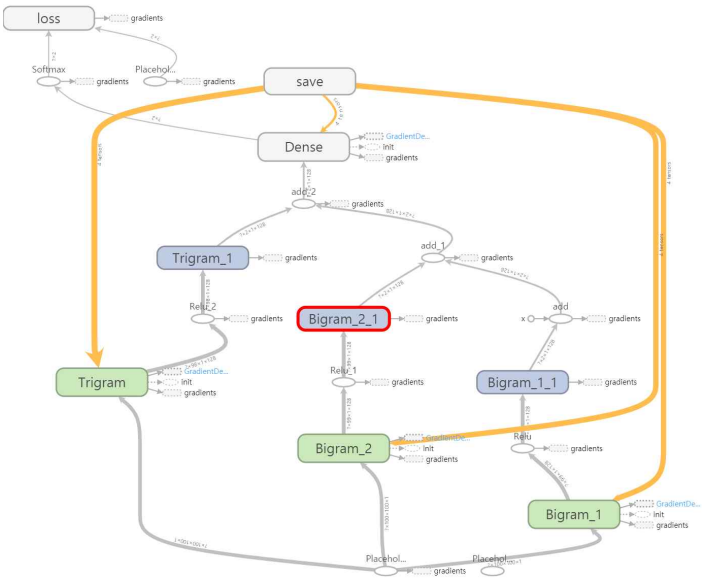
2018 강서학생탐구발표대회 탐구보고서

편향된 말뭉치가 전산 언어 처리에 미치는 영향 An Influence of biased corpus on NLP	출품번호
	미기재

출품 부문	공학 및 에너지



2018. 07 . 18.

소 속 청	학 교 명	학 년	성 명
강서교육청	염경중학교	3	강준서

목 차

Ⅰ. 서론

ⅰ) 개요

‘나무는 열매를 보면 알고, 사람은 그 언행을 보면 안다’라는 속담이 있다. 이 속담이 시사하듯, 사람의 언행을 보면 그 사람이 평소에 듣는 언어 환경을 유추할 수 있다. 청학동 선비와 뉴욕 할렘 가의 불량배가 말을 이해하는 방법이 다르고, 말을 만들어내는 방법이 다르다. 이는 사람을 둘러싼 언어 환경이 언어 발달에 큰 영향을 미치기 때문이다(Hammer, Farkas & Maczuga, 2010).

현재 인공지능을 이용한 언어 처리 분야가 발전하는 데 가장 큰 도움을 준 연구는, 문맥적인 정보를 보존하며 단어를 전산화하는 기법(Tomas Mikolov; 2013)의 발명이다. 인공지능은 거대한 말뭉치(Corpus)에서 비지도 학습의 형태로 문장들에서 스스로 문맥적 정보를 추출하여 단어를 표현한다. 이는 어린아이가 부모의 말에서 놀랍도록 빠른 속도로 스스로 문맥을 이해하며 말을 배우는 것과 유사하다.

따라서, 인공지능은 언어를 인식하고, 생성하는 데 말뭉치의 영향을 받게 된다. 본 탐구는 이 부분에서 문제를 제기한다. *말뭉치가 편향되면, 인공지능도 한 쪽으로 치우치게 판단할까?* 현재 인공지능은 추론 단계까지 나아가지 못하고, 주어진 입력에 반응하여 출력을 내놓는 수준에 불과하다. 즉, 딥 러닝 알고리즘을 위시한 여러 머신러닝 알고리즘은 귀납적으로 판단을 하게 되는데, 이 때문에 편향된 데이터셋에서 보편적인 규칙을 이끌어내기 힘들다. 이 때문에 ‘대통령’을 ‘Mr. President’ 으로 번역하거나(이는 영미권에서 여성 대통령이 극히 소수였기 때문에 문장에서 대부분 Mr. President로 표기했기 때문이다) ‘흑인 여성’의 얼굴을 제대로 인식하지 못하기도 한다(얼굴 데이터셋의 대부분이 백인이나 아시아인 남성, 여성이기 때문이다). 문제는, 기존의 데이터는 백인 위주의, 혹은 남성 위주로 편향되어 있다. 이런 결함으로 인해, 인공지능의 정치적, 사회적 중립성은 인공지능이 앞으로 넘어야 할 하나의 과제 중 하나이다.

(그림: 편향된 인공지능의 예, 마이크로소프트 테이)

이에 본 탐구에서는 언어 처리 인공지능의 차별적인 사고를 줄이기 위해 임베딩 과정에서 말뭉치가 편향되었을 때 인공지능 또한 편향되는지 실험하고, 이런 편향을 줄일 수 있는 방법을 찾아내고 실험한다.

ii) 탐구의 목적과 동기

초등학교 6학년 무렵, 어머니께서 보육교사 자격증 시험을 치르기 위해 집에 있던 컴퓨터로 관련 자료들을 공부하고 계셨었다. 자연스레 나 또한 옆에서 인터넷으로 관련된 강의를 들을 기회가 있었고, 가장 관심을 끌었던 것은 인지발달론 이였다. 그리하여 아동의 인지발달과 언어발달에 대해 가볍게 배울 기회가 있었고, 영유아가 언어 환경에서 스스로 말을 깨우친다는 것은 나에게 깊은 인상을 남겼다.

