

공학석사 학위논문

위조 리뷰 검출을 위한 텐서 공간
모델 기반 심층 인공 신경망 알고리즘

**Tensor Space Model based Deep Neural Network
Algorithm for Fake Review Detection**

2019년 2월

서울시립대학교 대학원
전자전기컴퓨터공학과
임 푸 림

위조 리뷰 검출을 위한 텐서 공간
모델 기반 심층 인공 신경망 알고리즘

**Tensor Space Model based Deep Neural Network
Algorithm for Fake Review Detection**

지도교수 김 한 준

이 논문을 석사학위 논문으로 제출함

2018년 12월

서울시립대학교 대학원
전자전기컴퓨터공학과
임 푸 림

임푸름의 전자전기컴퓨터공학 석사학위 논문을 인준함.

심사위원장 장 석 진 인

심 사 위 원 이 영 민 인

심 사 위 원 김 한 준 인

2018년 12월

서울시립대학교 대학원

국 문 초 록

소셜 미디어 서비스가 발전하면서 소셜 리뷰 데이터로부터 정보를 획득하는 빈도가 높아지고 있다. 하지만 소셜 리뷰는 익명의 사용자가 작성한 문서이기 때문에 신뢰도가 매우 낮다. 따라서 리뷰 데이터의 위조 여부를 판별하는 기술의 필요성이 점차 증가하고 있다. 본 논문은 위조 리뷰 검출 문제를 해결하기 위하여 텍스트 분류 기법을 이용하였다. 하지만 소셜 리뷰 데이터의 단문, 일상 단어 사용의 특성 때문에 기존의 텍스트 분류 기법 사용 시 검출 성능이 낮다. 이를 해결하기 위하여 문서를 텐서 공간 모델로 변형한 뒤, 이 텐서 공간 모델에 적합한 인공 신경망 아키텍처를 설계함으로써 높은 위조 리뷰 검출 성능을 달성하고자 한다.

주제어: 위조 리뷰 검출, 텍스트 분류, 딥러닝, 텍스트 마이닝

공학석사 학위논문

위조 리뷰 검출을 위한 텐서 공간
모델 기반 심층 인공 신경망 알고리즘

**Tensor Space Model based Deep Neural Network
Algorithm for Fake Review Detection**

2019년 2월

서울시립대학교 대학원
전자전기컴퓨터공학과
임 푸 림

목 차

제 1 장 서론	1
제 1 절 연구의 배경	1
제 2 절 연구 내용 및 범위	3
제 3 절 논문의 구성	4
제 2 장 관련 연구	5
제 1 절 단문 분류 기법	5
제 2 절 시멘틱 텐서 공간 모델 생성 기법	6
제 3 장 위조 리뷰 검출 프로세스	7
제 1 절 전체 프로세스	7
제 2 절 텐서 공간 모델 생성 과정	8
제 4 장 개념 임베딩 생성	13
제 5 장 인공 신경망 아키텍처	16
제 6 장 위조 리뷰 검출 성능 평가	19
제 1 절 실험 데이터 셋	19
제 2 절 실험 기법	23
제 3 절 인공 신경망 파라미터 튜닝	25
제 4 절 성능 평가	27
제 7 장 결 론	30
제 1 절 요약 및 기여내용	30
참 고 문 헌	31
Abstract	33

그림 차례

그림 3.1.1 위조 리뷰 검출 전체 프로세스	7
그림 3.2.1 텐서 공간 모델 개념도	8
그림 3.2.2 Bag-of-Words 표현 모델	8
그림 3.2.3 Word2Vec 표현 모델	9
그림 3.2.4 데이터 셋으로부터 개념 추출 과정	9
그림 3.2.5 개념과 데이터 셋을 이용한 텐서 공간 모델 생성 과정	10
그림 3.2.6 텐서 공간 모델 생성 과정 요약	11
그림 4.1 개념 임베딩 생성 과정	13
그림 4.2 문서 임베딩 생성 과정	14
그림 5.1 심층 신경망 디자인 1 (TSM DNN)	16
그림 5.2 심층 신경망 디자인 2 (Concept-wise TSM DNN)	16
그림 5.3 심층 신경망 디자인 3 (Concept-wise TSM CNN)	17
그림 6.4.1 각 기법 별 위조 리뷰 검출 성능 비교 그래프	28
그림 6.4.2 레이어 수 별 위조 리뷰 검출 정확도 변화	29

표 차 례

표 6.1.1 Yelp 리뷰 데이터 상세 정보	19
표 6.1.2 리뷰 샘플	20
표 6.1.3 위조 리뷰 데이터셋에서 추출한 개념들	21
표 6.4.1 각 기법 별 위조 리뷰 검출 성능 비교	27

제1장 서론

제1절 연구의 배경

최근 소셜 미디어 이용자가 급증하며 이용자들의 리뷰로부터 정보를 획득하는 빈도가 높아지고 있다. 하지만 소셜 데이터는 작성자의 신원이 확실하지 않아서 작성한 리뷰의 신뢰도가 매우 낮다. 예를 들어 의도적으로 악의적인 리뷰를 작성하거나, 보상을 대가로 긍정적인 리뷰를 작성하는 경우가 존재한다. 따라서 실제 리뷰가 아닌 위조된 리뷰를 검출하여 정확한 리뷰를 판별해야 할 필요성이 점차 증가하고 있다.

본 논문은 위조 리뷰 검출 문제를 해결하기 위하여 Text Classification 기술을 이용하였다. 하지만 리뷰 데이터의 경우 단문인 경우가 많고, 일상적인 단어를 사용하기에 기존에 존재하는 Text Classification 방법을 그대로 적용할 시 높은 성능을 기대할 수 없다. 이 문제를 해결하기 위하여 리뷰 데이터를 텐서 공간 모델 [1]로 변형하여 사용하였다. 텍스트 문서 데이터를 기계학습 입력 데이터로 적합한 구조로 변형하는 방법은 Bag-Of-Words[2], Word2Vec[3], Doc2Vec[4] 등이 존재한다. 그럼에도 불구하고 본 논문에서 텐서 공간 모델을 이용한 이유는 이 모델이 위키피디아의 온톨로지를 이용하여 만들기 때문에, 양질의 위키피디아 데이터가 텐서 공간 모델에 반영되기 때문이다.

단순히 텐서 공간 모델을 사용한 것만으로도 위조 리뷰 검출 성능을 높힐 수 있지만 본 논문은 이 모델에 적합한 심층 인공 신경망 아키텍처를 설계하여 더욱 높은 검출 성능을 낼 수 있도록 하였다. 텐서 공간 모델은 개념이라는 요소를 중요한 축으로써 구성된 모델이므로, 이를 고려하여 본 논문은 개념 임베딩이라는 것을 도입하였다. 이를 인공 신경망 아키텍처 내부에 포함하도록 하여 텐서 공간 모델에 매우 적합하도록 설계하였다.

제안 아키텍처의 위조 리뷰 검출 성능을 검증하기 위하여 본 논문은 Yelp 식당

리뷰 사이트에서 실제 리뷰와 위조 리뷰를 수집하여 실험을 하였다. 전통적인 기계 학습 기법 2개와 딥러닝 기법 4가지를 선정하였다. 대조 기법과의 공정한 성능 비교를 위하여 5 Fold cross-validation 결과를 성능 평가 지표로 사용하였다.

제2절 연구 내용 및 범위

본 논문은 위조 리뷰 검출을 위한 심층 인공 신경망 아키텍처를 제안하고자 한다. 리뷰 데이터를 텐서 공간 모델로 변형하여 사용하였는데, 변형 과정에 대한 설명과 이 모델에 적합한 인공 신경망 구조를 제시한다.

리뷰 데이터는 일반적인 텍스트 문서들에 비해 단문이며 일상적인 단어를 사용하기 때문에 기존의 기계학습 기법으로는 높은 성능을 달성할 수 없다. 이를 해결하기 위해 텐서 공간 모델을 도입하였고, 이 데이터 구조를 분석하여 적합한 인공 신경망 아키텍처를 설계한다. 이 과정에서 개념 임베딩과 문서 임베딩을 정의하여 신경망 아키텍처 내부에 포함한 뒤, 학습과정에서 이것들을 동시에 학습하여 위조 리뷰 검출 성능을 높인다.

제안 아키텍처의 성능 평가를 위하여 전통적인 기계학습 기법인 SVM, Naive Bayes에 딥러닝 기법인 DNN, CNN을 이용한다. 추가적으로 2017년도 가장 높은 텍스트 분류 성능을 보인 기법과의 위조 리뷰 검출 성능 비교를 통해 제안 아키텍처가 적절하게 설계되었다는 점을 증명한다.

제3절 논문의 구성

본 논문의 구성은 다음과 같다.

2장에서는 본 논문의 핵심인 위조 리뷰 분류와 유사한 단문 분류를 위한 기법들과 본 논문에서 사용한 텐서 공간 모델 관련 연구를 살펴본다.

