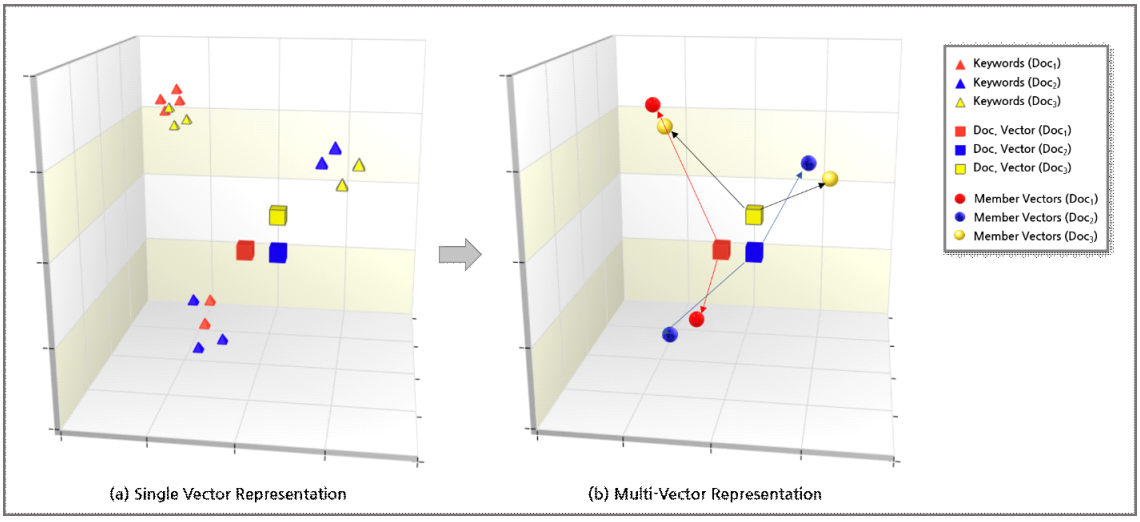
나타나있다. 단 <Table 1>에서 “IT”, “Medical”, 그리고 “Sports” 등의 주제명은 설명의 편의를 위해 임의로 삽입한 것이다. 예를 들어 Doc_1을 단일 벡터로 표현한다면 Doc_1을 구성하고 있는 키워드의 전체 평균을 사용하여 Doc_1 = (-0.0058, -0.0084, …, 0.0497)로 표현될 것이다. 하지만 제안 방법론은 해당 문서를 두 개의 멤버 벡터를 이용하여 Doc_1^1 = (0.0179, -0.0702, …, 0.0177)과 Doc_1^2 = (-0.0177, 0.0225, …, 0.0657)로 표현한다. <Table 2>는 <Table 1>의 세 문서를 단일 벡터로 표현하는 경우와 멀티 벡터로 표현하는 경우를 비교하고 있다.

<Table 2>에서 단일 벡터로는 명시적으로 확인되지 않는 세 문서의 관계가 다중 벡터 표현으로는 상세히 확인됨을 알 수 있다. <Figure 7>은 이를 직관적으로 확인하기 위해 <Table 2>의 벡터를 (D_1, D_2, D_N)의 3차원 공간에 시각화한 결과이다.

<Figure 7>에서 각 문서는 사각형으로, 각 문서에 포함된 키워드는 삼각형으로 표시되었다. <Figure 7(a)>는 각 문서를 하나의 벡터로 표현한 예로, 세 개의 문서는 각기 포함하고 있는 주제가 상이함에도 불구하고 서로 인접한 공간에 사상되었다. 한편 <Figure 7(b)>는 각 문서를 구성하고 있는 멤버 벡터를 원으로 함께 표현한 예이다. 예를 들어 Doc_1은 단일 벡터 표현에서 전체 공간의 중간 영역에 하나의 벡터로 나타나지만, 다중 벡터 표현으로는 좌측 상단의 Doc_1^1 과 하단의 Doc_1^2 의 두 가지 벡터로 나타남을 알 수 있다. 이러한 다중 벡터 표현을 통해 단일 벡터 표현에서는 명확하게 확인할 수 없었던 세 문서간 관계인 Doc_1^1 과 Doc_3^1 의 유사성, Doc_1^2 와 Doc_2^1 의 유사성, 그리고 Doc_2^2 와 Doc_3^2 의 유사성을 확인할 수 있다.