그리고 올해 여러 대회들을 거치며 딥 러닝에 대해 더 깊이 공부하고 싶었고, 자연스레 자연어 처리 분야에 눈을 돌렸다. 여러 이론들을 공부하며 자연어 처리에 대한 언어학적인 배경이 궁금하게 되었고, 독서 시간을 통해 도서관에서 〈언어의 진화: 최초의 언어를 찾아서〉라는 책을 찾아 읽게 되었다. 그러던 중 이전에 배웠던 아동의 언어발달과 관련된 내용을 읽고 사람마다 다른 가치관이 형성되는 것이 아동기의 언어 환경의 영향을 크게 받지 않을까 생각하게 되었다.

또한 〈시사in〉이라는 잡지에서 평소 존경하던 조경현 교수님의 인터뷰를 보게 되었는데, 인공지능이 학습 데이터의 편향을 그대로 학습한다는 언급이 있었다. 이에 데이터셋의 편향 뿐 아니라, 전산 언어 처리에서 활용하는 도구; 말뭉치의 편향도 인공지능에 영향을 주지 않을까 생각했다.

그렇게 이전의 경험들을 토대로, 인공지능이 편향된 말뭉치를 이용하여 학습하면, 그 출력도 말뭉치의 경향을 따라갈 것이라 가설을 설립했다. 말뭉치는 인공지능이 단어의 문맥적 의미를 학습하는 문장들의 집합이므로, 유아가 언어 발달에 그 환경의 영향을 받듯이 인공지능도 똑같이 귀납적 추론으로 단어의 의미를 학습하기 때문이다.

이에 본 탐구는 말뭉치의 편향을 인공지능이 그대로 따르는지 두 차례의 실험을 통해 검증하고, 이 편향이 전산 언어 처리 서비스에 미치는 영향과 이를 이용할 수 있는 방안에 대해 탐구해보도록 한다.

iii) 선행 연구

1) Word Embedding

자연어(정제되지 않은 언어)를 인공지능으로서 처리할 수 있게 하려면, 문자를 수(數)로 나타내야 한다. 이를 인코딩이라 하는데, Word Embedding은 One-Hot 인코딩된 단어를 저차원의 벡터로 나타내는 기법이다. 대표적으로 Word2Vec, GloVE 등이 있으며, 빈도수 기반으로 단어를 약 100 차원의 벡터로 표현한다.

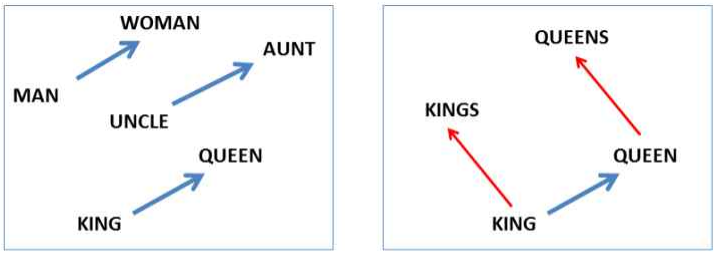
One-Hot 인코딩은 해당되는 요소만 1의 값을 갖고, 나머지는 전부 0으로 채우는 인코딩 기법이다. 예를 들어, 인코딩할 단어들의 정의역이 {강아지, 지우개, 사탕}이고, 사탕을 표현하고 싶다면 One-Hot 인코딩은 '사탕'이라는 단어를 [0, 0, 1]로 나타낸다.

	A	B	C	D	E	F	G	H	I
1	Original data:			One-hot encoding format:					
2	id	Color		id	White	Red	Black	Purple	Gold
3	1	White		1	1	0	0	0	0
4	2	Red		2	0	1	0	0	0
5	3	Black		3	0	0	1	0	0
6	4	Purple		4	0	0	0	1	0
7	5	Gold		5	0	0	0	0	1
8									

(그림: One Hot 인코딩)

하지만 One Hot 인코딩은 많은 문제점을 보유하고 있다. 위의 예시에서는 정의역의 원소 수가 3개 뿐 이였지만, 실제 상황에선 처리해야 할 단어들이 수십만 개에 달하기 때문에, One Hot 인코딩된 벡터를 신경망의 입력 값으로 사용하면 '차원의 저주'라는 현상, 즉 자유도[1]가 필요 이상으로 높아지게 된다. 또한 계산량이 폭발적으로 증가하며 데이터 간의 상관관계를 표현할 수 없다. 즉, One Hot 인코딩의 경우 어떤 사전에서 127째에 위치한 '돼지'라는 단어는 128번째에 있는 '남아프리카 콩고민주공화국'이란 단어와 가깝지만 23459번째에 있는 '포유류'란 단어와 어떠한 의미 관계도 갖지 않게 된다.

이 문제를 해결하기 위해 등장한 방법이 Word Embedding 이다. 본 탐구에서는 Word2Vec을 중점적으로 다루겠다. Word2Vec은 거대한 말뭉치(Corpus)에서 문맥을 통해 단어의 문맥적 정보를 파악하고, 이를 저차원의 벡터로 표현하는 방법론이다. 예를 들어, word2vec은 '돼지'라는 단어를 [0.782, -4.32, 4.55, ... , 0.01, 0.042]와 같은 벡터로 나타낼 수 있다. 아래는 word2vec의 특성을 잘 나타내는 도해이다.