3장에서는 두 가지를 설명한다. 첫 번째는 본 논문에서 제안하는 위조 리뷰 검출을 수행하는 전체 프로세스에 대하여 설명하고 두 번째는 본 논문에서 사용한 텐서 공간 모델의 생성 과정에 대하여 자세히 설명한다.

4장에서는 텐서 공간 모델에 적합한 인공 신경망 아키텍처를 설계하기 위하여 도입한 개념 임베딩과 문서 임베딩에 관하여 설명한다.

5장에서는 본 논문에서 제안하는 인공 신경망 아키텍처인 TSM DNN, Concept-wise DNN, Concept-wise CNN를 설명한다.

6장에서는 본 논문이 제안한 인공 신경망 아키텍처의 위조 리뷰 검출 성능을 확인하기 위한 실험에 대하여 설명하고 그 결과를 분석한다.

마지막 7장은 대조 기법에 비해 본 논문의 제안 인공 신경망 아키텍처의 높은 검출 성능을 보이는 원인에 대하여 정리한다.

제2장 관련 연구

제1절 단문 분류 기법

최근 딥러닝 기술이 발전함에 따라 단문 분류를 위해 순환신경망(Recurrent Neural Network)과 컨볼루션 신경망(Convolutional Neural Network)을 적용하는 연구가 활발히 진행되고 있다. 그 예로, Pengfei Liu[5]의 연구에서는 유명한 순환신경망 중 하나인 LSTM(Long Short Term Memory)을 이용하여 3가지 유형의 복잡한 아키텍처를 제안하였다. Ji Young Lee[6]의 연구에서는 순환신경망과 컨볼루션 신경망을 동시에 사용하여 분류 성능을 높이려고 하였다. 또한 Alexis Conneau[7]의 연구는 단어 단위로 데이터를 전처리 하여 입력하던 기존의 방식과는 달리 문자 단위로 입력데이터를 매우 깊은 컨볼루션 신경망을 설계하여 분류 성능을 높이려고 하였다. Siwei Lai[8]의 연구는 순환신경망과 컨볼루션 신경망을 조합하여 RCNN이라는 구조를 설계하였고, 이를 텍스트 분류에 적용함으로써 분류 성능을 높이려고 하였다.

다른 한편으로는 새로운 형태의 신경망을 설계하여 분류 성능을 높이려고 한 연구도 있다. Thang Luong[9]의 연구는 Recursive Neural Network로 정의한 신경망을 설계하여 분류 성능을 높이려고 하였는데, 이 신경망은 벡터 형태로 표현된 단어 2개를 문서 내의 순서대로 입력 받아서 단어 벡터와 동일한 크기의 출력 값과 동시에 스코어를 출력하여 높은 스코어를 출력한 단어 들을 결합 후 스코어링과 동시에 출력한 벡터를 결합한 2개의 단어와 대치한다. 또다시 스코어를 비교하여 대치하는 과정을 모두 대치하여 비교할 스코어가 존재하지 않을 경우까지 수행한다. 이를 통해 문서의 계층 구조와 시멘틱 정보를 학습할 수 있다. 따라서 학습 결과 문서의 문맥에 관련된 정보를 알 수 있게 된다.

2절 시멘틱 텐서 공간 모델 생성 기법

딥러닝 아키텍처를 설계하여 분류 성능을 높임과 동시에 최근에는 문서 데이터를 기계학습에 사용하기 위한 데이터 구조를 전통적인 방법인 TF-IDF가 아닌 Bag-Of-Words, Word2Vec 등으로 변형하여 입력 데이터로 사용하여 분류 성능을 높이고 있다. 본 논문은 특별히 텐서 공간 모델[1]을 이용하여 데이터 구조를 변형하였다. 텐서 공간 모델은 선행연구[10]에서 다른 벡터 공간 데이터 구조와 비교 시 더욱 높은 성능을 보였다. 이 구조는 문서 데이터 셋 내에서 ‘개념’을 추출하고 이를 위키피디아 페이지 정보를 이용하여 문서 내의 TF-IDF정보를 필터링하는 것을 통해 생성한다. 자세한 생성 과정은 3.2절을 통하여 분석한다. 이러한 생성 과정 때문에 문서는 Term by Concept의 매트릭스 구조를 가지게 된다. 문서를 텐서 공간으로 변환하는 것을 통해 단어 정보만을 이용한 기존의 벡터 차원의 다른 데이터 구조들과 달리 문서 내의 의미 정보를 이용할 수 있도록 만든다. 하지만 텐서 공간 데이터로 변형하였기 때문에 기존의 단어 중심으로 해석한 딥러닝 기법들의 입력으로 사용하기에 적합하지 않아서 본 논문은 텐서 공간 모델에 적합한 인공 신경망 아키텍처를 제안한다.

제3장 위조 리뷰 검출 프로세스

제1절 전체 프로세스

본 논문이 제안하는 위조 리뷰 검출을 수행하는 전체 프로세스에 대하여 설명한다. 먼저 실제 리뷰와 위조 리뷰를 1대1 비율로 수집하여 정리한다. 그 뒤, 위조 리뷰 데이터를 텐서 공간 모델 구조로 변형한다. 이 때 수행하는 모든 과정은 3.2절에서 설명한다. 생성한 텐서 공간 모델을 위조 리뷰와 실제 리뷰의 비율을 유지한 채 5개로 분할한다. 마지막 과정으로 5장에서 소개하는 본 논문의 제안 아키텍처를 분할 한 데이터를 이용하여 5 Fold cross-validation을 수행한다. 5 Fold cross-validation을 이용한 이유는 본 논문의 실험에 사용한 리뷰 데이터의 수가 12,000개로 학습 데이터와 테스트 데이터의 분할에 따라 분류기 성능의 변화가 크기 때문에 공정한 정확한 분류기의 성능 비교를 위해 사용하였다.

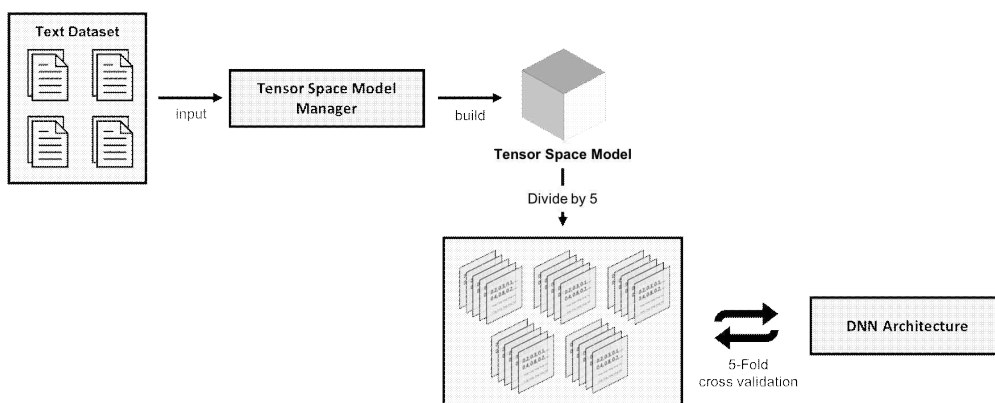


그림 3.1.1 위조 리뷰 검출 전체 프로세스

본 논문은 앞서 설명한 프로세스(그림 3.1.1)를 이용하여 인공 신경망 아키텍처를 학습 하고 평가한다.

제2절 텐서 공간 모델 생성

본 논문은 텐서 공간 모델로 배경 연구[]의 핵심 아이디어를 사용하였다. 이 모델은 텍스트 데이터 셋에서 개념을 추출하여 문서 내의 TF-IDF 정보를 필터링 하는 것을 통하여 만들게 된다.

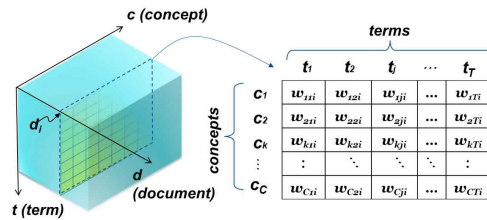


그림 3.2.1 텐서 공간 모델 개념도

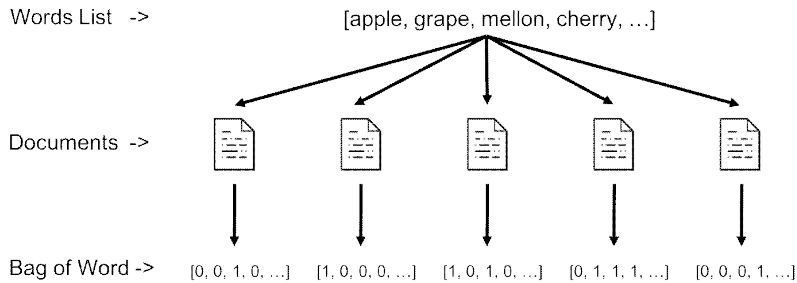


그림 3.2.2 Bag-of-Words 표현 모델

이 텐서 공간 모델은 그림 3.2.1과 같이 텍스트 문서를 Term by Concept 형태의 매트릭스 구조로 만든다. 일반적으로 텍스트 문서 데이터를 기계학습에 사용하기 위한 데이터 구조로는 Bag-Of-Words(그림 3.2.2), Word2Vec(그림 3.2.3) 등이 존재한다. 하지만 본 논문에서 사용한 텐서 공간 모델은 위키피디아의 잘 구축된 온톨로지를 이용하여 만들기 때문에, 온톨로지의 유용한 정보가 데이터 구조 내에 포함되게 된다. 본 논문은 이 점을 잘 활용하여 위조 리뷰 검출 성능을 높일 수 있었다.