〈Table 2〉 Comparison of Single Vector and Multi-Vector Representation

Doc.	Representation		D_1	D_2	…	D_N
Doc_1	Single Vector		-0.0058	-0.0084	…	0.0497
	Multi-Vectors	Member Vector for IT	0.0179	-0.0702	…	0.0177
		Member Vector for Medical	-0.0177	0.0225	…	0.0657
Doc_2	Single Vector		0.0481	-0.0404	…	-0.0012
	Multi-Vectors	Member Vector for IT	0.0156	-0.0799	…	0.0194
		Member Vector for Sports	0.0968	0.0190	…	-0.0320
Doc_3	Single Vector		0.0352	0.0138	…	0.0231
	Multi-Vectors	Member Vector for Medical	-0.0141	0.0135	…	0.0669
		Member Vector for Sports	0.1091	0.0142	…	-0.0426



〈Figure 7〉 Visualized Comparison of Single Vector and Multi-Vector Representation

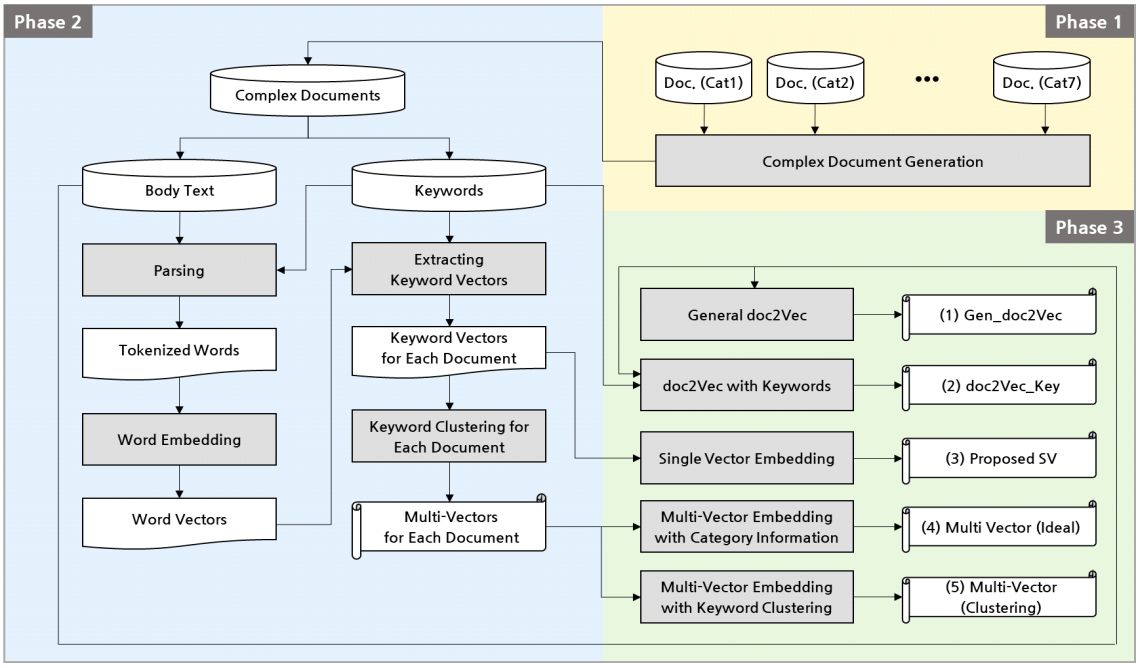
4. 실험

4.1 실험 개요

본 장에서는 제안 방법론의 유용성을 평가하기 위한 실험 수행 과정 및 결과를 요약한다. 본 연구에서는 문서를 다중 벡터로 나타내는 새로운 임베딩 방안을 제시하였다. 하지만 다양한 임베딩 방법 중 어떤 방법에 의해 도출된 벡터가 원본 문서를 더 정확하게 나타내는지 평가할 수 있는 직접적인 기준은 존재하지 않는다. 따라서 본 장에서는 임베딩 결과의 활용 측면에서 다양한 임베딩 방법론을 통해 도출된 문서 벡터의 품질을 간접적으로 평가한다. 구체적으로는 카테고리가 식별되어 있는 문서들에 대해 다양한 방식으로 문서 임베딩을 수행하고, 그 결과 각 문서와 유사한 것으로 판단되는 문서들을 식별한다. 이렇게 식별된 유사 문서들이 기준 문서와 동일한 카테고리에 속하는 경우 임베딩이 정확하게 이루어진 것으로 판단하고, 기준 문서와 상이한 카테고리에 속하는 문서를 유사 문서로 판단한 경우 임베딩이 부정확하게 이루어진 것으로 판단하고자 한다.

따라서 본 연구의 실험 데이터는 각 문서별 키워드 목록과 함께 각 문서의 소속 카테고리가 명시되어 있어야 한다. 본 실험에서는 이러한 조건을 만족하는 데이터로 한국학술정보(KISS) 사이트에서 총 7개에 주제에 대해 3,147개의 논문을 수집하였다. 전반적인 실험은 Python 3.6을 이용하여 진행하였으며, 토큰 분리 작업에는 Komoran, word2Vec 모델링에는 Gensim, 벡터 연산에는 Numpy 패키지를 주로 사용하였다. 본 절에서는 실험의 전체 개요를 소개하고, 본 장의 이후 절에서는 실험의 주요 과정 및 결과를 소개한다. 우선 전체 실험 개요는 <Figure 8>와 같다.

