텐서공간모델과 문맥 기반 임베딩모델의 결합을 통한 자동 문서분류*

권순관 O 김한준

서울시립대학교 전자전기컴퓨터공학부

ghghfhddl@naver.com O khj@uos.ac.kr

Text Classification through Integrating Contextual Embedding Model into Tensor Space Model

SoonKwan Kwon O Han-joon Kim

Department of Electrical and Computer Engineering, University of Seoul

요 약

본 논문은 문맥 기반 임베딩 기법인 ELMo와 3차원 텐서공간모델을 결합하여 문서분류(text classification) 성능을 개선하는 기법을 제안한다. 최근 자연어처리 및 텍스트 마이닝 분야에서 문서의 문맥을 고려한 임베딩 기법들이 다수 제안되고 있다. 이는 동일한 단어라도 주변 문맥을 고려하여 서로 다른 임베딩 벡터를 생성한다. 문맥을 고려한 대표적인 임베딩 모델 중의 하나가 ELMo이다. 본 논문은 본 연구진이 이전 제안한 3차원 텐서공간모델(일명 TextCuboid)을 구성하는데 있어서 ELMo 임베딩 벡터를 활용하며, 이는 풍부한 문맥 정보를 포함하고 있어 문서분류 성능을 높이는데 기여할 수 있다. 최종적으로 본 논문은 기존 3차원 TextCuboid 모델을 확장하여 4차원 구조의 이중 채널 TextCuboid를 제안한다.

1. 서 론

최근 인공신경망 기반 단어 임베딩(word embedding) 기술이 발전하면서 이를 자동 문서분류(text classification) 기법에 활용하는 연구가 활발히 이뤄지고 있다[1,2]. 단어 임베딩이란 문서에서 등장한 단어를 의미를 반영한 밀집 벡터로 표현하는 기법이며, 이는 의미적으로 유사한 단어들의 임베딩 벡터를 상대적으로 가까운 거리에 매핑한다. 특히 ELMo[3]와 같은 문맥 기반 임베딩(contextual embedding) 기법은 동일한 단어라 할지라도 문서 내 단어의 문맥에 따라 서로 다른 임베딩 벡터를 생성한다.

본 논문은 자동 문서분류 성능을 개선할 목적으로, 본 연구진이 이전 제안한 3차원 텐서공간모델(일명 TextCuboid)[4]을 구성하는데 있어서 ELMo 임베딩 벡터를 활용한다. 또한 풍부한 단어의 문맥 정보를 담기 위해, 기존 3차원 TextCuboid 표현모델을 확장하여 4차원 구조의 이중 채널 TextCuboid를 제안한다.

2. 관련 및 배경 연구

2.1 ELMo(Embeddings from Language Model)

초기 단어 임베딩 기법은 Word2Vec[5]과 GloVe[6] 등을 포함한다. 그러나 이 기법들은 다의어나 동음이의어 관계를 가진 단어들을 구별하지 못하는 단점을 가진다. ELMo는 대형 말뭉치(Corpus)를 가지고 양방향 LSTM 모델을 학습함으로써 기존 임베딩 기법의 단점을 극복하였다. 결과적으로 ELMo는 입력 데이터로서 문장을 입력받아 문맥을 반영한 단어의 임베딩 벡터를 생성한다.

2.2 3차원 텐서공간모델 (TextCuboid)

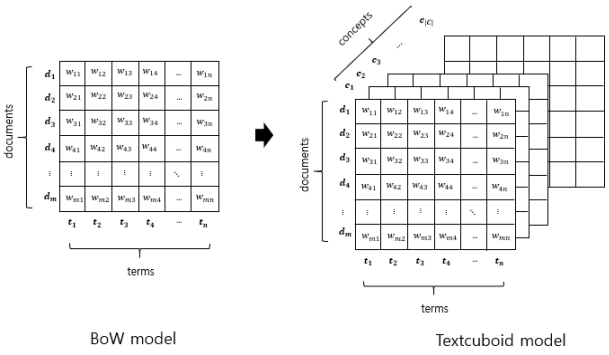

그림 1 BoW 모델과 TextCuboid 모델의 비교

이전 제안한 텐서공간모델인 TextCuboid는 문서 내 출현한 단어가 가지는 의미 정보를 담아내기 위해 하나의 문서를 2차원 Term-Concept 행렬로 표현한다. 비