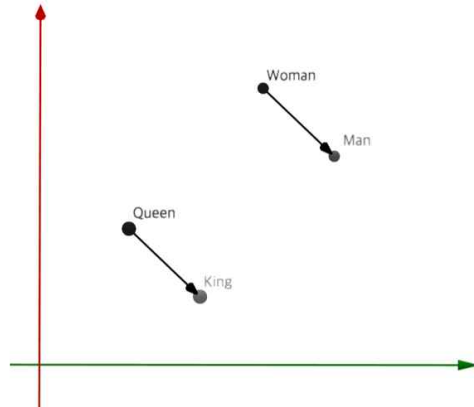


그림 3.2.3 Word2Vec 표현 모델

본 논문에서 사용한 텐서 공간 모델은 개념이라는 요소를 이용하여 데이터 구조를 만들게 된다. 텍스트 문서 데이터 셋에서 개념을 추출하는 과정은 그림 3.2.4와 같다. 먼저 데이터 셋에서 높은 빈도로 출현하는 단어를 추출 한 뒤, 이를 엘라스틱 서치 검색엔진에 질의를 하게 된다. 이 때, 엘라스틱 서치 검색엔진은 위키피디아 데이터베이스를 미리 인덱싱 해 놓는다. 검색 결과 위키피디아 페이지 제목을 반환하게 되는데, 위키피디아 페이지는 제목에 개념이 본문에 개념에 대한 설명으로 구성 된다. 따라서 위키피디아 페이지 제목이 개념이 되는 것이다.

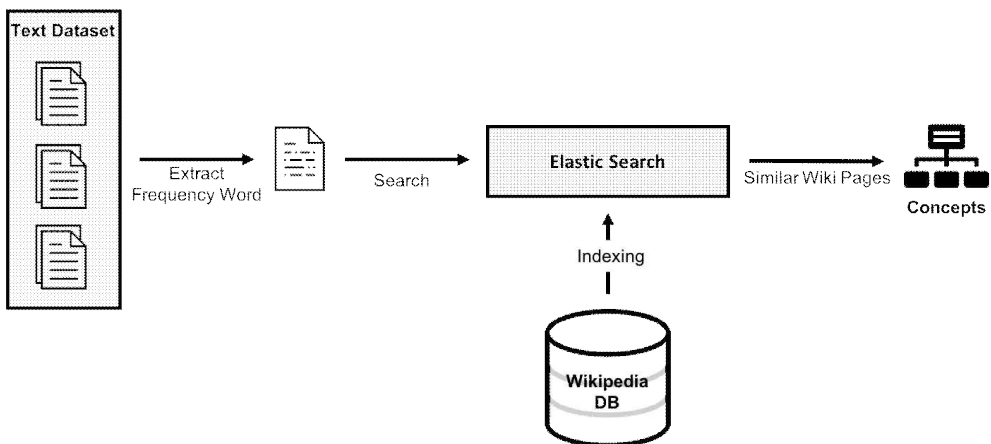


그림 3.2.4 데이터 셋으로부터 개념 추출 과정

그 결과에 대한 간단한 예를 들면 다음과 같다. 데이터 셋 내에서 ‘사과’, ‘포도’, ‘배’, ‘바나나’라는 단어가 자주 출현하였을 경우, 이 정보를 엘라스틱 서치 검색 엔진에 질의한다. 그 결과 ‘과일’이라는 위키피디아 페이지 제목이 반환되게 될 것이라 예상 할 수 있고, 이 데이터 셋의 개념은 ‘과일’이 되는 것이다. 이렇게 추출하는 개념은 문서의 총 수와 복잡도 등을 고려하여 적절한 수를 설정하여 추출이 가능하다. 너무 많은 개념을 추출하려고 하면 데이터 셋과 연관성이 부족한 개념을 이용하게 되고, 너무 적은 개념을 추출하려 하면 위키피디아 온톨로지 정보가 적게 반영되므로 텐서 공간 모델을 사용할 이유가 사라지게 된다. 따라서 경험적으로 적절한 개념의 수를 설정하여 추출하는 것이 필요하다.

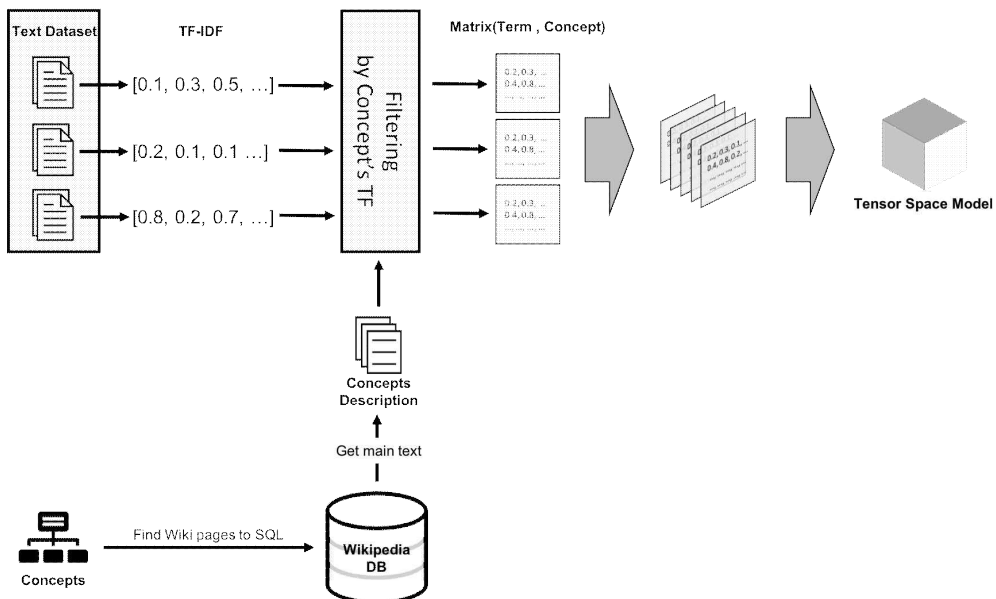


그림 3.2.5 개념과 데이터 셋을 이용한 텐서 공간 모델 생성 과정

그 다음으로 추출한 개념을 이용하여 텐서 공간 모델을 만드는 과정은 그림 3.2.5와 같다. 먼저 텍스트 문서 데이터 셋에서 문서 별로 TF-IDF(Term Frequency - Inverse Document Frequency)를 추출한다. 그 다음 위키피디아 데이터베이스로부터 모든 개념의 설명 내용을 가져와서 각 문서의 TF-IDF정보를

필터링 하게 된다. 그 결과 텍스트 문서가 Term by Concept의 매트릭스 형태가 된다. 이를 데이터 셋 관점에서 보면 Document by Term by Concept의 큐브 형태가 된다. 이 모든 과정을 간단하게 보면 그림 3.2.6과 같다.