<Figure 8>의 전체 실험 과정은 총 세 단계로 구분된다. 우선 Phase 1은 실험을 위한 복합 문서(Complex Document)를 생성하는 단계로, 복합



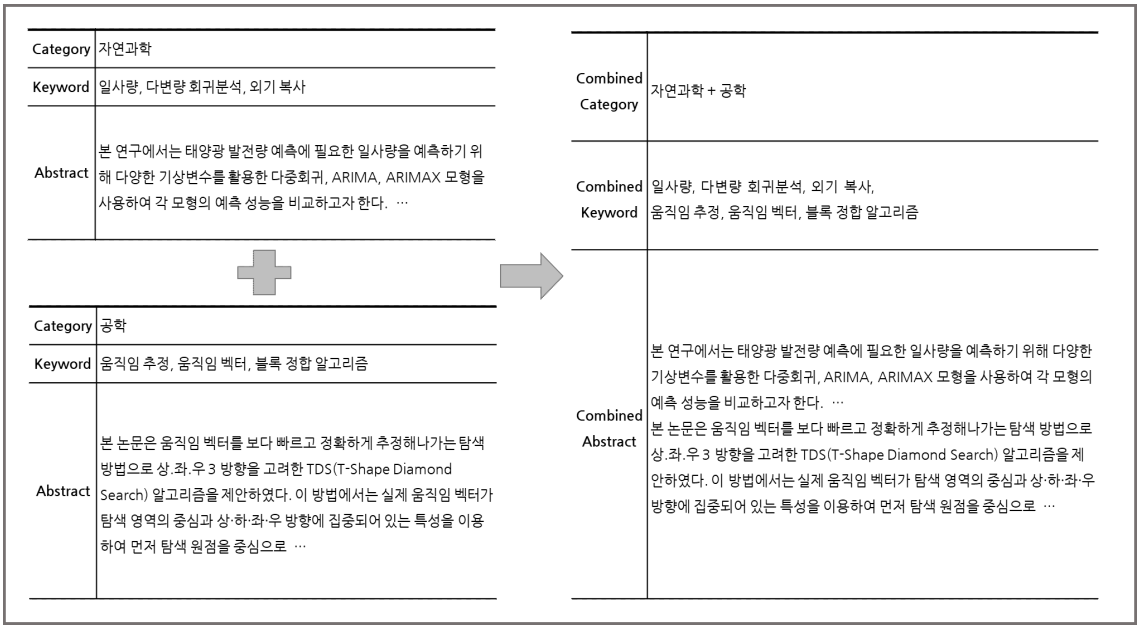
〈Figure 8〉 Overall Process of Experiment

문서 생성의 필요성 및 과정은 다음 절인 4.2절에서 다룬다. 다음으로 Phase 2는 제안 방법론에 의해 다중 벡터를 생성하는 과정으로, 이 부분은 앞 장인 3장에서 자세히 소개한 바와 같다. 마지막 단계인 Phase 3에서는 동일한 문서 집합을 여러 방법론에 의해 임베딩한다. 제안 방법론을 포함하여 총 다섯 가지 방법으로 문서 임베딩을 수행하며, 이에 대한 자세한 내용은 4.3절에서 다룬다. 또한 이와 같이 다섯 가지 방법으로 수행된 문서 임베딩에 대한 성능 평가 결과는 본 장의 마지막 절인 4.4절에서 소개한다.

4.2 복합 문서 생성

제안 방법론은 다양한 주제를 포함하고 있는 복합 문서를 하나의 벡터로 나타내는 기존 문서 임베딩 방식에 대한 한계를 지적하고 있다. 실생활에서 유통되는 대부분의 문서들은 하나의 문서가 여러 주제를 포함하고 있는 복합 문서로 간주될 수 있으며, 제안 방법론의 우수성은 다양한 주제를 포함하고 있는 문서의 임베딩에서 더욱 명확하게 나타날 것으로 예상한다. 이처럼 복합 문서의 표현 과정에서 나타나는 문서 임베딩 방법론의 성능 차이를 보다 명확히 비교하기 위해 본 실험에서는 각 문서의 카테고리 정보를 이용하여 인위적으로 복합 문서를 생성하였으며, 그 구체적인 과정은 다음과 같다.

실험에 사용된 문서는 '인문과학', '사회과학', '자연과학', '공학', '농학', '의약학', 그리고 '예체능'의 7개 카테고리에 속한 3,147개의 논문의 초록이다. 총 7개의 카테고리에 대해 2개의 카



〈Figure 9〉 Generation of Complex Document

테고리 조합(예: '자연과학 + 공학')을 만들 수 있는 경우의 수는 총 21개이며, 이들 각 조합에 대해 50개씩의 복합 문서를 생성한다. 예를 들어 '자연과학'에 속한 논문 하나의 초록과 '공학'에 속한 논문 하나의 초록을 병합하여 '자연과학 + 공학' 분야의 복합 문서를 생성할 수 있다(Figure 9).