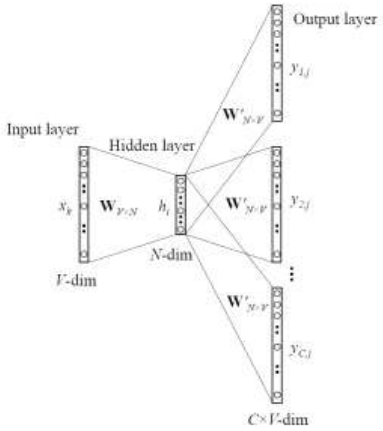
(Mikolov et al., NAACL HLT, 2013)

1) 연립방정식에서 미지수의 개수가 많아지면 더 많은 식이 있어야 풀 수 있는 현상과 유사하다.

위의 도해에서, Man-Woman 간의 관계와 Uncle-Aunt 간의 관계가 유사하고, 동시에 King-Queen간의 관계 또한 유사하다. 이는 문장에서 문맥을 통해 단어의 의미관계를 파악하는 Word2Vec의 특성 때문이며, 이를 통해 의미를 기반으로 하는 자연어 처리 시스템이 등장할 수 있게 되었다.

Word2Vec은 중심 단어와 주변 단어와의 관계를 통해 벡터공간에 단어를 매핑하는 Skip-Gram과 거꾸로 주변 단어와 중심 단어와의 관계를 통해 매핑하는 CBoW로 학습 방식이 나뉜다. 근래는 Skip-Gram이 더 많이 쓰이기 때문에, 본 보고서에선 Skip Gram만을 선행 연구로서 언급하겠다.

2) Skip Gram



Skip-Gram 모형

사람을 알기 위해선 그 친구를 살피라고 하였다. Skip Gram은 문맥상 비슷한 위치에 등장하는 단어는 의미가 유사할 것이라는 가정을 토대로 세워진다. 따라서, 언어의 문맥적 특성을 보존하기 위해 단어 주변에 어떤 단어가 올지 예측하는 모델을 세우고 목적함수를 극대화하여 언어의 문맥상 의미를 보존한다.

Skip Gram에서 사용하는 목적함수는 다음과 같다: $p(o|c) = \frac{\exp(u_o^T v_c)}{\sum_{w=1}^{W} \exp(u_w^T v_c)}$ 2)

즉, 중심단어 c가 주어졌을 때 그 주변단어 o가 등장할 조건부확률을 극대화하는 것이 목표이고, optimization 과정에서 구하게 된 단어벡터인 v를 임베딩된 벡터로서 사용한다. 하지만, 학습이 잘 되어 있다는 가정 하에서 u를 사용해도 큰 문제는 없다. 결론적으로, Skip Gram 모델은 아래와 같은 1개의 Dense Layer와 1개의 Softmax Layer로 이루어진 신경망을 최적화하는 것으로 볼 수 있다. (즉, 이 때, 학습 데이터는 (중심 단어, 주변 단어)의 쌍으로 이루어진다)

따라서, 단어 임베딩을 거치면 문맥적 의미가 보존된 채로 단어를 n차원 벡터로 나타낼 수 있다.