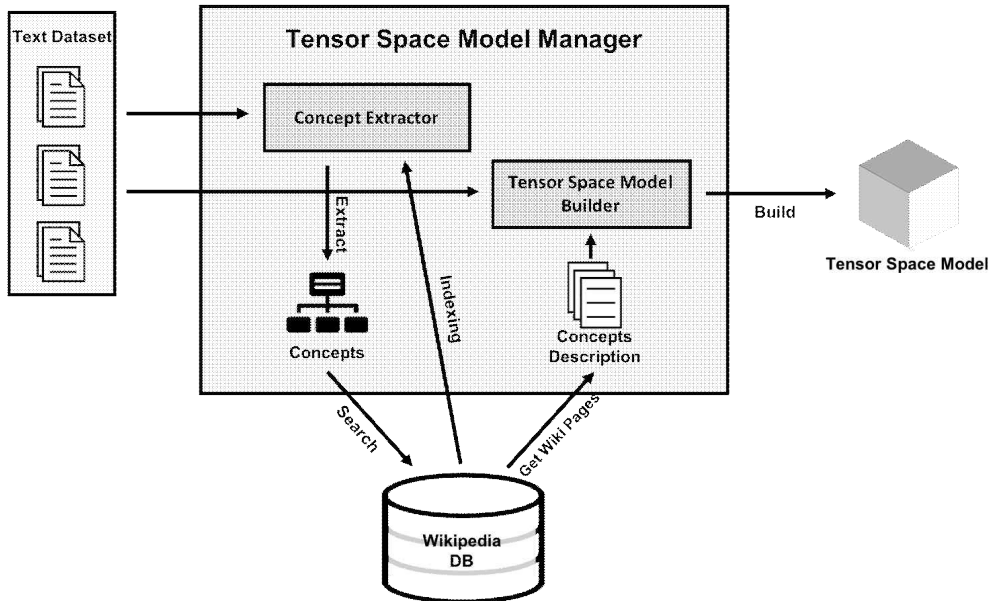


그림 3.2.6 텐서 공간 모델 생성 과정 요약

이와 같은 과정을 통해 만든 결과적으로 위키피디아의 온톨로지 정보가 텐서 공간 모델 내에 포함되게 된다. 이러한 점에서 비교해 볼 때, Bag-Of-Words 구조의 경우 데이터 셋에 포함된 정보만을 이용하게 된다는 점에서 텐서 공간 모델이 더욱 유용하다. Word2Vec 구조의 경우 단어 간의 관계를 학습하여 사용하므로 Bag-Of-Words 구조보다 좋지만, 양질의 대용량 데이터 셋을 추가로 필요하며, 이를 활용하여 단어 간의 관계를 기계학습을 통해 학습해야만 한다. 이 점에서 텐서 공간 모델은 위키피디아 데이터베이스를 이용하여 일련의 프로세스 한 번에 만든다는 점에서 유리하여, 위키피디아 온톨로지라는 훌륭한 정보가 반영된 데이터 구조이다.

추가로 본 논문에서 사용한 텐서 공간 모델은 문서 단위로 데이터를 변형하여

사용한다는 점에서 유사한 Doc2Vec 구조와 비교 해 볼 수 있다. Doc2Vec 구조는 Word2Vec과 유사하게 간단한 신경망을 이용하여 학습하는 과정이 필요하다. 학습 방법으로는 Distributed Memory 방법(DM)과 Distributed Bag-Of-Word 방법(DBOW)이 존재한다. 주로 사용하는 DM방법에 관해 간략히 설명하자면 Word2Vec 학습 방법과 매우 유사하지만 특정 문서 자체의 One-Hot Encoding 결과를 추가하여 학습한다. Word2Vec 학습 과정 중 동일한 문서내의 단어라는 정보를 추가하는 것이다. 그 결과 Word2Vec보다도 기계학습에 입력으로 사용 시 좋은 결과를 보인다고 한다. 하지만 Doc2Vec 구조 또한 학습 과정이 필요하기 때문에 학습을 위한 양질의 대용량 데이터가 필요하고, 학습 과정의 파라미터 설정에 따른 성능 편차가 발생한다.

제4장 개념 임베딩 생성

텍스트를 기계학습에 사용하기 위한 일반적인 데이터 구조인 Bag-of-Words, Word2Vec과 본 논문에서 사용한 시멘틱 텐서 공간 모델의 가장 큰 차이점은 개념이라는 요소가 추가되었다는 점이다. 개념이라는 요소가 중요한 축으로 구성되는 텐서 공간 모델의 특성 상, 단어 중심으로 데이터를 다루는 기존의 심층 인공 신경망 아키텍처의 입력 데이터로 사용하기에 적합하지 않다.

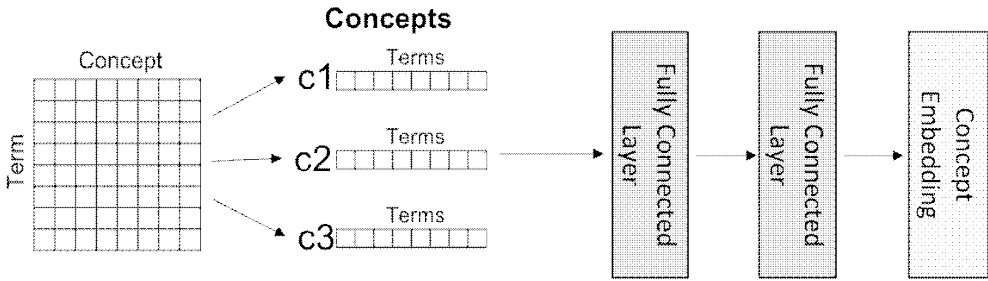


그림 4.1. 개념 임베딩 생성 과정

이를 고려하여 본 논문은 개념 임베딩을 도입하였다. 개념 임베딩이란 그림 4.1과 같이 텐서 공간 모델의 매트릭스 정보를 개념 별로 나누어 2개의 Fully-Connected Layer를 거치는 것을 통해 만든다. 수식을 통해 개념 임베딩이 만들어 지는 과정을 설명한다. 문서 1개의 텐서 공간 모델을 $C_{document}$ 라 하면,

$$C_{document} = \begin{matrix} & \text{concepts} \\ \text{terms} & \begin{pmatrix} 0 & 1 & \dots & 2 \\ 5 & 4 & \dots & 0 \\ \dots & \dots & \ddots & 3 \\ 2 & 1 & \dots & 1 \end{pmatrix} \end{matrix} \quad (1)$$

와 같다. 다음으로 $C_{document}$ 의 i 번째 개념 벡터를 c_i 라 하면,

$$c_i = C_{document}[i] \quad (2)$$

와 같이 표현 할 수 있다. i 번째 개념의 개념 임베딩을 E_{c_i} 라 정의하고, 이를 구하는 과정을 설명한다. i 번째 개념 임베딩을 구하는 k 번째 레이어의 출력 값을 $y_{c_i}^{(k)}$, 가중치를 $w_{c_i}^{(k)}$, 바이어스를 $b_{c_i}^{(k)}$, 활성화 함수를 $f^{(k)}$ 로 정의한다.

$$y_{c_i}^{(1)} = f^{(1)}(w_{c_i}^{(1)} c_i + b_{c_i}^{(1)}) \quad (3)$$

$$E_{c_i} = f^{(2)}(w_{c_i}^{(2)} y_{c_i}^{(1)} + b_{c_i}^{(2)}) \quad (4)$$

위의 수식과 같은 과정을 통해 만든 개념 임베딩은 생성 과정 중, 각 개념 간의 연결이 존재하지 않는다. 즉, 텐서 공간 모델의 개념은 서로 독립적인 요소이므로, 이를 적절하게 고려한 임베딩이 만들어지게 되는 것이다.

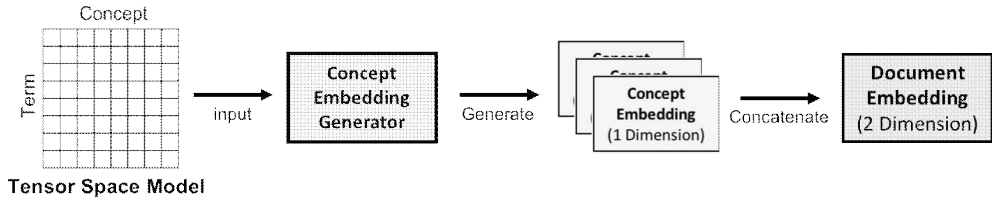


그림 4.2. 문서 임베딩 생성 과정

다음으로 그림 4.2와 같이 문서의 1개에서 텐서 공간 모델 내 모든 개념 임베딩을 구한 뒤, 하나로 모은 것을 문서 임베딩이라 명명하였고, $E_{document}$ 로 정의한다.

$$E_{document} = [E_{c_1}; E_{c_2}; \dots; E_{c_{n-1}}; E_{c_n}] \quad (5)$$

문서 임베딩 생성 과정을 알고리즘으로 표현하면 다음과 같다.

알고리즘 4.1. 문서 임베딩 생성 과정

Input: TensorSpaceModel $C_{document}$
FullyConnectedLayer $L1, L2$

Initialize $nConcepts = \text{numOfConcept}(C_{document});$
 $nWords = \text{numOfWord}(C_{document});$
DocumentEmbedding $E_{document} \leftarrow \text{Matrix}(nWords, nConcepts);$

for i in $1:nConcepts$ do:
 $C_i \leftarrow C_{document}[i];$
 $outL1 \leftarrow L1.\text{output}(C_i);$
 $E_{ci} \leftarrow L2.\text{output}(outL1);$
 $E_{document}.\text{setRow}(i, E_{ci});$

Output: $E_{document}$

알고리즘 4.1. 문서 임베딩 생성 과정

알고리즘 4.1의 문서 임베딩을 생성하는 과정을 본 논문이 제안하는 심층 인공 신경망 아키텍처 내에 포함하였다. 이를 통하여 텐서 공간 모델의 개념 요소를 적절하게 고려한 아키텍처를 설계할 수 있다.