복합 문서의 생성에 사용될 문서를 카테고리별로 선정하는 방식은 두 가지로 적용하여 각각의 성능을 실험하였다. 첫 번째 기준은 무작위 추출 방식으로, 각 카테고리마다 각 조합에 참여할 문서를 무작위로 50개씩 선정하였다. 두 번째 방식은 각 카테고리의 중심에 기반을 두어 대표 문서를 선정하는 방식이다. 구체적으로는 각 카테고리에 속한 문서들의 키워드 벡터 평균을 산출하여 이를 카테고리의 중심으로 식별한 뒤, 해당 카테고리의 문서 중 카테고리의 중심에 근접한 상위 50개씩의 문서를 각 카테고리의 대표 문서로 선정하여 복합 문서 생성에 사용하였다.

전술한 방식에 따라 무작위 선정 방식에 의해 복합 문서 1,050개를 생성하고, 중심 기반 방식에 의해 복합 문서 1,050개를 선정하였다. 복합 문서의 구성에 사용된 원본 문서 350개는 단일 문서 집합에서 제외하였다. 그 결과 총 3,847개의 문서(단일 문서 2,797개 + 복합 문서 1,050개)에 대한 학습을 통해 워드 벡터를 학습하였다. 워드 임베딩은 word2Vec 모델링을 사용하였으며, 구체적으로 벡터의 차원은 300차원, 학습 횟수는 50회, 그리고 window size는 5로 지정하였다. 그 결과 총 36,798개의 단어 벡터를 도출하였다.

4.3 문서 벡터 생성

본 절에서는 제안 방법론의 성능의 비교에 사용된 5가지의 문서 임베딩 방식, 즉 (1) Gen_doc2Vec, (2) doc2Vec_Key, (3) Proposed SV, (4) Multi-Vector (Ideal), 그리고 (5) Multi-Vector (Clustering) 방식을 소개한다(Table 3).

<Table 3>에서 (1) Gen_doc2Vec은 일반적인 doc2Vec 모델링을 통한 문서 임베딩 결과를 나타내며, (2) doc2Vec_Key는 일반적인 doc2Vec 모델링을 수행하되 용어 사전으로 키워드 리스트를 사용한 결과를 나타낸다. (3) Proposed SV는 문서에 명시된 키워드들의 워드 벡터 평균으로 문서 벡터를 도출한 결과이다. (1) ~ (3)은 모두 각 문서를 하나의 벡터로 표현한다는 점에서 공통점을 갖는다. 한편 (4)와 (5)는 각 문서를 다중 벡터로 표현하는 방식이다. 이 중 (4) Multi-Vector(Ideal)은 복합 문서를 구성하는 두 원본 문서의 원 카테고리 정보를 활용한다. 예를 들어 '자연과학 + 공학' 분야의 복합 문서의 경우 원 문서 두 개는 각각 '자연과학'과 '공학'의 카테고리에 속한다. 이 때 복합 문서를 구성하는 키워드 집합을 '자연과학' 카테고리의 문서에서 명시된 키워드 부분 집합과 '공학' 카테고리의 문서에서 명시된 키워드 부분 집합으로 구분한다. 이후 각 부분 집합에 속한 단어 벡터들의 평균 벡터를 구하고, 이들 두 개의 평균 벡터를 해당 복합 문서의 멤버 벡터로 사용한다. 하지만 이 방식은 성능 비교를 위한 실험용으로만 수행 가능한 방식으로, 현실 세계의 복합 문서를 구성하는 키워드 집합은 이와 같은 방식으로 분할 가능한 사전 정보를 갖고 있지 않다. 따라서 제안 방법론에서는 키워드에 대한 군집화를 통해 키워드 집합을 부분 집합으로 분할하며, 이러한 방식으로 문서의 멤버 벡터를 도출한 결과가 (5) Multi-Vector(Clustering) 이다. 즉 본 연구에서 제안하는 핵심 방법론은 (5)이며, (4)는 실제로는 적용이 불가능하지만 제안 방법론 성능의 상대적 비교를 위해 소개한 이상적인 모델이다.

4.4 성능 평가

4.4.1 성능 평가 척도

본 부절에서는 앞에서 소개한 5가지 문서 임베딩 방법론의 성능을 평가하는 방법을 소개한다. 구체적으로는 5가지 문서 임베딩을 반복적으로 수행하고, 각 방식에 기반을 두어 복합 문서 각각에 대해 유사도가 가장 높은 문서를 식별한다. 만약 식별된 문서가 기준 문서와 동일

〈Table 3〉 Five Approaches for Document Vector Generation

Method	Base	Used Words	Number of Vectors	External Information
(1) Gen_doc2Vec	doc2Vec	Full Words	Single	Not Required
(2) doc2Vec_Key	doc2Vec	Keywords	Single	Not Required
(3) Proposed SV	Keyword Vectors	Keywords	Single	Not Required
(4) Multi-Vector(Ideal)	Keyword Vectors	Keywords	Multi	Original Category
(5) Multi-Vector(Clustering)	Keyword Vectors	Keywords	Multi	Not Required

한 카테고리에 속하는 경우 임베딩이 정확하게 이루어진 것으로 판단하고, 기준 문서와 상이한 카테고리에 속하는 문서를 유사 문서로 판단한 경우 임베딩이 부정확하게 이루어진 것으로 판단한다.