2) v_c는 중심단어의 벡터, u_o는 주변단어의 벡터

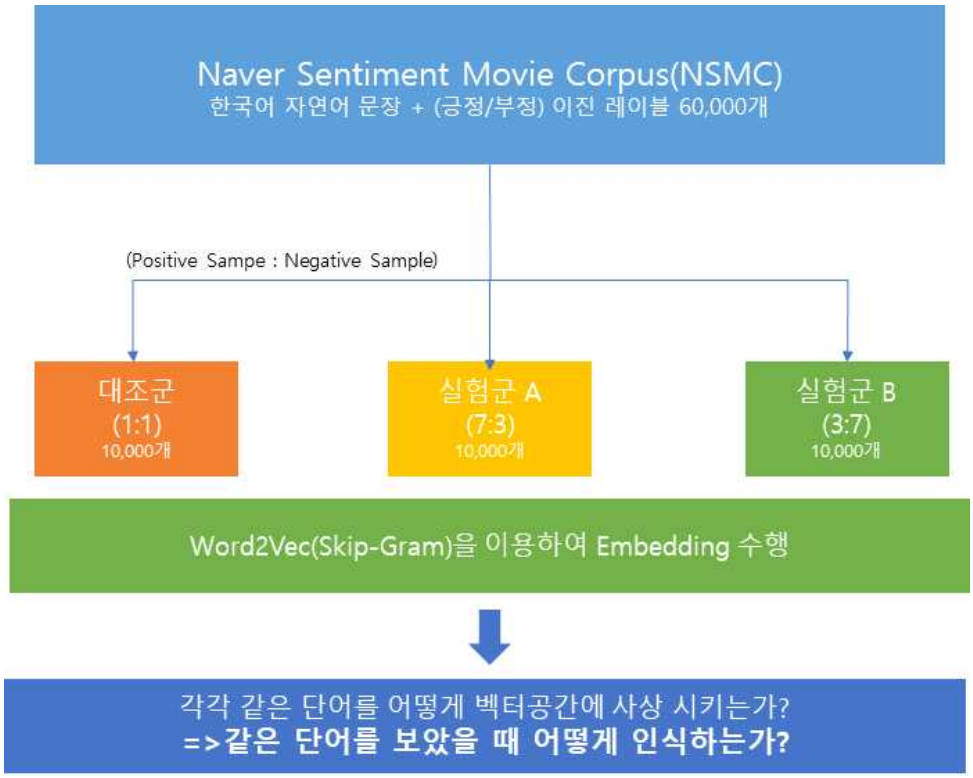
Ⅱ. 본론 1: 편향된 말뭉치가 임베딩에 미치는 영향

ⅰ. 가설 설정

Positive Label로 편향된 말뭉치를 임베딩했을 때, 편향되지 않은 말뭉치로 임베딩한 모델보다 중의적인 단어와 가까운 단어들에 긍정적인 단어가 많을 것이고, 반대의 경우에도 그럴 것이다.

위의 가설의 근거는 이렇다: 임베딩 과정에서 사람이 인식할 수 없지만 문장에서 단어가 긍정적으로 사용되었는지 혹은 부정적으로 사용되었는지에 대한 정보가 문맥적인 정보가 보존되면서 연속 벡터공간에 사상될 것이므로, 해당 단어가 긍정적으로 사용된 빈도가 높을수록 긍정적으로 사용된 단어들과 거리가 가까워질 것이고 반대의 경우도 그러할 것이다.

ⅱ. 탐구 설계



사용 말뭉치	NSMC Corpus(네이버 D2 Lab 제공, 영화 리뷰)
추출할 데이터 양	150,000개의 샘플 중 무작위로 50,000개씩 3세트를 추출
Pos Tagger	Twitter Postagger
Embedding 방법	Word2Vec - Skip Gram

본 탐구에서는 세종 꼬꼬마 형태소 분석기를 이용하여 품사 태깅된 말뭉치에서, 1개의 대조군, 2개의 실험군에 각각 Postive와 Negative 샘플의 비율을 달리하여 1만개의 문장을 추출하고, 이를 학습 데이터셋으로 임베딩을 학습시킨다. 이를 통해 얻어낸 단어장의 Lookup Table을 토대로 긍정·부정적으로 동시에 해석될 수 있는 단어 4개에 대해 가장 가까운 공간에(문맥상 비슷한 정보를 갖는) 있는 5개의 단어를 계산한다. 이를 통해서 편향된 말뭉치를 사용하여 단어를 사상시켰을 때, 인공지능이 단어를 어떻게 인식하는지 주변 단어를 통해 간접적으로 도출한다.

iii. 탐구 수행

[코드 저장소] 코드 별첨.

사용 언어	Python 3.6(with JPype)
사용 외부 패키지	Gensim(word2vec 모델), Konlpy.tag.twitter(트위터 형태소 분석기)
말뭉치	NSMC(from Naver D2 Lab)
모델	word2vec(Skip-Gram)
실험군 개수	5개(각각 긍정, 부정 샘플의 비율을 달리 함)

차수	실험 내용
1차~4차	Pos Tagger로 서울대학교 연구진의 꼬꼬마 포스태거(kkma)를 사용. *jpype._jexception.OutOfMemoryErrorPyRaisable: java.lang.OutOfMemoryError: Java heap space* 메모리 관련 오류 발생, konlpy 이슈 검색 및 구글링 결과 꼬꼬마 포스태거는 대용량 데이터 처리에는 적합하지 않음을 인지, 메모리 할당을 늘렸으나 1400 문장을 넘어가니 속도가 급격히 느려짐, 포스태거를 트위터 포스태거로 변경
5차~7차	Gensim 패키지 사용 미숙으로 인한 문법 오류 발생, 패키지 문서를 읽고 수정 비교할 단어 목록 수정(말뭉치에 존재하지 않는 단어 존재: '신파'를 '초딩'으로 수정)
8차	실험은 성공적이었으나, 결론을 도출하기에 부족함, 실험군을 2개 추가(a_2, b_2)하고 사용하는 샘플의 양을 5배 늘림(10,000->50,000)
	대조군의 결과 1. 초딩와 가까운 단어: 고등학생, 나오는구나, 고딩, 초등학생, 본거 2. 반전와 가까운 단어: 긴장감, 스릴, 결말, 지루함, 교훈 3. 개와 가까운 단어: 개도, 점도, d, 류, 박자 4. 매력와 가까운 단어: 맥락, 교훈, 내게는, 해변, 배경
	실험군 a의 결과 1. 초딩과 가까운 단어: 중딩, 중학교, 초등학생, 고등학교, 초등학교 2. 반전와 가까운 단어: 긴장감, 스릴, 씬, 엔딩, 부분 3. 개와 가까운 단어: 점주, 시즌, 편다, 광구, 점줌 4. 매력과 가까운 단어: 긴장감, 교훈, 느껴졌습니다, 환상, 따뜻함을
	실험군 b의 결과 1. 초딩과 가까운 단어: 좀비, 좌파, 멍청이, ㅠㅅ, 독립영화 2. 반전과 가까운 단어: 긴장감, 결말, 스릴, 교훈, 지루함 3. 개와 가까운 단어: 류, 로더, 점주, 씹, 론데 4. 매력와 가까운 단어: 사실, 감, 현실, 교훈, 음악
9차	실험 결과 하단 첨부, 결론 도출

