**한국어의 문법적 특성을 반영한**

**사전학습 언어모형 구축**

**톡위드 ( 강혜빈, 구채원, 조소연)**

**통계학과 201911501 강혜빈**

**통계학과 201911503 구채원**

**통계학과 201911536 조소연**

**부산대학교 정보컴퓨터공학부**

**2022후기 착수보고서**

**2022년 5월 13일**

**지도교수 : 권 혁 철**

**<목차>**

1. **과제의 배경 및 목표**

1.1 과제 배경 3

1.2 과제 목표 3

1.3 과제 예시 4

1. **요구 조건 분석** 5

2.1 데이터 수집 5

2.2 데이터 전처리 5

2.3 BERT 모델 fine tuning 5

1. **현실적 제약 사항 분석 결과 및 대책** 6

3.1 제약사항 6

3.2 대책 6

1. **설계 문서** 7

4.1 기술 소개 7

4.2 개발 환경 15

4.3 시스템 구성도 15

1. **추진 체계 및 일정** 15
2. **구성원 역할 분담** 16
3. **참고문헌** 16

**<그림 차례>**

**그림 1 희소표현과 밀집표현**

**그림 2 Word2vec**

**그림 3 Self Attention, 계산과정**

**그림 4 트랜스포머 모델**

**그림 5 BERT 임베딩**

**그림6 BERT 토큰화**

**1. 과제 배경 및 목표**

# **1.1 과제 배경**

자연어처리 분야에서는 다양한 응용 분야에 대한 태깅된 학습 데이터 구축 비용 및 시간이 딥러닝 적용에 있어서 장벽으로 가로막고 있는 현실 때문에 단순히 텍스트 데이터를 수집하여 학습 데이터로 활용하는 사전 학습(pre-training)이 이를 극복하기 위한 방법으로 주목을 받는다.

자연어 처리 분야는 인간이 겪는 언어 장벽과 비슷하게 한국어에 적용하려고 할 경우, 영어권 중심의 사전 학습 모델을 그대로 적용할 수 없기 때문에 한국어에 대해 사전 학습 과정을 거쳐야 한다.

한국어는 영어와 달리 조사가 존재하는데, 현재 배포된 사전학습 언어모델은 영어의 언어 단위 구성방법(Tokenizer)를 사용하기에 조사의 구조적 역할을 인식하지 못하여 구문분석이 온전히 이루어지지 못한다.

이를 해결하기 위해 형태소 단위의 사전학습 언어모델을 개발하고자 한다.

# **1.2 과제목표**

한국어 의존 구문분석에 특화된 사전학습 언어모형 구축

* BERT 모델과 Self Attention을 이용한 사전학습 언어모형 구축
* BERT 모델: ‘은’,’는’,’이’,’가’ 등의 조사의 구조적 역할을 반영할 수 있도록 형태소 단위의 토큰을 사용하여 강화 예정
* Self Attention: 한국어의 특성을 기반으로 각 토큰이 주위 토큰들과 가지는 관계 및 중요성을 학습하여 가중치를 부여한다. 이를 통해 구문의 중의성을 해소할 수 있도록 함

# **1.3 과제예시**

예를 들어, "백설공주가 독이 든 사과를 먹었다."라는 문장의 경우,

“백설공주/-가/독/-이/든(들-+-ㄴ) 사과/-를/먹-/-었-/-다.”

“백설공주가”는 조사 “가”를 통해 “든"이라는 서술어와 의존관계를 가지지

않으며, "먹었다."의 주격 명사구임을 인식하고, "독이 든 사과를"은 조사

“를”으로 하나의 명사구로 "먹었다."의 목적격임을 인식하여 문장의 구조적

중의성을 해소한다.