복합 문서는 두 카테고리의 문서를 병합하여 구성되었기 때문에, 본 실험에서는 각 문서에 대해 유사 문서를 두 개씩 추천하여 정확성을 평가하고자 한다. 다중 벡터인 (4)번과 (5)번의 경우 문서를 구성하는 두 개의 멤버 벡터 각각에 대해 가장 인접한 문서를 추천한다. 따라서 단일 벡터인 (1) ~ (3)번의 경우 유사 문서를 두 개씩 추천하는 다중 벡터 방법론과의 형평성을 유지하기 위해, 각 문서의 벡터와 인접한 문서를 두 개씩 추천한다. 유사 문서의 카테고리 일치 여부는 두 카테고리를 모두 맞춘 경우(Totally Correct), 둘 중의 하나만 맞춘 경우(Partially Correct), 그리고 둘 다 맞추지 못한 경우(Completely Incorrect)의 세 가지로 구분된다. 이 때, 원 문서의 카테고리가 각각 “공학”과 “인문과학”인 경우, 유사 문서로 추천된 두 개의 문서 중 하나라도 “공학”이거나 “인문과학”인 경우와 두 개가 모두 “공학”이거나 모두 “인문과학”인 경우는 Partially Correct로 판정한다. 만약 유사 문서로 추천된 두 개의 문서가 “공학”과 “인문사회” 중 어디에도 속하지 않는 경우 이를 Completely Incorrect로 판정한다. 이러한 척도 하에서는 Totally Correct가 많고 Completely Incorrect가 적을수록 임베딩이 정확하게 이루어진 것이라고 판단할 수 있다.

4.4.2 성능 분석 결과

본 부절에서는 앞에서 소개한 성능 평가 척도에 따라 5가지 문서 임베딩 방법론의 정확성을 비교한 결과를 제시한다(Figure 10).

<Figure 10(a)>는 각 카테고리의 중심에 기반을 두어 카테고리별 대표 문서 50개씩을 선정한

Measures	Gen_doc2Vec	doc2Vec_Key	Proposed SV	Multi-Vector (Ideal)	Multi-Vector (Clustering)
Totally Correct	112	108	92	314	249
Partially Correct	677	660	704	563	626
Completely Incorrect	261	282	254	173	175
Total Documents	1,050	1,050	1,050	1,050	1,050

(a) Performance Evaluation of 5 Methods – Similarity-based Document Composition

Measures	Gen_doc2Vec	doc2Vec_Key	Proposed SV	Multi-Vector (Ideal)	Multi-Vector (Clustering)
Totally Correct	107	106	84	285	261
Partially Correct	648	651	699	602	617
Completely Incorrect	295	293	267	163	172
Total Documents	1,050	1,050	1,050	1,050	1,050

(b) Performance Evaluation of 5 Methods – Random Document Composition

〈Figure 10〉 Performance Comparison of Various Document Embedding Methods

실험 결과이고, <Figure 10(b)>는 각 카테고리별로 대표 문서 50개씩을 임의로 선정한 실험 결과이다. <Figure 11>는 <Figure 10>의 결과를 전체 문서 대비 각 판정에 해당하는 문서 수의 비율로 제시한 것이며, <Figure 12>은 이를 그래프로 도식화한 것이다.

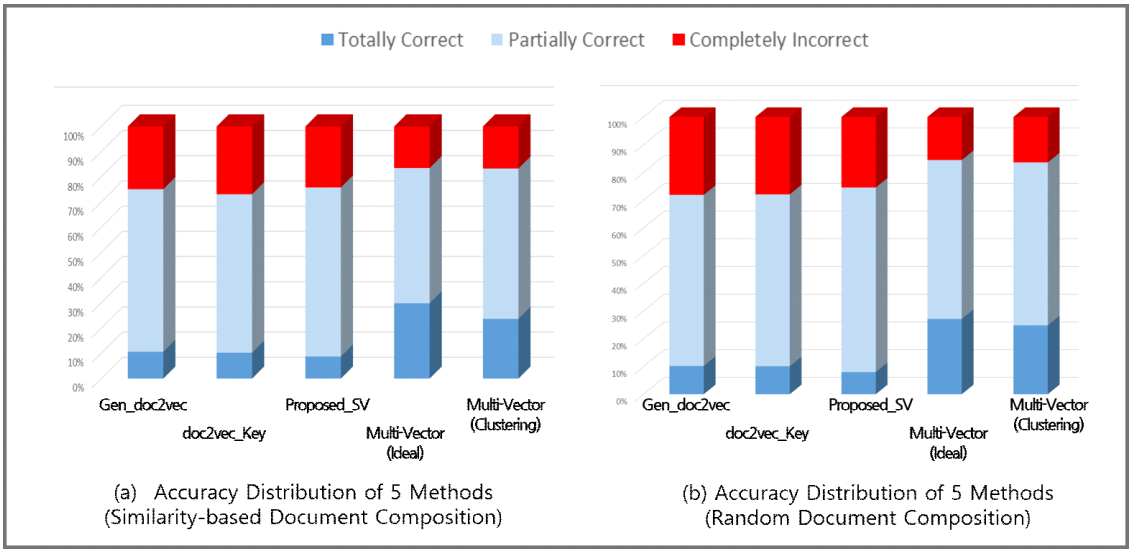
Measures	Gen_doc2Vec	doc2Vec_Key	Proposed SV	Multi-Vector (Ideal)	Multi-Vector (Clustering)
Totally Correct	10.67%	10.29%	8.76%	29.90%	23.71%
Partially Correct	64.48%	62.86%	67.05%	53.62%	59.62%
Completely Incorrect	24.86%	26.86%	24.19%	16.48%	16.67%