① Positive Label : Negative Label이 1:1인 경우(대조군)

단어	유의어
초딩	고등학생, 나오는구나, 고딩, 초등학생, 본거
반전	긴장감, 스릴, 결말, 지루함, 교훈
개	개도, 점도, d, 류, 박자
매력	맥락, 교훈, 내게는, 해변, 배경

② Positive Label : Negative Label이 9:1인 경우 (실험군 A_1)

단어	유의어
초딩	중딩, 중학교, 초등학생, 고등학교, 초등학교
반전	긴장감, 스릴, 씬, 엔딩, 부분
개	점주, 시즌, 편다, 광구, 점줌
매력	긴장감, 교훈, 느껴졌습니다, 환상, 따뜻함을

③ Positive Label : Negative Label이 7:3인 경우(실험군 A_2)

단어	유의어
초딩	고딩, 유치원, 시상식, 중학교, 좋아하던
반전	긴장감, 결말, 스릴, 지루함, 설정
개	점도, 점주, 점임, 기, 점준
매력	교훈, 개성, 영상, 배경, 음악

④ Positive Label : Negative Label이 3:7인 경우(실험군 B_1)

단어	유의어
초딩	잘만, 보네, 녀석, 거든, 종편
반전	긴장감, 스릴, 씬, 엔딩, 부분
개	탄은, 노잼, 루, 한개, 점주
매력	설정, 교훈, 기술, 현실, 쥐똥

⑤ Positive Label : Negative Label이 1:9인 경우(실험군 B_2)

단어	유의어
초딩	빨갱이, 좌파, 멍청이, ㅆ, 독립영화
반전	긴장감, 스릴, 결말, 교훈, 지루함
개	점임, d, 점도, 노잼, 류
매력	현실, 쥐똥, 교훈, 흐름, 개성

Ⅲ. 본론 2: 편향된 임베딩을 이용한 문장 분류 성능 비교

ⅰ. 가설 설정

편향된 말뭉치로부터 사상된 임베딩 벡터는 편향을 보였다. 그렇다면 이 편향된 벡터를 기반으로 문장 분류 모델을 구축했을 때 모델의 출력 또한 치우쳐진 경향을 띨 것이다.

가설의 근거는 (탐구 1)과 동일하다.

ⅱ. 탐구 설계

문장 분류 신경망을 구축하고, NSMC 데이터셋을 이용하여 학습시킨다. 그 후, (탐구 1)과 동일하게 긍정적 리뷰와 부정적 리뷰가 1:1로 동일하게 구성된 말뭉치로 임베딩한 대조군과, 각각 긍정적 리뷰와 부정적 리뷰의 비율이 1:9와 9:1인 실험군 A와 B를 구성한다. 각각의 실험-대조군에 대해 임베딩을 학습시키고, 따로 준비한 NSMC 테스트 데이터셋을 각각 실험군의 임베딩으로 표현하고 신경망에 입력시켜 편향된 임베딩을 사용해서 문장을 분류했을 때 성능이 어떻게 변하는지를 탐구한다.

사용 데이터셋	NSMC Dataset(네이버 D2 Lab 제공, 영화 리뷰)
추출할 데이터 양	150,000개의 샘플 중 무작위로 10,000개씩 3세트를 추출
Pos Tagger	Twitter Postagger
신경망 모델 구조	하단 그림

문장 분류에 사용할 신경망의 아키텍쳐는 Yoon Kim(2014)의 것을 따른다(약간의 차이가 있음).