**2. 요구 조건 분석**

# **2.1 데이터 수집**

-vocabulary를 만들기 위한 원시말뭉치 수집

# **2.2 데이터 전처리**

-일부 특수문자, 한자 등 noise 제거 후 한국어 문장 분리기 사용

-higgingface의 transformers 라이브러리를 사용하여 Tokenizer

# **BERT 모델 fine tuning**

-한국어 구문의 구조적 중의성 해소에 특화된 모형 구축

# **3 현실적 제약 사항 분석 결과 및 대책**

# **3.1 제약사항**

1) 저작권 문제

- 원시 말뭉치 수집과정에서 뉴스기사를 크롤링, 2차가공하는 경우 저작권에 위배가 될 수 있다.

2) 컴퓨터 성능

-모형을 구축하는 과정에서 고성능의 GPU가 필요하다.

# **3.2 대책**

1. 제약사항 1의 대책

* 모두의 말뭉치를 주로 사용하고, 당장은 배포를 하지 않는 방향으로 졸업과제를 마무리한다.

1. 제약사항 2의 대책

* 부산대학교 인공지능 연구실의 컴퓨터로 모형을 구축한다.

# **4. 설계 문서**

# **4.1 기술 소개**

**4-1) 워드 임베딩**

(1) 희소표현과 밀집표현

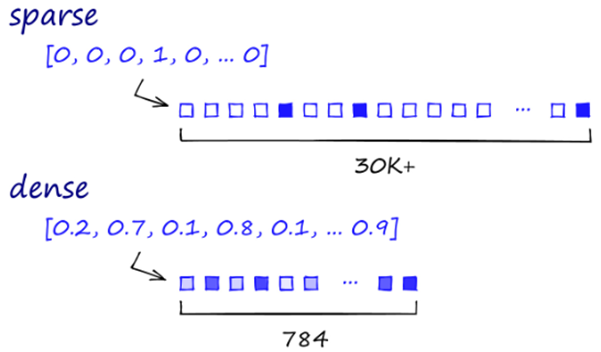
컴퓨터는 인간의 자연어의 의미를 파악하기 어렵기 때문에 단어를 벡터로 표현하여 수치로 나타내는 과정이 필요한데 이러한 벡터 표현에는 희소 표현(Sparse Representation)과 밀집 표현(Dense Representation)이 있다.

-희소표현

희소 표현의 예로는 원-핫 인코딩이 있다. 이는 표현하고자 하는 단어의 인덱스의 값만 1이고, 나머지 인덱스에는 전부 0으로 표현되는 벡터 표현 방법이다. 하지만 단어의 개수가 늘어나면 벡터의 차원이 한없이 커지고 단어 벡터 간의 유사도를 계산할 수 없는 등의 단점이 있다.

-밀집표현

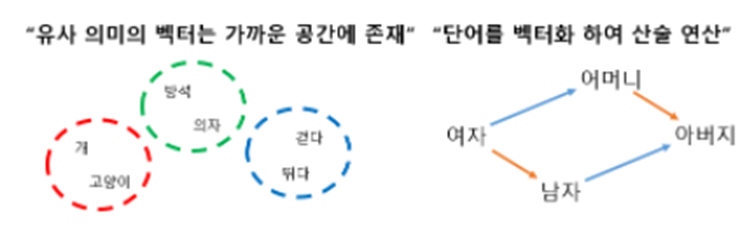
이와 반대되는 개념으로 밀집표현에서는 사용자가 설정한 값으로 모든 단어의 벡터 표현의 차원을 맞추고 벡터의 원소가 0과 1 외에도 다양한 실수 값을 가지게 된다. 이 경우에는 벡터의 차원을 줄이고 단어 벡터 간의 유사도를 계산할 수 있는 장점이 있다. 이러한 밀집 표현이 바로 워드 임베딩(Word Embedding)이다.



(2) 워드투벡터\_유사도 계산

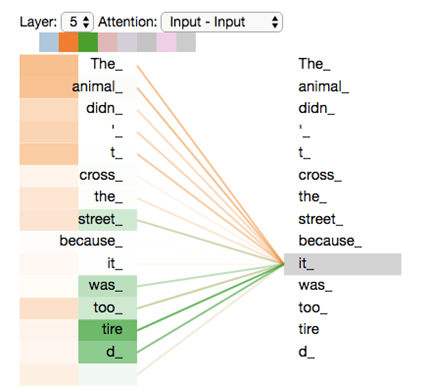
희소표현에서 단어 벡터 간의 유사도를 계산할 수 없는 단점을 개선하기 위하여 사용되는 대표정 방법으로 워드투벡터(Word2Vec)가 있다.