개념과 문서 임베딩을 이용 시 높은 위조 리뷰 검출 성능을 기대한다. 이는 5장에서 소개할 인공 신경망 아키텍처에서 개념, 문서 임베딩을 아키텍처 내부에 포함함으로써 위조 리뷰 검출 학습 시에 개념, 문서 임베딩이 동시에 학습이 되기 때문이다. 즉, 개념, 문서 임베딩을 따로 학습하지는 않지만, 결과적으로 위조 리뷰 검출 학습 과정에서 개념, 문서 임베딩이 중요한 정보를 잘 표현하도록 학습된다는 것이다.

제5장 심층 인공 신경망 아키텍처

본 논문은 개념 요소를 이용하여 텍스트 문서를 텐서 공간 모델로 변형한 데이터를 인공 신경망 아키텍처의 입력으로 사용하고 있다. 이번 장은 입력 데이터인 텐서 공간 모델에 적합한 인공 신경망 3가지를 제안하고자 한다.

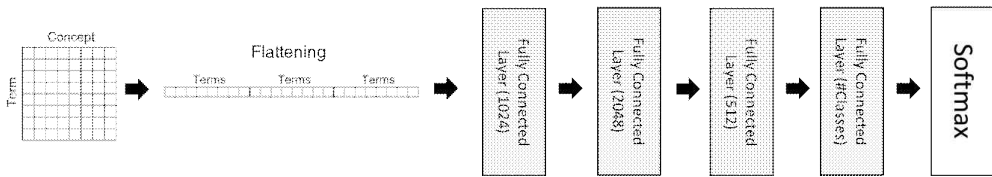


그림 5.1 심층 신경망 디자인 1 (TSM DNN)

첫 번째 제안 인공 신경망 아키텍처는 TSM DNN이라 명명하였으며 그림 5.1과 같다. 이 아키텍처는 텐서 공간 모델 데이터를 입력 받을 시, 단순히 데이터를 펼쳐서(Flattening) 벡터 형태로 만들어 몇개의 완전 연결 레이어를 거치게 되는 구조이다. 매우 단순한 구조이지만, 데이터를 벡터 형태로 만들 때 개념 별로 잘르는 구조를 택하였기에 본 논문이 사용한 텐서 공간 모델에 적합한 인공 신경망 아키텍처라고 할 수 있다.

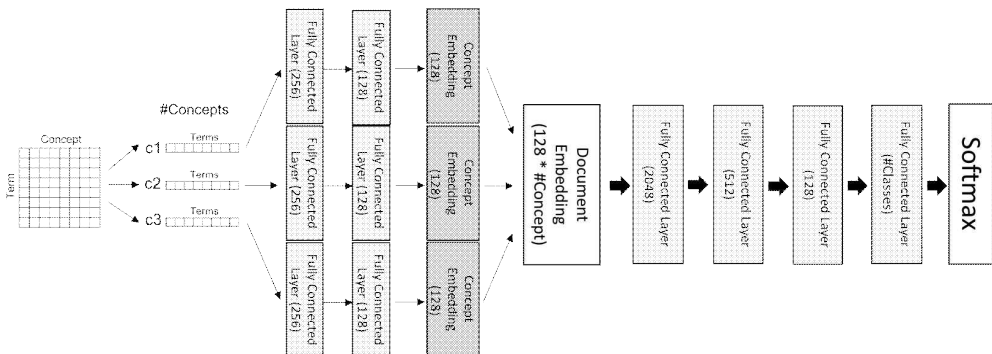


그림 5.2 심층 신경망 디자인 2 (Concept-wise TSM DNN)

두 번째 인공 신경망 아키텍처는 Concept-wise TSM DNN이라 명명하였다. 이 아키텍처는 앞서 4장에서 소개한 개념 임베딩과 이를 모은 문서 임베딩을 아키텍처 내부에 포함하는 것을 통해 텐서 공간 모델에 적합한 모델을 설계하였다. 텐서 공간 모델 데이터를 각 개념 별로 Term데이터를 잘라서 개념 임베딩을 만든 뒤, 모든 개념 임베딩을 2D형태로 모아서 문서 임베딩을 만들게 된다. 그 뒤 문서 임베딩을 몇 개의 완전 연결 레이어를 거치게 함으로써 문서 분류를 수행하게 되는 구조이다. Concept-wise TSM DNN 아키텍처는 개념 임베딩을 통해 텐서 공간 모델의 개념 요소를 적극적으로 이용하였기에 적합한 인공 신경망 아키텍처이다.

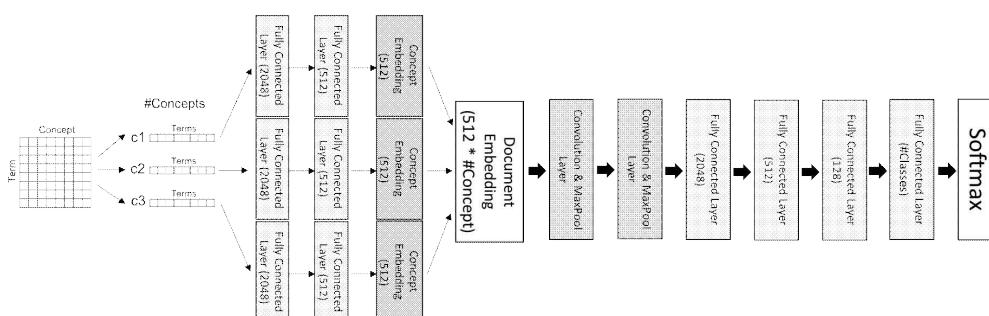


그림 5.3 심층 신경망 디자인 3 (Concept-wise TSM CNN)

세 번째 제안 인공 신경망 아키텍처는 Concept-wise TSM CNN이다. (그림 5.3) 이 아키텍처는 두 번째 아키텍처와 유사하지만 문서 임베딩 이후에 2개의 컨볼루션 레이어를 추가하였다. 일반적으로 컨볼루션 레이어는 2차원(Dimension) 데이터인 이미지 분류에 사용시 높은 성능을 보인다. 문서 임베딩 또한 2차원 데이터이므로 컨볼루션 레이어를 사용하여 분류 성능을 높일 수 있다. 하지만 컨볼루션 레이어를 2개 이상 중첩하여 사용하게 되면 파라미터의 수가 폭발적으로 증가하게 된다. 이를 해결하기 위하여 본 논문은 컨볼루션 레이어의 다음에 Maxpooling 레이어를 배치하고 stride를 2로 설정하여 입력과 출력의 크기를 일정하게 유지하도록 만들었다. 세 번째 아키텍처 또한 개념 임베딩을 인공 신경망 내부에 포함함으로써 텐서 공간 모델의 개념 요소를 적절하게 고려한 아키텍처라고 할 수 있다.

본 논문이 제안하는 세 가지 인공 신경망 아키텍처는 텐서 공간 모델을 입력으로 사용하기에, 단어 단위를 중심으로 설계하는 일반적인 딥러닝 기반 단문 분류 기법들과 달리 텐서 공간 모델의 핵심 요소인 개념을 중심으로 아키텍처를 설계하였다. TSM DNN 아키텍처는 개념 단위로 데이터를 나누어 펼침으로써 이를 고려하였다. Concept-wise TSM DNN과 Concept-wise TSM CNN은 아키텍처 내에 개념 임베딩을 포함하는 것을 통해 이를 고려하였다.

제6장 위조 리뷰 검출 성능 평가

제1절 실험 데이터 셋

본 논문은 위조 리뷰 검출 성능 평가를 위하여 해외 외식 추천 사이트인 Yelp에서 유저들의 실제 리뷰를 수집하였다. 이 사이트는 위조 리뷰 데이터를 삭제하지 않고, 위조 리뷰로 분류하여 관리하고 있다. Yelp는 위조 리뷰를 검출할 때, 리뷰 데이터만을 이용하여 위조 여부를 판별하는 것이 아니라 작성자의 활동 내역이나 계정 생성 일자 등의 여러 가지 요소를 이용하여 거의 완벽한 수준의 성능을 보인다. 이런 완벽한 수준의 검출 성능을 보임에도 불구하고 Yelp의 검출 데이터를 사용한 이유는 본 논문의 방법이 오직 리뷰 데이터만을 이용하여 위조 리뷰를 검출하기 때문이다.

Yelp 사이트 리뷰 데이터	
총 데이터 수	12,000 건
위조 리뷰 수	6,000 건
실제 리뷰 수	6,000 건
최소 리뷰 길이	800 자

표 6.1.1 리뷰 데이터 상세 속성

본 논문에서 사용한 리뷰 데이터의 상세 정보는 표 6.1.1과 같다. 위조 리뷰 6,000건 실제 리뷰 6,000건을 이용하였다. 추가적으로 긍정, 부정 리뷰의 비율을 1:1로 맞추었는데, 위조 긍정 리뷰 3,000건, 위조 부정 리뷰 3,000건, 실제 긍정 리뷰 3,000건, 실제 부정 리뷰 3,000건을 이용하였다. 이렇게 리뷰를 나누어 수집한 이유는 위조 리뷰를 작성하는 경우를 2가지로 나누어 볼 수 있기 때문이다. 의

도적으로 식당의 평가를 높이거나 낮추는 경우가 존재하기 때문에, 이를 고려하여 같은 비율로 학습 데이터를 구성하면 더욱 높은 위조 리뷰 검출 성능을 보일 수 있을 것이라고 기대하였다. 그리고 리뷰는 최소 800자 이상의 문자를 사용한 데이터만을 사용하였다. 너무 짧은 리뷰를 이용하는 경우 본 논문의 텐서 공간 모델을 만들 때, 0으로 채워진 문서로 만들어질 가능성이 있다. 또한, 리뷰 데이터만으로 위조 여부를 판별하기 때문에, 한, 두 줄의 리뷰만으로는 사람의 수준에서도 판별할 수 없어서 최소 800자 이상의 리뷰만을 선정하였다.