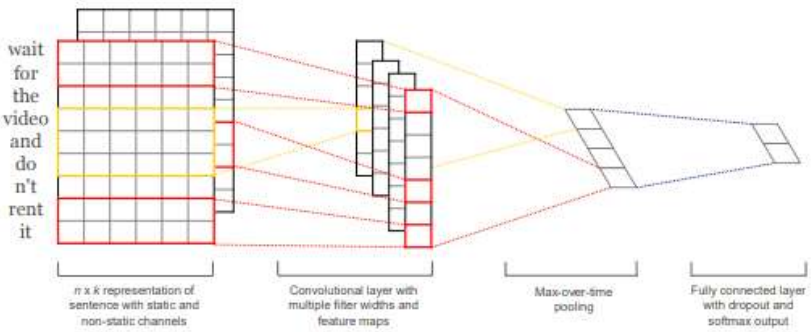
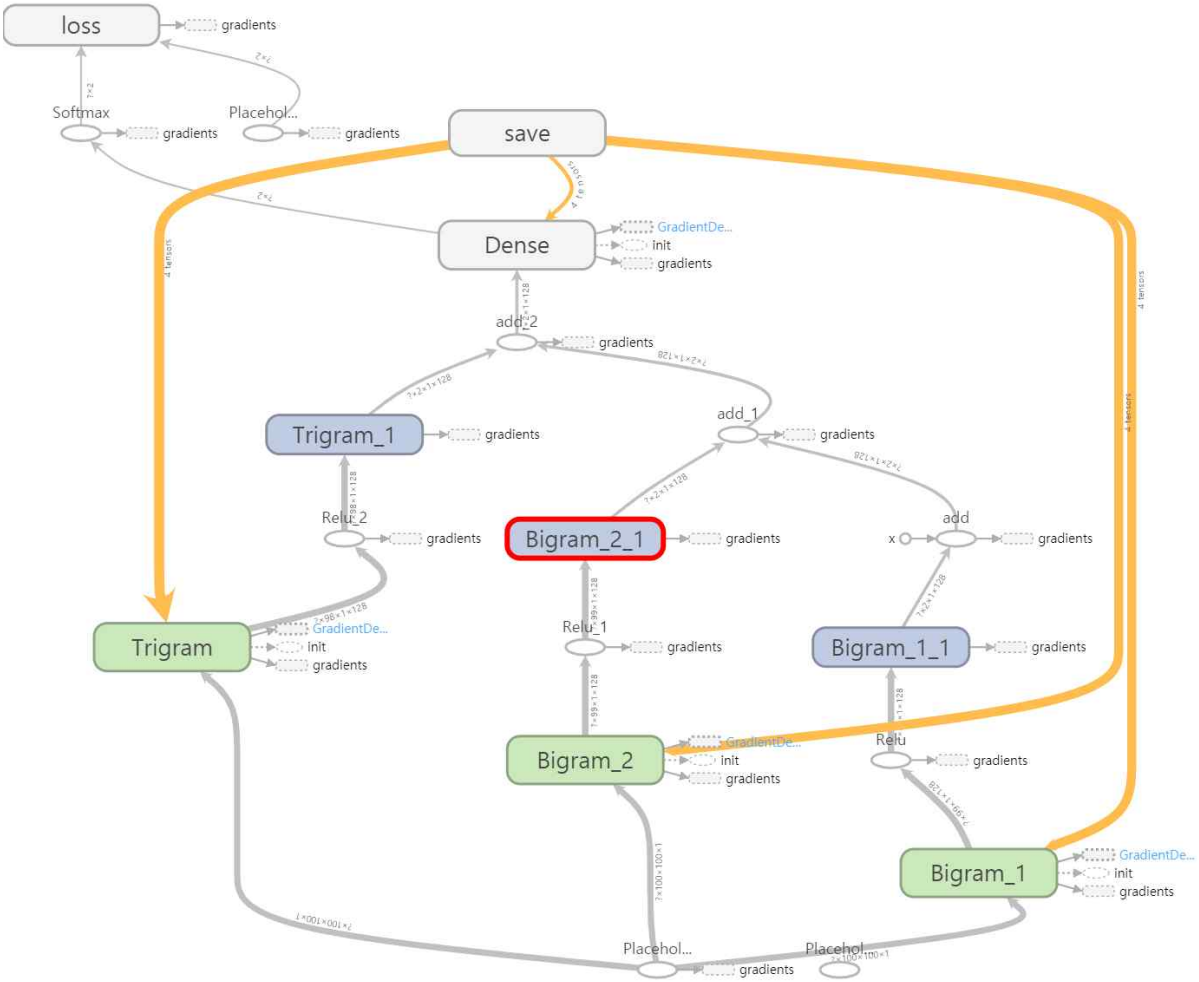


Figure 1: Model architecture with two channels for an example sentence.

임베딩 레이어	본 탐구에서는 편향된 말뭉치가 미치는 영향을 조사하기 위해 사전 임베딩하여 신경망에 입력한다.
합성곱 레이어	3개의 필터를 사용하고, 각각 크기는 (2, 2, 3)이다. 문장에서 공통적으로 반복되는 특징을 감지한다.
최대 풀링 레이어	불필요한 정보를 줄인다.
소프트맥스 레이어	합성곱, 풀링 레이어에서 감지한 특징을 기반으로 문장을 이진 분류한다.

iii. 탐구 수행



(그림: 실험에 사용한 텐서플로우 그래프-Sentence Classification Model with Convolutional Neural Network)

차수	실험 내용	
1차	모델 구축 직후 테스트, 문법 관련 오류로 수정 후 재실험	
2차	텐서플로우 버전 업데이트로 시각화 관련 변경 사항 적용	
3차	학습 완료. 소요 시간 약 40분(Tensorboard로 시각화)	
4차	위에서 학습한 문장 분류 모델을 통해서 편향된 임베딩으로 표현된 입력 값들로 테스트. 실험 후 데이터 추출	
	1) 대조군 (1:1)	정확도: 87.4%
	2) 실험군 A(1:9)	정확도: 67.6%
	3) 실험군 B(9:1)	정확도 62.8%

Ⅳ. 결론

ⅰ. 데이터 해석 및 결론 도출

① Positive Label : Negative Label이 1:1인 경우(대조군)

단어	유의어
초딩	고등학생, 나오는구나, 고딩, 초등학생, 본거
반전	긴장감, 스릴, 결말, 지루함, 교훈
개	개도, 점도, d, 류, 박자
매력	맥락, 교훈, 내게는, 해변, 배경