* 이 논문은 2022년도 정부(교육부)의 재원으로 한국연구재단의 지원(No. NRF-2022R1A2C1011937, 정형 테이블 데이터셋에 대한 딥러닝 기반 데이터 융합 기술 개발)을 받아 수행되었으며, 또한 과학기술정보통신부 및 정보통신기술진흥센터의 대학 ICT 연구센터지원사업의 연구결과로 수행되었음 (IITP-2023-2018-0-01417).

교하여 전통적인 텍스트 표현 모델인 BoW (Bag-of-Word) 혹은 TF-IDF는 하나의 문서 말뭉치를 Term-Document 행렬로 표현하며, 이는 단어의 의미정보를 담지 못한다. 이 문제를 극복하기 위해 제안된 TextCuboid는 위키피디아에서 추출한 개념(concept)을 활용하여 하나의 문서 말뭉치를 3차원 Term-Concept-Document 텐서(tensor)로 표현한다(그림 1 참조).

3. 제안 기법

앞서 서술한 바와 같이, TextCuboid 텍스트 표현모델은 단어의 의미 정보를 담는 개념축을 포함한다. 이전 연구에서는 위키피디아에서 추출한 개념들이 개념공간(concept space)을 구성하는데, 본 논문에서는 ELMo 임베딩 벡터의 각 차원을 TextCuboid의 개념공간의 요소로 활용하였다. 단어(Term)축을 이루는 단어 공간의 구성을 위해서는, 분류대상인 텍스트 데이터셋으로부터 특징선택기법(예: Chi-Square 통계량 기반)을 통해 적정 개수(예: 1,000)의 단어를 추려낸다. 결과적으로 선별된 단어들에 대한 1024차원 ELMo 임베딩 벡터를 결합하여 각 문서당 1000*1024 크기의 Term-by-Concept 행렬을 얻을 수 있다(그림 2 참조).

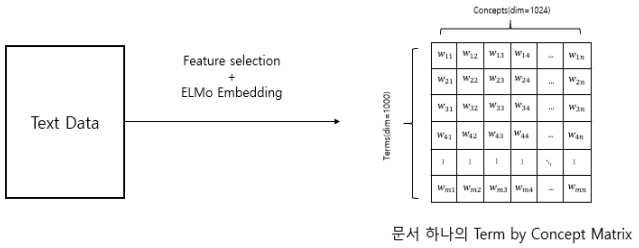

그림 2 TextCuboid 모델에서 하나의 문서에 대한 Term-by-Concept 표현 행렬

또한, 각 문서에 대한 Term-by-Concept 행렬을 하나의 흑백 이미지로 간주할 때, 그 행렬은 하나의 채널에 해당한다. 따라서 단어에 대한 서로 다른 두 가지 ELMo 임베딩 벡터를 가지고 채널 수를 확장할 수 있는 것이다. 다시 말해서, 각 문서는 2개의 Term-by-Concept 행렬로 표현될 수 있는데, 결국 하나의 문서 말뭉치에 대하여 4차원 구조의 TextCuboid를 생성할 수 있다. 우리는 이를 이중 채널 TextCuboid라 칭한다. 실제 문서 분류를 위해서는 이중 채널을 가지는 텐서정보를 학습할 수 있도록 합성곱 신경망을 포함하는 TextCNN[7] 아키텍처를 활용하였다. TextCuboid의 생성 과정은 다음과 같다.

3-1. 단어수준의 ELMo 임베딩 학습

우리는 ELMo 임베딩을 활용한 TextCuboid를 생성하기 위해서 3가지 유형의 ELMo 임베딩 모델을 생성하였다. 기본적으로 Tensorflow-hub에서 제공하는 사전 학습된 ELMo 임베딩 모델을 활용하며, 추가적으로 분류할 텍스트 데이터셋 및 분류할 데이터셋과 유사한 데이터셋으로 학습한 2개의 ELMo 임베딩 모델을 생성하였다.