긍정 리뷰	실제 리뷰	A free standing Miracle Mile- hell yes! Since the downfall of the locations I grew up with at Park Central, and ChrisTown Malls, the closing of the locations at both Arrowhead and what used to be The Colonade Mall (turned into some kind of strip mall place) ...
	위조 리뷰	This is probably my most favorite cheese steak shop in the whole wide world. For starters it's open til 2 Sunday through Thursday. And open til 3 Fridays and Saturdays. And the food itself is so bomb. ...
부정 리뷰	실제 리뷰	Unfortunately we got an asshole waiter who was nothing short of snarky comments and attitude. Here's one of many examples. When I initially booked our reservation I asked the reservationist if they did anything special for birthdays. He said yes. ...
	위조 리뷰	I have been to Zeitgeist before and I loved it. So last night, I specifically asked my friend if we could go there for warm-up cocktails before heading to our friend's Hippie Happy Birthday House Party in Oakland...therefore, we were dressed as hippies for the party. ...

표 6.1.2 리뷰 샘플

실제 위조, 실제 리뷰를 살펴보면 표 6.1.2와 같다. 일반적으로 긍정적인 실제 리뷰의 경우 무난하게 메뉴를 칭찬하는 것을 볼 수 있고, 긍정적 위조 리뷰의 경우 상당히 과장되게 칭찬하는 것을 볼 수 있다. 부정적 리뷰의 경우 실제 리뷰는

식당의 어떠한 점이 마음에 들지 않았는지 자세하게 설명하는 반면, 위조 리뷰는 식당이 아닌 자신들의 상황을 묘사에 치중하는 것을 알 수 있었다. 이러한 실제와 위조 리뷰 간의 다른 패턴을 심층 인공 신경망 아키텍처를 이용하여 학습함으로써 위조 리뷰 검출을 수행하도록 한다.

실제 리뷰		위조 리뷰	
긍정 리뷰	부정 리뷰	긍정 리뷰	부정 리뷰
Breakfast Chicken fried steak Christmas dinner Danish cuisine English cuisine Finnish cuisine Hamburger International...Id's products Kansas Cit-style barbecue Salad Taco Bueno Taco Cabana Tamil cuisine Types of restaurant Wendy's	Automated restaurant Cha chaan teng Chipotle Mexican Grill Customer service Dining car English cuisine Fast food restaurant Hamburger History of McDonald's In-N-Out Burger International...Id's products Nice guy Taco Cabana Types of restaurant Wendy's Zappos	Christmas dinner Danish cuisine Fast food restaurant Finnish cuisine Hamburger International...Id's products Salad Taco Bueno Taco Cabana Tamil cuisine Types of restaurant Wine and food matching	Automated restaurant Cha chaan teng Chipotle Mexican Grill Dining car Etiquette in Japan Fast food restaurant History of McDonald's In-N-Out Burger International...Id's products Nasi Padang Side dish Tableware Taco Cabana Tamil cuisine Types of restaurant

표 6.1.3 위조 리뷰 데이터 셋에서 추출한 개념들

본 논문에서 사용하는 텐서 공간 모델의 경우 문서 데이터 셋 내에서 개념을 추출하여 이용하여야만 한다. 텐서 공간 모델을 만들기 위해 추출한 개념들은 표 6.1.3과 같다. 위조 리뷰 데이터 셋의 단어 복잡도와 문서의 양을 고려하여 총 31개의 개념을 추출하였다. 그리고 데이터 셋 내에서 희소 단어를 제거한 단어 종류의 수는 157개였다. 따라서 텐서 공간 모델은 단어 수 x 개념의 수이므로, 문서 1

개당 157×31 의 크기를 가지는 매트릭스 형태가 되었다. 데이터 셋 관점에서 볼 때, 텐서 공간 모델은 $12,000 \times 157 \times 31$ 의 큐브 형태가 된다.

제2절 실험 기법

본 논문은 위조 리뷰 검출 성능 평가를 위해 6가지 대조 기법을 선정 하였다. 그 중 2가지는 전통적인 기계학습 기법인 SVM, Naive Bayes이고, 나머지 4가지는 딥러닝 방법으로 DNN[12], CNN[13], VDCNN-depth17, VDCNN-depth29 기법을 선정하였다.

먼저 전통적인 기계학습 기법들은 텍스트 문서 데이터를 TF-IDF 값으로 변형하여 입력데이터로 사용하였다. SVM의 경우 커널 함수로 RBF를 사용하였고, C값은 1.0를 사용하였다. Naive Bayes 기법의 경우 Multinomial Naive Bayes를 사용하였고, alpha값으로 1.0으로 설정하였다.

다음으로 딥러닝 기법들에 대하여 설명하겠다. DNN과 CNN 기법의 경우 복잡한 구조를 설정하지 않고 몇 개의 레이어를 사용한 것을 이용하였다. 또한 텍스트 문서를 Bag-Of-Words 데이터로 변형하여 입력데이터로 사용하였다. DNN 기법은 총 7개의 완전 연결 레이어를 이용하여 구성하였으며, 각 레이어 순서대로 #words, 1024, 2048, 4096, 1024, 256, 2의 뉴론을 가지도록 설정하였다. 활성화 함수로는 Relu를 이용하였고, 마지막 레이어는 분류를 위하여 Softmax함수를 사용하였다. 이 기법은 최적화 방법으로 Adam을 사용하여 학습하였고 Batch 크기는 64를 10epoch 동안 진행하였다.

CNN 기법은 컨볼루션 레이어와 Maxpooling 레이어 2개를 순서대로 2번 반복 후에 완전 연결 레이어 3개로 구성하였다. 첫 번째 컨볼루션 레이어는 커널 크기를 3, 필터 8개, Padding를 1로 설정하였다. 두 번째 컨볼루션 레이어는 커널 크기를 3, 필터 32개, Padding를 1로 설정하고 필터는 32개로 설정하였다. Maxpooling 레이어는 두 개 모두 동일하게 커널 크기를 3, Padding을 1, Stride를 2로 설정하였다. 이 후에 나온 결과를 1차원 형태로 펼친(Flatten) 다음 완전 연결 레이어의 입력으로 한다. 각 완전 연결 레이어는 순서대로 512, 64, 2개의 뉴론을 가지도록 설정하였고, 활성화 함수로 Relu를 이용하고 마지막 레이어의 출력 값에 Softmax 함수를 적용하여 분류를 수행하도록 하였다. 이 기법은 최적화

방법으로 Adam을 사용하였고 학습률은 0.01, Batch 크기 32를 10epoch 동안 수행하도록 하였다.

마지막으로 VDCNN-depth17, VDCNN-depth29 기법들은 Conneau의 기법[]들이다. 4가지 depth를 가지는 기법을 제안하였는데, 이 중에서 실험적으로 성능이 가장 좋았던 2가지(depth17, depth29)를 선정하였다. 이 기법은 문서 데이터를 Bag-Of-Words 등으로 변형하여 사용하지 않고 텍스트 데이터를 문자 단위로 입력하여 문서 분류를 수행하는 기법이다. 문서 내의 최대 문자 수를 1014개로 제한하여 고정된 입력 크기를 가지도록 설정하였다. 그래서 최대 문자 수를 넘는 텍스트는 이용하지 않는다. 이 기법은 최적화 방법으로 모멘텀을 이용하였으며, 학습률은 0.01로 설정하고 Batch 크기 64를 50epoch 동안 학습하였다.

총 데이터 수가 12,000개라서 학습과 평가 데이터를 어떻게 나누는지에 따라서 성능의 변화가 심하였다. 그래서 본 논문의 제안 기법들과 대조 기법들을 공정한 평가를 위해 5Fold Cross-Validation을 수행한 위조 리뷰 검출 정확도 값을 비교 지표로 이용하였다.

제3절 인공 신경망 파라미터 튜닝

이번 절은 본 논문의 제안 아키텍처들의 파라미터를 설명한다. 먼저, TSM DNN 아키텍처는 $157 * 31$ 크기의 벡터를 입력 받아 4개의 완전 연결 레이어를 거치게 된다. 각 완전 연결 레이어는 순서대로 1024, 2048, 512, 2개의 뉴론을 가지도록 설정하였다. 활성화 함수는 LeakyRelu를 사용하였으며, 각 완전 연결 레이어의 출력 이후에 Batch Normalization을 수행하도록 하였다. 마지막 완전 연결 레이어 이후에는 Softmax 함수를 이용하여 분류를 수행하도록 하였다.