(a) Performance Evaluation of 5 Methods – Similarity-based Document Composition

Measures	Gen_doc2Vec	doc2Vec_Key	Proposed SV	Multi-Vector (Ideal)	Multi-Vector (Clustering)
Totally Correct	10.19%	10.10%	8.00%	27.14%	24.86%
Partially Correct	61.71%	62.00%	66.57%	57.33%	58.76%
Completely Incorrect	28.10%	27.90%	25.43%	15.52%	16.38%

(b) Performance Evaluation of 5 Methods – Random Document Composition

〈Figure 11〉 Ratio of Performance Evaluation for Each Document Composition



(a) Accuracy Distribution of 5 Methods (Similarity-based Document Composition)

(b) Accuracy Distribution of 5 Methods (Random Document Composition)

〈Figure 12〉 Accuracy Distribution of 5 Document Embedding Methods

<Figure 12>에서 중심 기반 선정 방식과 무작위 선정 방식의 두 경우 모두 Multi-Vector(Ideal) 방식이 가장 우수한 성능, 즉 Totally Correct가 가장 높고 Completely Incorrect가 가장 낮게 나타나는 결과를 보임을 확인하였다. 또한 제안하는 방식인 Multi-Vector(Clustering)의 경우 원 문서의 카테고리 사전 정보를 사용하지 않았음에도 Multi-Vector(Ideal)와 거의 유사한 정확도를 갖는 것으로 나타났다. 즉 전반적으로 다중 벡터 방식이 단일 벡터 방식에 비해 우수한 성능을 나타낸 것으로 확인되었다. 특히 Totally Correct, 즉 두 개의 카테고리를 모두 맞춘 문서의 비율은 전통적인 doc2Vec를 포함한 세 가지 단일 벡터 방식의 경우 8% ~ 10.67%로 나타난 반면, 본 연구에서 제안하는 Multi-Vector(Clustering)의 경우 이 비율이 23.71% ~ 24.86%로 높게 나타남을 확인할 수 있다. 위의 결과에서 더욱 주목해야 할 부분은 막대 그래프의 최상단에 위치한 Completely Incorrect의 비율이다. 이는 유사 문서로 추천된 두 개의 문서가 원 카테고리 두 곳 중 어느 곳에도 속하지 않는 경우의 비율을 나타낸다. 예를 들어 “인문과학”과 “공학” 카테고리의 문서 두 개를 병합한 복합 문서에 대해 유사 문서 두 개를 추천했는데, 이들 두 문서가 “의약학” 또는 “예체능” 등에 속하는 경우를 나타낸다. 세 가지의 단일 벡터 방식의 경우 이 비율이 24.19% ~ 28.10%으로 나타난 반면, Multi-Vector(Clustering)의 경우 이 비율이 16.38% ~ 16.67%로 낮게 나타났다. 이와 같은 결과는 복합 문서의 의미적 분해를 통해 각 문서를 다중 벡터로 표현함으로써, 문서가 갖는 특질에 따라 각 문서를 더욱 정확하게 임베딩할 수 있음을 보여준다.

5. 결론

텍스트 분석을 다루는 기존의 연구들은 주로 텍스트 구조화 이후의 단계, 즉 분류, 군집화, 토픽 모델링 등 분석 및 활용 단계에 초점을 맞추어 수행되어 왔다. 그러나 최근 텍스트 구조화 작업이 분석 결과의 품질을 실질적으로 좌우한다는 발견에 따라 이 과정에 대한 중요성이 강조되고 있으며, 이에 따라 문서 임베딩에 대한 연구가 활발히 수행되고 있다. 이에 본 연구에서는 doc2Vec으로 대표되는 기존 문서 구조화 방법의 한계를 지적하고, 이를 극복하기 위한 방안을 제시하였다.