워드투벡터는 주로 함께 등장하는 단어들 사이의 벡터값을 유사하게 설정하는 방식을 사용하여 유사도를 계산한다. 그 결과 고차원에서 저차원으로 **단어의 의미를 여러 차원에다가 분산**하여 표현할 수 있게 된다.



**4-2) Self Attention**

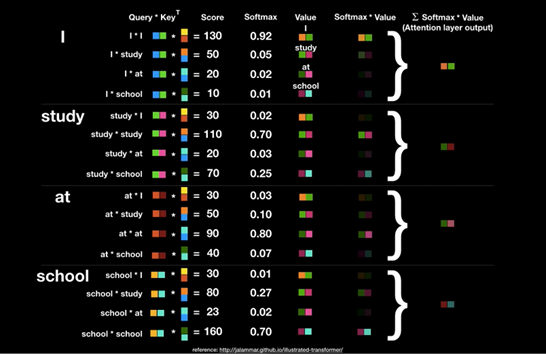
(1) 개요

****

Self Attention은 Transformer의 주요 매커니즘으로, 입력한 문장(context) 내의 각 단어 간의 Attention, 연관성을 이용하여 문장 내의 다른 위치에 있는 단어들의 힌트를 받아 현재 타겟 단어를 더 잘 인코딩할 수 있게 하는 과정이다. 네트워크가 주변 문맥의 단어를 참고하고, 어떤 단어와 연관성이 있으며 Attention이 강한지 결정할 수 있도록 학습한다.

위의 그림을 번역하면 '그 동물은 길을 건너지 않았다. 왜냐하면 그것은 너무 피곤하였기 때문이다.' 라는 의미가 된다. 우리는 피곤한 주체가 동물이라는 것을 아주 쉽게 알 수 있지만 기계는 그렇지 않다. 하지만 셀프 어텐션은 입력 문장 내의 단어들끼리 유사도를 구하므로서 그것(it)이 동물(animal)과 연관되었을 확률이 높다는 것을 찾아낸다.

(2)계산과정



각각의 토큰으로부터 세 벡터를 생성한다.

- Query vector (각 단어에 대한 가중치)

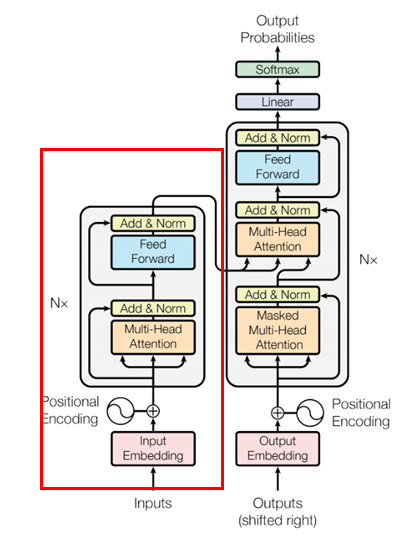
- Key vector (단어가 Query와 얼마나 연관되었는지 비교하는 가중치)

- Value vector (의미에 대한 가중치)

Query 벡터와 Key 벡터를 dot product 연산으로 점수를 매기고, 이 점수를 Key벡터 차원 수의 제곱근으로 나눈 뒤 softmax 확률 값을 얻는다. 각 단어의 Value 벡터에 Softmax 점수를 곱한다. 이 과정을 각 단어마다 해주어 Attention을 얻게 된다.

**4-3) BERT**

(1)개요



BERT는 구글이 공개한 사전 훈련된 모델로 Transformer를 기반으로 하며 모델의 성능에 큰 영향을 미치는 단계인 데이터 임베딩 과정에서 BERT를 사용하며 문맥을 반영한 임베딩을 사용한다.