다음으로 Concept-wise TSM DNN 아키텍처는 개념 임베딩 생성을 위한 완전 연결 레이어를 각각 256, 128개의 뉴론을 가지도록 설정하였다. 그 결과 문서 임베딩의 크기는 $128 * 31$ 의 2차원 모양이 되었다. 이 후, 문서 임베딩을 펼친(Flatten) 다음 4개의 완전 연결 레이어를 거치게 되는데, 각 레이어는 2048, 512, 128, 2개의 뉴론을 가지도록 설정하였다. 각 완전 연결 레이어 이후에는 Batch Normalization을 수행하도록 하였고, 활성화 함수는 LeakyRelu를 사용하였다. 마지막 완전 연결 레이어 이후에는 Softmax 함수를 이용하여 분류를 수행하도록 하였다.

마지막 모델인 Concept-wise TSM CNN의 경우 컨볼루션 레이어를 거치며 크기가 작아지기 때문에 개념 임베딩의 크기를 4배로 늘렸다. 즉, 개념 임베딩을 생성하는 완전 연결 레이어의 뉴론 수를 각각 2048, 512로 설정하였고, 그 결과 문서 임베딩은 $512 * 31$ 이 되었다. 생성된 문서 임베딩을 1D 컨볼루션 레이어를 거치도록 하였다. 컨볼루션 레이어의 커널 크기는 3, Padding은 1로 설정하였으며, 2개의 컨볼루션 레이어의 필터 개수는 각각 64, 128개로 하였다. 그리고 컨볼루션 레이어 다음 커널 크기 3, Stride는 2, Padding은 1인 1D Maxpooling 레이어를 배치하였다. 이 후, 완전 연결 레이어 4개를 배치하고 각각 2048, 512, 128, 2개의 뉴론을 가지도록 설정하였다. Concept-wise TSM CNN의 모든 완전 연결 레이어 이후에는 Batch Normalization을 거치도록 하였고 활성화 함수는 LeakyRelu를 사용하였다. 마지막 완전 연결 레이어 이후에는 Softmax 함수를 이

용하여 분류를 수행하도록 하였다.

본 논문에서 제안한 TSM DNN, Concept-wise TSM DNN, Concept-wise TSM CNN의 학습은 최적화 함수로 Adam, 학습률 0.01, Batch 크기 32를 5 epoch 수행하도록 하였다.

추가적으로 더욱 많은 레이어를 가지는 인공 신경망 아키텍처를 제안하지 않았는지를 실험을 통해 설명하기 위하여 Concept-wise TSM CNN의 문서 임베딩 이후 레이어 수를 2배로 한 Concept-wise TSM CNN 2 아키텍처를 만들었다. 이 아키텍처는 문서 임베딩 생성까지 Concept-wise TSM CNN 아키텍처와 동일하지만 문서 임베딩 이후 배치한 컨볼루션, Maxpooling, 완전 연결 레이어의 수를 각각 4, 4, 8개 배치하였다.

제4절 성능 평가

이번 절은 대조 기법과의 위조 리뷰 검출 비교를 통해 본 논문의 제안 기법의 성능을 평가한다.

	SVM	Naïve Bayes	DNN	CNN	VDCNN Depth-17	VDCNN Depth-29	TSM-DNN	Concept-wise DNN	Concept-wise CNN	Concept-wise CNN 2
Accuracy	54.21%	46.05%	76.02%	75.69%	90.08%	90.50%	93.56%	94.34%	94.99%	94.49%

표 6.4.1 각 기법 별 위조 리뷰 검출 성능 비교

각 기법 별 위조 리뷰 검출 성능은 표 6.4.1과 같다. 먼저 전통적인 기계학습 방법인 SVM과 Naive Bayes 기법의 경우 50% 내외의 성능을 보였는데, 이는 아예 학습이 되지 않은 결과이다. 단순히 위조 여부를 판별하는 Binary Classification 이기 때문에 위조 여부에 대한 확률은 50%이기 때문이다.

다음으로 DNN, CNN 기법의 경우 각각 75, 76%의 성능을 보였다. 전통적인 기계학습 기법과 비교 시 훨씬 높은 성능을 보였지만, 본 논문의 제안 아키텍처와 비교 시 앞도적인 성능 차이를 보였다.

2017 년도 State-Of-The-Art 텍스트 분류 성능을 보인 VDCNN-Depth17, VDCNN-Depth29 기법의 경우 90%를 넘는 높은 검출 성능을 보였다. 이 기법들은 문서를 Bag-Of-Words 또는 Word2Vec과 같이 다른 구조로 변형하지 않고 문자 단위로 입력하여 위조 리뷰 검출을 수행하였음에도 높은 성능을 보였다. 하지만 본 논문의 제안 기법들은 가장 높은 성능을 보인 Concept-wise TSM CNN의 경우 95%에 근접한 성능을 보였는데 이는 5% 가까이 높은 검출 성능이다. 잘못 분류할 확률은 반으로 줄인 결과로 매우 놀라운 성능이다. 그림 6.4.1에서 막대 그래프를 통해 각 기법 간의 위조 리뷰 검출 성능을 보면 확실히 제안 기법들의 성능이 높은 것을 알 수 있다.

본 논문의 제안 심층 인공 신경망 아키텍처들은 개념 임베딩이라는 요소를 인공 신경망 아키텍처 내부에 포함함으로써 텐서 공간 모델의 개념 요소를 적절하게 이

용하여 위조 리뷰 여부를 잘 판별할 수 있게 학습하는 것으로 생각한다.

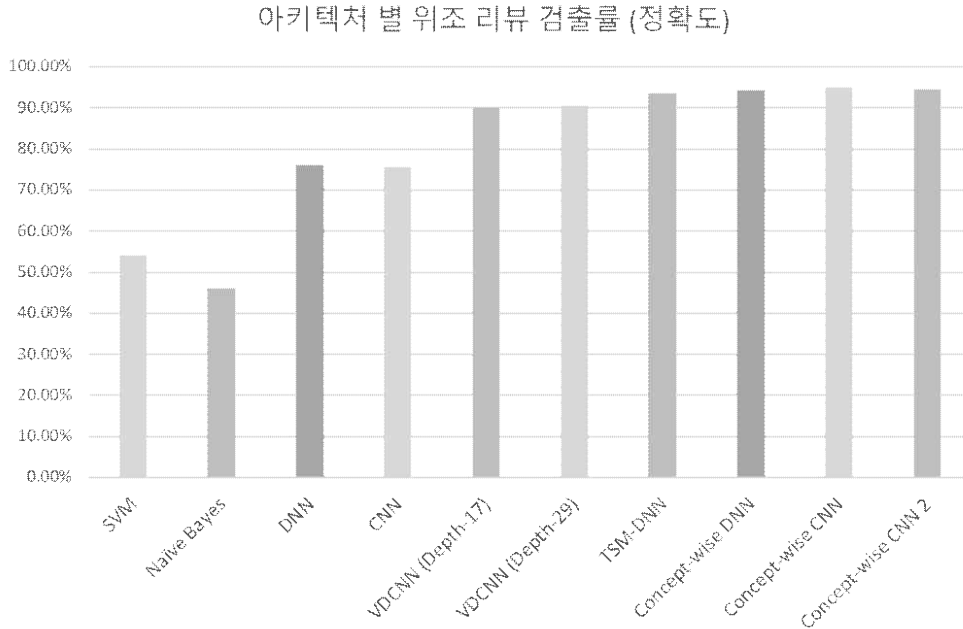


그림 6.4.1 각 기법 별 위조 리뷰 검출 성능 비교 그래프

추가적으로 본 논문의 제안 아키텍처들은 최근 인공 신경망 아키텍처와 비교 시 상당히 적은 수의 레이어를 사용하고 있다. 그럼에도 불구하고 매우 깊은 구조를 가지는 VDCNN 기법들보다도 높은 성능을 보였는데, 이는 단순히 레이어 수를 2 배로 만든 Concept-wise TSM CNN 2 아키텍처의 위조 리뷰 검출 성능이 오히려 본래 아키텍처인 Concept-wise TSM CNN보다 낮은 것을 통해 알 수 있다. 본 논문에서 실험한 데이터 셋은 총 문서의 수가 12,000개 여서 매우 깊은 수준의 신경망을 학습할 시 데이터 셋 자체를 학습할 가능성이 높다.(Over Fitting) 이에 대하여 인공 신경망의 총 레이어 수의 변화에 따른 성능 차이를 실험하였다. 그림 6.4.2를 보면 Concept-wise TSM CNN의 레이어 8개를 정점으로 레이어 수가 많거나 적을 시 더 낮은 성능을 보인 것을 알 수 있다. 이는 전문적인 용어로 Bias-Variance Tradeoff라고 하는데 너무 큰 Capacity를 가지는 신경망을 사용

시 데이터 셋 자체를 학습(Over Fitting)하고 너무 작은 Capacity를 가지는 신경망을 사용 시, 데이터 셋 내의 패턴을 학습하지 못(Under Fitting)하는 현상이 발생한다. 즉, 본 논문의 제안 아키텍처인 Concept-wise TSM CNN의 경우 두 현상의 사이에 적절한 지점을 선택하여 가장 높은 성능을 보이게 한 훌륭한 아키텍처이다.