3-2. 문서 수준의 ELMo 임베딩 행렬 생성

문서 단위로 ELMo 임베딩을 하면 문서가 갖고 있는 각 단어 토큰(token) 수만큼 1024차원의 임베딩 벡터가 출력된다. 예를 들어, 텍스트 데이터셋 내 문서가 100개의 단어 토큰을 가진다면, ELMo 임베딩을 통해 100*1024 크기의 행렬을 얻게 된다. 분류할 텍스트 데이터셋에서 각 문서별로 ELMo 임베딩을 진행했다.

3-3. 이중 채널 TextCuboid 생성

분류 대상 문서로부터 주요 단어를 선별하고, 각 단어에 대한 임베딩 벡터를 개념축에 할당함으로써, 각 문서에 대한 Term-by-Concept 행렬을 생성한다. 또한 서로 다른 데이터셋으로 학습한 ELMo 임베딩 모델을 활용하여, 하나의 문서에 대한 2개의 Term-by-Concept 행렬을 겹쳐서 4차원 구조의 이중 채널(2-channel) TextCuboid를 만들 수 있다 (그림 3 참조).

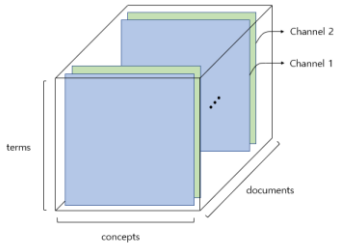

그림 3 이중 채널 TextCuboid 구조

4. 실험 및 성능평가

제안 기법의 성능을 평가하기 위해서, 우리는 문서분류 실험에 자주 사용되는 20Newsgroups과 이와 연관성이 높은 AG News 데이터셋을 사용하였다. 우리는 이 데이터셋을 8:2의 비율로 데이터를 분할하고 각각 학습데이터 및 테스트 데이터로 설정하여 실험을 진행하였다. 여기서 AG News 데이터셋은 분류가 아닌 ELMo 임베딩 학습을 위해서만 사용된다.

표 1 ELMo 임베딩 기반 TextCuboid 구분

TextCuboid 유형
TextCuboid(Hub)
TextCuboid(NG)
TextCuboid(AG)
2-Channel TextCuboid(Hub +NG)
2-Channel TextCuboid(NG+AG)
2-Channel TextCuboid(Hub +AG)

Hub: Tensorflow-hub에서 제공하는 ELMo
NG: 20Newsgroups 데이터셋으로 학습한 ELMo
AG: AG News 데이터셋으로 학습한 ELMo

우리는 표 1의 6가지 TextCuboid에 대해 합성곱 인공신경망을 포함하는 TextCNN을 활용하여 자동분류를 수행하였다. 그림 4의 TextCNN은 합성곱 레이어(convolution layer)와 글로벌 맥스풀링 레이어(global maxpooling layer)를 2개의 완전 연결 레이어(fully-connected layer)를 통해 마지막 20개의 노드를 가지는 출력 레이어와 연결한다. 합성곱 레이어에서의 커널 크기는 (1, 1024)이고 필터 수는 1024, 패딩은 0, 스트라이드는 1로 설정했고 완전 연결 레이어는 순서대로 512, 128개의 뉴런을 가진다. 또한 활성화 함수 관련하여, 모든 레이어에 ReLU 함수가 사용되었고 최적화 방법(optimization)은 Adam, 배치 크기(batch size)는 128로 설정하여 10에포크(epoch) 동안 학습이 수행되었다.