구체적으로 제안 방법론은 문서 임베딩을 수행하는 과정에서 문서에 대한 사전 지식, 즉 문서 작성자가 직접 선정한 키워드 정보를 적극 반영함으로써, 문서의 핵심 용어에 집중하여 문서 벡터를 생성하는 방안을 제시하였다. 이와 더불어 일반적으로 다양한 주제를 포함하고 있는 문서를 단 하나의 벡터로 표현하는 기존 문서 임베딩 방법론의 한계를 지적하고, 문서의 의미적 분해를 통해 각 문서를 다중 벡터로 표현함으로써 문서의 특질을 보다 정확히 표현하는 방안을 제시하였다. 실제 문서 3,147건에 대한 실험을 통해 복합 문서의 단일 벡터 표현에서 나타나는 왜곡 현상을 보정하고 각 문서를 더욱 정확하게 임베딩할 수 있음을 확인하였다. 제안 방법론을 통해 더욱 정교한 텍스트 구조화를 수행할 수 있으며, 특히 실무적 관점에서 분류, 군집화, 그리고 토픽 모델링 등 다양한 텍스트 분석 결과의 품질을 향상시키는 효과를 거둘 수 있을 것으로 기대한다.

본 연구는 다음과 같은 측면에서 보완이 필요하다. 본 연구의 실험에서는 제안 방법론의 성능

을 평가하기 위해 복합 문서를 임의로 생성하여 실험에 사용하였다. 이는 문서 임베딩의 품질을 평가하기 위한 새로운 방안을 제시했다는 측면에서 기여로 인정받을 수 있으나, 이러한 복합 문서는 실제로는 존재하지 않기 때문에 이에 대한 결과를 일반화하기에는 다소 무리가 있다. 따라서 향후에는 실제로 유통되고 있는 다양한 유형의 문서들에 대해 제안 방법론의 성능을 평가하기 위한 방안이 마련될 필요가 있다. 또한 본 연구에서는 각 문서의 주제가 두 가지로 구성되어 있다는 다소 경직된 가정 하에 실험을 수행하였다. 하지만 실제 생활에서 유통되는 문서는 해당 도메인 및 문서의 특성에 따라 상이한 수의 주제로 구성되어 있다. 따라서 향후 연구에서는 각 문서를 구성하고 있는 주제의 수가 서로 다를 뿐 아니라 그 수가 미리 알려지지 않은 상황을 가정한 환경에서의 보다 엄밀한 성능 평가가 이루어져야 한다.

참고문헌(References)

Aggarwal, C. C. and C. Zhai, *Mining Text Data*, Springer, Boston, 2012.

Bengio, Y., R. Ducharme, P. Vincent, and C. Janvin, “A Neural Probabilistic Language Model,” *The Journal of Machine Learning Research*, Vol.3, (2003), 1137~1155.

Firth, J. R., “A Synopsis of Linguistic Theory 1930-1955”, *Studies in Linguistic Analysis*, Blackwell, Oxford, 1957.

Hinton, G. E., “Learning Distributed Representations of Concepts,” *Proceedings of the 8th Annual Conference of the Cognitive Science Society*, Vol.1, (1986), 1~12.

Hotho, A., A. Nürnberger, and G. Paaß, “A Brief Survey of Text Mining,” *LDV-Forum,* Vol.20, No.1(2005), 19~62.

Kenter, T. and M. Rijke, “Short Text Similarity with Word Embedding,” *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, (2015), 1411~1420.

Kim, N., D. Lee, H. Choi, and W. X. S. Wong, “Investigations on Techniques and Applications of Text Analytics,” *The Journal of The Korean Institute of Communication Sciences*, Vol.42, No.2(2017), 471~492.

Kim, Y., “Convolutional Neural Networks for Sentence Classification,” *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing(EMNLP)*, (2014), 1746~1751.

Kiros, R., Y. Zhu, R. Salakhutdinov, R. S. Zemel, A. Torralba, R. Urtasun, and S. Fidler, “Skip-Thought Vectors,” *Proceedings of the 28th International Conference on Neural Information Processing Systems*, Vol.2, (2015), 3294~3302.

Lai, S., L. Xu, K. Liu, and J. Zhao, “Recurrent Convolutional Neural Network for Text Classification,” *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, (2015), 2267~2273.

Liu, J., W. Chang, Y. Wu, and Y. Yang, “Deep Learning for Extreme Multi-label Text Classification,” *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, (2017), 115~124.

Mikolov, T., A. Deoras, D. Povey, L. Burget, and J. Černocký, “Strategies for Training Large

Scale Neural Network Language Models," 2011 IEEE Workshop on Automatic Speech Recognition & Understanding, (2011), 196~201.

Mikolov, T., I. Sutskever, K. Chen, G. Corrado and J. Dean, "Distributed representations of words and phrases and their compositionality," *Proceedings of the 26th International Conference on Neural Information Processing Systems*, Vol.2, (2013), 3111~3119.

Quoc, L. and T. Mikolov, "Distributed Representations of Sentences and Documents," *Proceedings of the 31st International Conference on Machine Learning*, Vol.32, (2014), 1188~1196.

Salton, G., A. Wong, and C. S. Yang, "A Vector Space Model for Automatic Indexing," *Communications of the ACM*, Vol.18, No.11(1975), 613~620.

Tan, A., "Text Mining: The State of the Art and the Challenges," *Proceedings of the Pacific Asia Conference on Knowledge Discovery and Data Mining PAKDD'99 workshop on Knowledge Discovery from Advanced Databases*, (1999), 65~70.