예상과는 일치하는 부분도 있었고 다르게 나타나는 부분도 있었다. 우선 예상과 다른 부분은, Positive Sample과 Negative Sample의 비율을 다르게 하면 유의어의 분포가 크게 달라질 것이라 생각했는데 '매력'의 경우 모든 모델에서 공통적으로 '교훈'을 유의어로서 출력했고, '반전' 또한 '긴장감'을 모든 모델에서 유의어로서 출력했다. 이는 영화 리뷰라는 말뭉치의 특성에서 비롯된 것이다. 즉, 많은 문장에서 반전과 긴장감을 함께 언급했다는 말이다(그것이 부정적으로 사용되었건 긍정적으로 사용되었건 간에)

② Positive Label : Negative Label이 9:1인 경우 (실험군 A_1)

단어	유의어
초딩	중딩, 중학교, 초등학생, 고등학교, 초등학교
반전	긴장감, 스릴, 씬, 엔딩, 부분
개	점주, 시즌, 편다, 광구, 점줌
매력	긴장감, 교훈, 느껴졌습니다, 환상, 따뜻함을

실험 군 A_1은 긍정적인 리뷰의 비율이 훨씬 높은 데이터 셋이다. 예상만큼 긍정적인 단어들이 눈에 띄진 않는다. 주로 정보 전달이나 중립적인 스탠스의 단어들이 대부분인데, 이는 대다수의 사람들이 영화가 실제로 수작(秀作)이라 생각하지 않더라도 높은 점수를 주는 데에서 유래된 것으로 보인다. 그럼에도 환상, 따뜻함, 스릴, 따뜻함 등의 긍정적인 단어들이 많이 눈에 띈다.

③ Positive Label : Negative Label이 7:3인 경우(실험군 A_2)

단어	유의어
초딩	고딩, 유치원, 시상식, 중학교, 좋아하던
반전	긴장감, 결말, 스릴, 지루함, 설정
개	점도, 점주, 점임, 기, 점준
매력	교훈, 개성, 영상, 배경, 음악

실험 군 A_2는 A_1보다 중립적이지만 아직 positive하게 편향된 데이터셋이다. A_1에 비해 긍정적이고 영화를 칭찬하는 표현이 줄었지만 대조군에 비해서는 아직 긍정적인 스탠스의 단어들이 많이 분포하고 있다. 초딩: 좋아하던 이나 매력: 개성, 음악 등이 그 예시가 될 수 있다. 그러나, 반전: 지루함과 같은 부정적인 표현도 적지만 존재한다.

④ Positive Label : Negative Label이 3:7인 경우(실험군 B_1)

단어	유의어
초딩	잘만, 보네, 녀석, 거든, 종편
반전	긴장감, 스릴, 씬, 엔딩, 부분
개	탄은, 노잼, 루, 한개, 점주
매력	설정, 교훈, 기술, 현실, 쥐똥

실험군 B_1은 대조군에 비해 Negative한 샘플이 많이 분포되어 있다. 역시 가설 설정 단계에서 예상한 대로 개: 노잼, 매력: 쥐똥 같은 부정적인 스탠스가 강한 단어들이 많이 분포하고 있으며 A 실험 군이나 대조군에서 보여주었던 '초딩'의 유의어 분포가 정보 전달의 목적에서 '초등학생을 낮잡아 부르는 말'로서 쓰이는 용도로 쓰이는 경향을 보인다. (초딩: 녀석, 초딩: 잘만 보네)

⑤ Positive Label : Negative Label이 1:9인 경우(실험군 B_2)

단어	유의어
초딩	빨갱이, 좌파, 멍청이, ㅄ, 독립영화
반전	긴장감, 스릴, 결말, 교훈, 지루함
개	점임, d, 점도, 노잼, 류
매력	현실, 쥐똥, 교훈, 흐름, 개성

실험 군 B_2는 A_1과 마찬가지로 한쪽으로 매우 치우친 말뭉치에서 학습된 모델이다. 초딩: 빨갱이 같은 상식적으로 연상하기 힘든 정도의 부정적인 연상을 하며, 단어들 또한 매우 공격적이다(이는 정치색을 띄는 영화에 대해 노골적이고 원색적인 비난이 많기 때문이라고 예상된다)

데이터를 분석함으로써, **학습하는 말뭉치에 Positive한 경향을 보이는 샘플이 많으면 인공지능 또한 Positive하게 단어를 인식하고, Negative한 샘플이 많으면 Negative하게 인식함**을 증명할 수 있었다. 이는 유아가 긍정적인 말을 많이 듣고 언어를 습득했을 때 긍정적인 사고를 갖는 경향을 띄고, 부정적이고 폭력적인 말을 듣고 자랐을 때, 가치관이 부정적이고 폭력적인 경향을 띠는 것과 유사하다.

1) 대조군 (1:1)	**정확도: 87.4%**
2) 실험군 A(1:9)	**정확도: 67.6%**
3) 실험군 B(9:1)	**정확도 62.8%**

또한, 위와 같이 편향적으로 임베딩 된 단어 벡터들을 입력 값으로 문장 분류 모델을 구축하였을 때, 긍정적으로 편향된 모델과 부정적으로 편향된 모델 모두 정확도가 떨어지는 결과를 낳았으며, 별점 4점 이상의 리뷰도 부정적인 리뷰로 인식하거나, 반대로 부정적으로 편향된 경우 별점 2점 이하의 리뷰도 긍정적인 리뷰로 판단하는 등 말뭉치에 존재한 편향을 증폭시켰다.