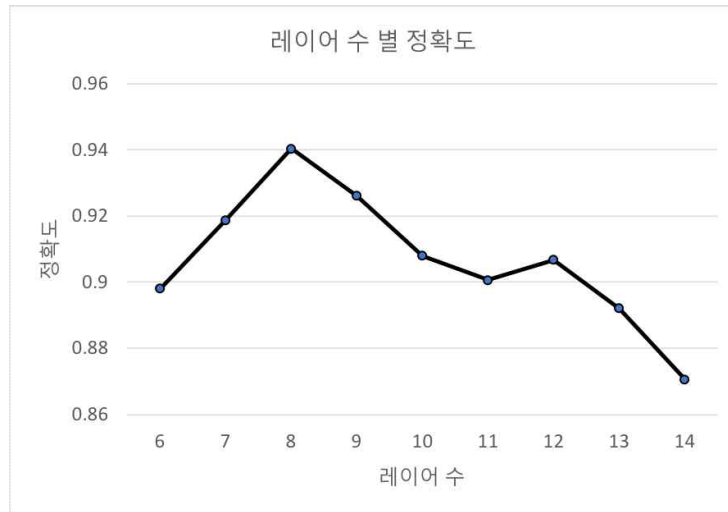


그림 6.4.2 레이어 수 별 위조 리뷰 검출 정확도 변화

제7장 결론

제1절 요약 및 기여 내용

본 연구는 위조 리뷰 검출 문제를 해결하기 위하여 텐서 공간 모델을 사용하였고, 이에 적합한 인공 신경망 아키텍처를 제안하였다. 텐서 공간 모델은 개념이라는 요소가 문서에 추가된 모델이므로, 단어를 중심으로 해석한 기존의 기법들의 입력 데이터로 사용 시 적합하지 않다. 따라서 텐서 공간 모델의 생성 과정을 분석하여 이 모델의 특성을 이해하는 과정을 수행하였다. 그 결과 개념 요소를 고려한 개념 임베딩과 문서 임베딩을 정의하고 이를 제안 아키텍처 내부에 포함하는 것을 통해 이 모델에 적합한 아키텍처를 설계할 수 있었다. 제안 아키텍처는 위조 리뷰 검출 학습 시 개념 임베딩과 문서 임베딩이 함께 학습되어 위조 리뷰 검출 성능을 매우 높은 점을 전통적인 기계학습 기법 2가지와 딥러닝 기법 4가지와의 5 Fold cross-validation 결과를 통해 증명하였다. 결과적으로 본 논문은 위조 리뷰 검출을 위한 일련의 프로세스를 제안하였다.

향후 연구 과제로는 개념 임베딩과 문서 임베딩을 이용하여 성능을 높였지만 이것을 사용함으로써 인해 위조 리뷰 검출 성능이 증가하였다는 증거가 부족하다. 따라서 최종 학습된 제안 아키텍처에서 문서 임베딩 결과를 분석하여 실제로 의미 있는 표현정보를 담고 있는 지를 확인하여 보아야 할 것이다. 이는 실제 리뷰와 위조 리뷰간의 문서 임베딩을 시각화 또는 거리함수를 이용한 분석을 통해 수행할 수 있을 것이다.

참 고 문 헌

- [1] 김한준, 장재영, 위키피디어 기반 개념 공간을 가지는 시멘틱 텍스트 모델, 한국전자거래학회지 제19권 3호, 2014, pp. 107-123.
- [2] Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean, “Efficient Estimation of Word Representations in Vector Space”, *CoRR*, 2013.
- [3] Long Ma, Yanqing Zhang, “Using Word2Vec to process big text data”, 2015 IEEE International Conference on Big Data, 2015.
- [4] Q V Le, T Mikolov, “Distributed Representations of Sentences and Documents”, In Proceedings of the international Conference on Machine Learning, 2014, pp. 1188-1196.
- [5] Pengfei Liu, Xipeng Qui, Xuanjin Huang, “Recurrent Neural Network for Text Classification with Multi-Task Learning”, *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, 2016.
- [6] Ji Young Lee, Franck Dernoncourt, “Sequential Short-Text Classification with Recurrent and Convolutional Neural Networks”, *HLT-NAACL*, 2016.
- [7] Conneau Alexis, Holger Schwenk, Yann LeCun, Loic Barrault, “Very Deep Convolutional Networks for Text Classification”, *Proceeding of the Fifth Conference of the European Chapter of the Association for Computational Linguistics*, 2017.
- [8] Siwei Lai, Liheng Xu, Kang Liu, Jun Zhao, “Recurrent Convolutional Neural Networks for Text Classification”, *Proceedings of the Twenty-Ninth AAAI Conference on Artificial intelligence*, 2015.
- [9] Thang Luong, Richard Socher, Christopher Manning, “Better word representations with recursive neural networks for morphology”, *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*. 2013.
- [10] Han-joon Kim, Jiyun Kim, Jinseog Kim, Pureum Lim, “Towards perfect text classification with Wikipedia-based semantic Naive Bayes learning”,

Neurocomputing, Vol.315, 2018.

- [11] Conneau Alexis, Holger Schwenk, Yann LeCun, Loic Barrault, “Very Deep Convolutional Networks for Text Classification”, *Proceeding of the Fifth Conference of the European Chapter of the Association for Computational Linguistics*, 2017.
- [12] Weubi Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, FE. Alsaddi, “A survey of deep neural network architectures and their applications”, *Neurocomputing*, Vol.234, 2017.
- [13] Yoon Kim, “Convolutional neural networks for sentence classification”, *arXiv preprint arXiv:1408.5882*, 2014.

Abstract

Tensor Space Model based Deep Neural Network Algorithm for Fake Review Detection

Pureum Lim

School of Electrical & Computer Eng.

Graduate School of the Univ. of Seoul

Supervised by Prof. Han-joon Kim, Ph. D.

Recently, As Social media services evolve, People is getting information from social review. But, Social reviews are very unreliable because it is written by anonymous users. Therefore, The technique needs for determine fake of review is a growing. This paper uses a Text Classification technique to solve the problem of fake review detection. However, the detection performance is low when using the conventional Text Classification method because Social reviews have the characteristics of the short review and daily word usage. To solve this problem, we intend to achieve a high fake review detection performance by transforming a document into a Tensor Space Model and designing an artificial neural network architecture suitable for this model.

**key words : Fake Review Detection, Text Classification, Deep Learning,
Text Mining**

감사의 글

지난 2년간 석사 과정을 돌이켜 보면 너무나도 짧은 순간이었던 것 같습니다. 학부과정과는 다른 공부를 하고 연구를 수행하며 몇 개의 논문을 작성해보기도 한 시간은 저에게 있어 매우 뜻 깊었던 시간이었습니다. 힘든 시기도 있었지만 끊임없이 노력하여 졸업을 앞두고 보니 큰 보람을 느낍니다. 이제 다시 경험할 수 없는 시간을 보내고 사회로 진출할 시간이 다가와 갑니다.

그동안 경험해 보지 못한 공부를 훌륭하게 배울 수 있도록 끊임없는 지도를 해주신 김한준 교수님께 너무나도 감사드립니다. 힘든 시기에도 제가 처한 상황에 공감하고 포기하지 않도록 조언해 주시고 길을 찾을 수 있도록 조언해 주셔서 정말 감사드립니다. 또한 졸업 논문에 대한 충고와 조언을 해주신 장석진 교수님과 이영민 교수님 감사드립니다. 졸업 논문 뿐만 아니라 석사 과정동안 수강하였던 과목들에 대해서도 많은 것을 배울 수 있었습니다.

연구실 생활 도중 1년간은 3명에서 지내다 보니 쓸쓸하게 보낸 시간이 많았지만 그래도 연구실 장을 맡아 든직하게 있어주었던 상원형에게 감사드립니다. 항상 열심히 공부하여 카이스트 박사과정 났게 된 태준이도 앞으로 좋은 일만 있길 바랍니다. 마지막으로 6개월만에 연구실의 모든 것을 이어받아 이끌어 나갈 회원씨와 승엽씨도 얼마 남지 않은 기간 알고 있는 모든 것을 알려주도록 노력할 것이고, 열심히 해서 좋은 결과 얻길 바랍니다.

저와 인연이 되었던 모든 분들께 감사의 말을 드리며, 앞으로도 열심히 살도록 하겠습니다.