Turian, J., L. Ratinov, and Y. Bengio, "Word Representations: A Simple and General Method for Semi-Supervised Learning," *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, (2010), 384~394.

Yu, H., S. Lee, and Y. Ko, "Incremental Clustering and Multi-Document Summarization for Issue Analysis based on Real-time News," *Journal of Korean Institute of Information Scientists and Engineers*, Vol.46, No.4(2019), 355~362.

Abstract

Multi-Vector Document Embedding Using Semantic Decomposition of Complex Documents

Jongin Park* · Namgyu Kim**

According to the rapidly increasing demand for text data analysis, research and investment in text mining are being actively conducted not only in academia but also in various industries. Text mining is generally conducted in two steps. In the first step, the text of the collected document is tokenized and structured to convert the original document into a computer-readable form. In the second step, tasks such as document classification, clustering, and topic modeling are conducted according to the purpose of analysis. Until recently, text mining-related studies have been focused on the application of the second steps, such as document classification, clustering, and topic modeling. However, with the discovery that the text structuring process substantially influences the quality of the analysis results, various embedding methods have actively been studied to improve the quality of analysis results by preserving the meaning of words and documents in the process of representing text data as vectors.

Unlike structured data, which can be directly applied to a variety of operations and traditional analysis techniques, Unstructured text should be preceded by a structuring task that transforms the original document into a form that the computer can understand before analysis. It is called "Embedding" that arbitrary objects are mapped to a specific dimension space while maintaining algebraic properties for structuring the text data. Recently, attempts have been made to embed not only words but also sentences, paragraphs, and entire documents in various aspects. Particularly, with the demand for analysis of document embedding increases rapidly, many algorithms have been developed to support it. Among them, doc2Vec which extends word2Vec and embeds each document into one vector is most widely used.

However, the traditional document embedding method represented by doc2Vec generates a vector for each document using the whole corpus included in the document. This causes a limit that the document vector is affected by not only core words but also miscellaneous words. Additionally, the traditional

* Graduate School of Business IT, Kookmin University
** Corresponding Author: Namgyu Kim
School of Management Information Systems, Kookmin University
77 Jeongneung-ro, Seongbuk-gu, Seoul 136-702, Korea
Tel: +82-2-910-5425, Fax: +82-2-910-4017, E-mail: ngkim@kookmin.ac.kr

document embedding schemes usually map each document into a single corresponding vector. Therefore, it is difficult to represent a complex document with multiple subjects into a single vector accurately using the traditional approach. In this paper, we propose a new multi-vector document embedding method to overcome these limitations of the traditional document embedding methods.

This study targets documents that explicitly separate body content and keywords. In the case of a document without keywords, this method can be applied after extract keywords through various analysis methods. However, since this is not the core subject of the proposed method, we introduce the process of applying the proposed method to documents that predefine keywords in the text.

The proposed method consists of (1) Parsing, (2) Word Embedding, (3) Keyword Vector Extraction, (4) Keyword Clustering, and (5) Multiple-Vector Generation. The specific process is as follows. all text in a document is tokenized and each token is represented as a vector having N-dimensional real value through word embedding. After that, to overcome the limitations of the traditional document embedding method that is affected by not only the core word but also the miscellaneous words, vectors corresponding to the keywords of each document are extracted and make up sets of keyword vector for each document. Next, clustering is conducted on a set of keywords for each document to identify multiple subjects included in the document. Finally, a Multi-vector is generated from vectors of keywords constituting each cluster. The experiments for 3.147 academic papers revealed that the single vector-based traditional approach cannot properly map complex documents because of interference among subjects in each vector. With the proposed multi-vector based method, we ascertained that complex documents can be vectorized more accurately by eliminating the interference among subjects.

Key Words : Document Embedding, Multi-Vector Document Embedding, Word Embedding, Text Mining

Received : June 26, 2019 Revised : September 16, 2019 Accepted : September 19, 2019
Publication Type : Regular Paper Corresponding Author : Namgyu Kim

저 자 소 개

박 종 인
현재 국민대학교 비즈니스IT전문대학원 석사과정에 재학 중이다. 안동대학교에서 학사 학위를 취득하였으며, 한국지능정보시스템학회 학술대회 최우수 논문상, NTIS 정보활용 경진대회 과학기술정보통신부장관상 등을 수상하였다. 주요 관심분야는 텍스트 마이닝, 딥러닝, 데이터 사이언스 등이다.

김 남 규
현재 국민대학교 경영정보학부 교수 및 비즈니스IT전문대학원장으로 재직 중이다. 서울대학교 컴퓨터공학과에서 학사 학위를 취득하고, KAIST 테크노경영대학원에서 Database와 MIS를 전공하여 경영공학 석사 및 박사학위를 취득하였다. 한국지능정보시스템학회 부회장, 한국정보기술응용학회 부회장, 한국경영학회 상임이사, 한국경영정보학회 이사, 한국인터넷정보학회 이사를 역임하였다. 주요 관심분야는 텍스트 마이닝, 데이터 마이닝, 데이터 모델링 등이다.