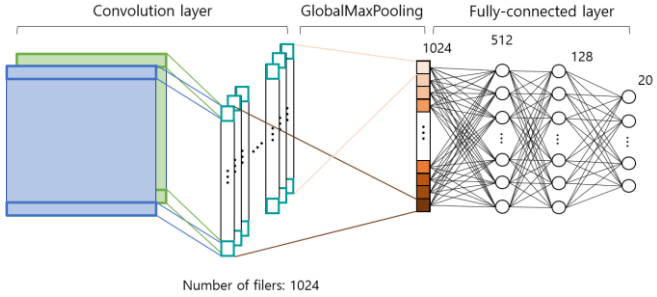

그림 4 이중 채널 Textcuboid의 TextCNN 구조

제안 기법의 효능을 정량적으로 평가하기 위해, 우리는 6개 유형의 TextCuboid를 TextCNN으로 분류한 정확도와 전통적인 머신러닝 모델인 나이브 베이즈(Naïve Bayes), SVM(Support Vector Machine), 로지스틱 회귀(Logistic Regression)로 분류한 정확도를 비교하였다. 공정한 비교를 위해 전통적인 머신러닝 모델과 TextCuboid에서 사용한 단어를 동일하게 설정하였다 (3.3절 참조)

표 2 TextCuboid를 활용한 분류모델과 전통적인 머신러닝 분류모델의 정확도 비교

Model		Accuracy (%)
제안 모델	TextCuboid(Hub) + TextCNN	79.5 %
	TextCuboid(NG) + TextCNN	83.6 %
	TextCuboid(AG) + TextCNN	78.2 %
	2-Channel TextCuboid(Hub +NG) + TextCNN	83.4 %
	2-Channel TextCuboid(NG+AG) + TextCNN	79.2 %
	2-Channel TextCuboid(Hub +AG) + TextCNN	79.3 %
전통 모델	Naïve Bayes	71.5 %
	SVM	71.4 %
	Logistic Regression	73.6 %

표 2 에서 보는 바와 같이, TextCuboid 를 활용한 모델들의 성능이 전통적인 머신러닝 모델들의 성능과 비교해 최소 4.6%, 최대 12.2% 더 높은 정확도를 보였다. 최고의 정확도를 가지는 임베딩 모델은 분류 대상 데이터셋인 20Newsgroup 으로 학습한 모델이다. 이중 채널 Textcuboid 를 사용한 모델들은 단일 채널 TextCuboid 모델들과 비교해 성능 향상이 이뤄지지 않아 이를 극복하기 위한 연구를 계속 진행하고 있다. 이중 채널 TextCuboid 가 개선되기 위해서는 분류대상 데이터셋과 연관성이 높은 데이터셋을 탐색하는 문제가 중요함을 인식하였다.

5. 결론 및 향후 연구

본 논문은 ELMo 임베딩 벡터로 개념 축을 재구성한 3 차원 텐서공간모델(TextCuboid)과 이를 확장한 이중 채널 TextCuboid 모델을 제안하였다. 제안한 모델들은 TextCNN 을 통해 문서분류 실험이 수행되었고, 전통적인 머신러닝 모델과 비교하여 상대적으로 우위를 보였다. 이중 채널 TextCuboid 는 유의미한 성능 향상을 보이지 않았지만, 개념 축을 구성하는 임베딩 벡터의 품질을 높여 성능 개선의 여지를 기대하고 있다. 이와 관련한 향후 연구는 이중 채널을 구성하기 위한 데이터셋의 효과적인 탐색 방안과 3 개 이상의 다중 채널(multi-channel) TextCuboid 를 구성하는 방안을 포함한다.

참 고 문 헌

[1] Liu, Wenbin, et al. "A multi-label text classification model based on ELMo and attention", MATEC Web of Conferences. Vol. 309. EDP Sciences (2020)

[2] Lin, Yuxiao, et al. "Bertgcn: Transductive text classification by combining gcn and bert", ACL-IJCNLP, pp. 1456-1462 (2021)

[3] Peters, Matthew E., et al. "Deep contextualized word representations", NAACL, pp. 2227-2237 (2018)

[4] 김한준, 장재영, "위키피디어 기반 개념 공간을 가지는 시멘틱 텍스트 모델", 한국전자거래 학회지 제 16 권 3 호, pp. 107-123 (2014)

[5] Tomas Mikolov, Kai Chen, et al. "Efficient Estimation of Word Representations in Vector Space", CoRR (2013)

[6] Pennington, J., Socher, R., et al. "Glove: Global vectors for word representation", EMNLP, pp. 1532-1543 (2014)

[7] Kim, Y., "Convolutional neural networks for sentence classification", EMNLP, pp. 1746-1751(2014)