따라서, 말뭉치에 편향이 존재하면, 인공지능은 이를 그대로 학습하고 증폭시켜 실 동작에 그 경향을 그대로 드러낸다. 이는 산업 인공지능이나 챗봇 같은 대인 서비스나, 게임업계에서 사용하는 비속어 감지 인공지능에 악영향을 줄 수 있다(과하게 감지할 수 있으므로)

ii. 향후 사용 방안 및 발전 가능성

일련의 실험들을 통해, 인공지능 또한 **편향된 정보로 학습을 했을 경우에 문장을 인식하고 분류할 때 그 경향을 따라간다**는 결론을 도출할 수 있었다.

본 탐구의 결과는, 말뭉치를 구성할 때, 어느 한쪽으로 편향되게 문장들을 구성했을 때 중립적이지 않은 스탠스로 문장을 왜곡할 수 있는 위험성을 보여준다. (탐구 2)에서는 직관적인 테스트 정확도를 기준으로 이 위험성을 드러냈지만, 실제 서비스의 경우 서론에서 언급한 마이크로소프트의 챗봇 '테이'와 마찬가지로 인종차별적인 언행을 하거나 정보를 처리함에 있어서 차별과 왜곡이 섞일 수 있다. 따랏서, 사람과 마찬가지로, 인공지능 모델 또한 언어 환경에 따라서 인식하고 사용하는 언어가 왜곡될 수 있으며, 인공지능 서비스 개발자, 엔지니어 혹은 제공자는 **효율적인 문제 해결이나 원만한 자연어 처리 서비스 제공을 위해선 말뭉치가 편향되지 않아야 함을 인지**해야 할 것이다.

또한 이런 편향을 줄이기 위해선, 말뭉치에 들어가는 문장의 개수를 늘리는 방법이 있다. 하지만 이 방법은 모집단이 편향되지 않아야 한다. 예를 들어 캘리포니아대에서 미국의 과격 사이트인 4chan에서 수집한 데이터를 토대로 챗봇을 학습시켰을 때, 서론의 '테이'와 같은 폭력적인 언행을 보이기도 하였다. 하지만 문장을 추출할 모집단의 형질(긍정-부정, 정치색, 과학적 견해 등)들의 비율이 고르게 분포되어 있다면 문장을 추출할수록 말뭉치의 비율도 고르게 분포될 것이기 때문에 이 경우에는 유효한 해결책이 될 수 있다. 또한 임베딩을 레이어의 형태로 네트워크 내부에 포함하고, 정규화 레이어를 추가하는 방법이 존재한다. 이 경우 총 연산의 양은 늘어나나 편향이 존재해도 이에 제약을 둘 수 있는 이점이 있다.

또한 이를 역이용할 수 있다. 본 탐구의 결과는, 말뭉치의 구성을 조정함으로서 인공지능에게 의도적으로 주관을 심을 수 있음을 증명했고, 실제로 (탐구 1)에서 긍정적인 리뷰와 부정적인 리뷰의 비율을 수정하면서 인공지능을 의도적으로 긍정적 또는 부정적으로 주관을 심었다. 이는 각각의 실험군들이 같은 단어를 어떻게 다르게 인식하는지 벡터공간에서 가장 가까이 있는(문맥적으로 의미가 유사하다고 판단한) 단어들로 간접적으로 도출할 수 있었다. 따라서, 이를 이용하여 의도적으로 긍정적으로 인공지능을 편향되게 학습시킨 후, 언어 생성 모델과 결합하여 '선플 다는 인공지능'이나 '위로해주는 인공지능'을 학습시킬 수 있다. 하지만 이를 악용하면 상대 진영에 대해 악의적인 자료들로 학습시킨 인공지능으로 '제 2의 댓글부대'를 만드는 등 정치적으로 사용될 수 있기에 이에 대한 대책은 필요할 것이다.

결론적으로, 편향된 말뭉치는 인공지능에 의미 왜곡이나 잘못된 주관을 심어줄 수 있다. 따라서 말뭉치가 편향되지 않게 데이터 전 처리할 필요가 있으나, 이를 역이용하면 의도적으로 언어의 의미 해석과 언어 생성에 있어 의도적으로 주관을 심은 인공지능을 학습시킬 수 있다.

Ⅳ. 참고문헌 및 논문

ⅰ. 참고문헌

https://ratsgo.github.io/from%20frequency%20to%20semantics/2017/03/30/word2vec/
Word2Vec의 학습 방식

https://ratsgo.github.io/from%20frequency%20to%20semantics/2017/05/10/postag/
포스태거 성능 비교

http://kkma.snu.ac.kr/
꼬꼬마 세종 포스태거

ⅱ. 참고논문

Efficient Estimation of Word Representations in Vector Space, Tomas Mikolov 外 3인

Convolutional Neural Networks for Sentence Classification, Yoon Kim