BERT등장 이전에는 데이터의 전처리 임베딩을 Word2Vec, GloVe, Fasttext 방식을 많이 사용했지만, 요즘의 고성능을 내는 대부분의 모델에서 BERT를 많이 사용한다.

BERT는 레이블이 없는 데이터로 만든 사전 훈련된 모델을 바탕으로 레이블이 있는 다른 작업에서 추가적으로 훈련을 병행하면서 하이퍼파라미터를 재조정한다. 이 과정을 파인 튜닝(fine tuning)이라고 하며 성능이 높게 나온 기존의 사례들을 참고하므로 높은 성능을 가질 수 있다.

(2) BERT의 사전훈련방법

**1) 마스크드 언어 모델(Masked Language Model, MLM)**

문장내 랜덤하게 단어를 마스킹하고 치환하여 이를 예측하는 방식으로 훈련한다.

**원시 말뭉치:**

[CLS]The man went to the store. [SEP] He bought a snack.

**마스킹 및 랜덤 처리:**

15%의 단어들은 [MASK]로 변경, 10% 단어들 랜덤으로 변경

[CLS]The man went to the [MASK]. [SEP] He bought a banana.

**원하는 출력값 :**

[CLS]The man went to the store. [SEP] He bought a snack.

[CLS] 토큰은 BERT가 분류 문제를 풀기 위해 추가된 특별 토큰이고 [SEP] 토큰은 문장이나 데이터를 분리하여 구분하는 용도로 사용한다. [MASK] 토큰은 단어를 마스킹 처리하기 위해 사용한다.

10%의 단어를 랜덤으로 변경하여 토큰이 아닌 단어도 예측하게 하여 모든 단어에 대한 contextual representation이 학습되도록 한다.

**2) 다음 문장 예측(Next Sentence Prediction, NSP)**

: 두 문장이 주어졌을 때, 두 문장의 순서를 예측하는 방식으로 훈련한다.

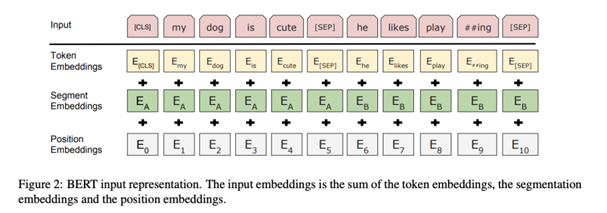
Sentence A: The man went to the store.

Sentence B: He bought a gallon of milk.

Sentence C: Dogs are so cute.

연관이 있으면 label = IsNext, 없다면 label = Not Next

(3) BERT의 임베딩

****

BERT는 3개의 임베딩을 합산한 결과를 취한다.

**①position embedding:** 위치 정보를 학습하기 위한 임베딩

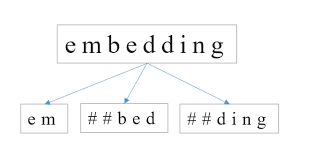
**②segment embedding:** 두 개의 문장을 구분하기 위한 임베딩

**③token embedding:** 실질적인 입력이 되는 워드 임베딩

(4) 토큰화

토큰화란 주어진 코퍼스(corpus)에서 토큰(token)이라 불리는 단위로 나누는 작업을 말한다. 보통 의미 있는 단위로 토큰을 정의하며 단어 토큰화와 문장 토큰화가 있다.

BERT의 서브워드 토크나이저: WordPiece

BERT는 단어보다 더 작은 단위로 쪼개는 WordPiece를 사용한다.

토큰이 단어집합에 존재할 경우 해당 토큰을 분리하지 않고, 존재하지 않을 경우 서브워드로 분리한 뒤 첫번째 서브워드를 제외한 나머지는 앞에 “##”을 붙여 토큰화한다. 기호 표시를 사용하면 원래 단어로 손쉽게 복원이 가능하다.

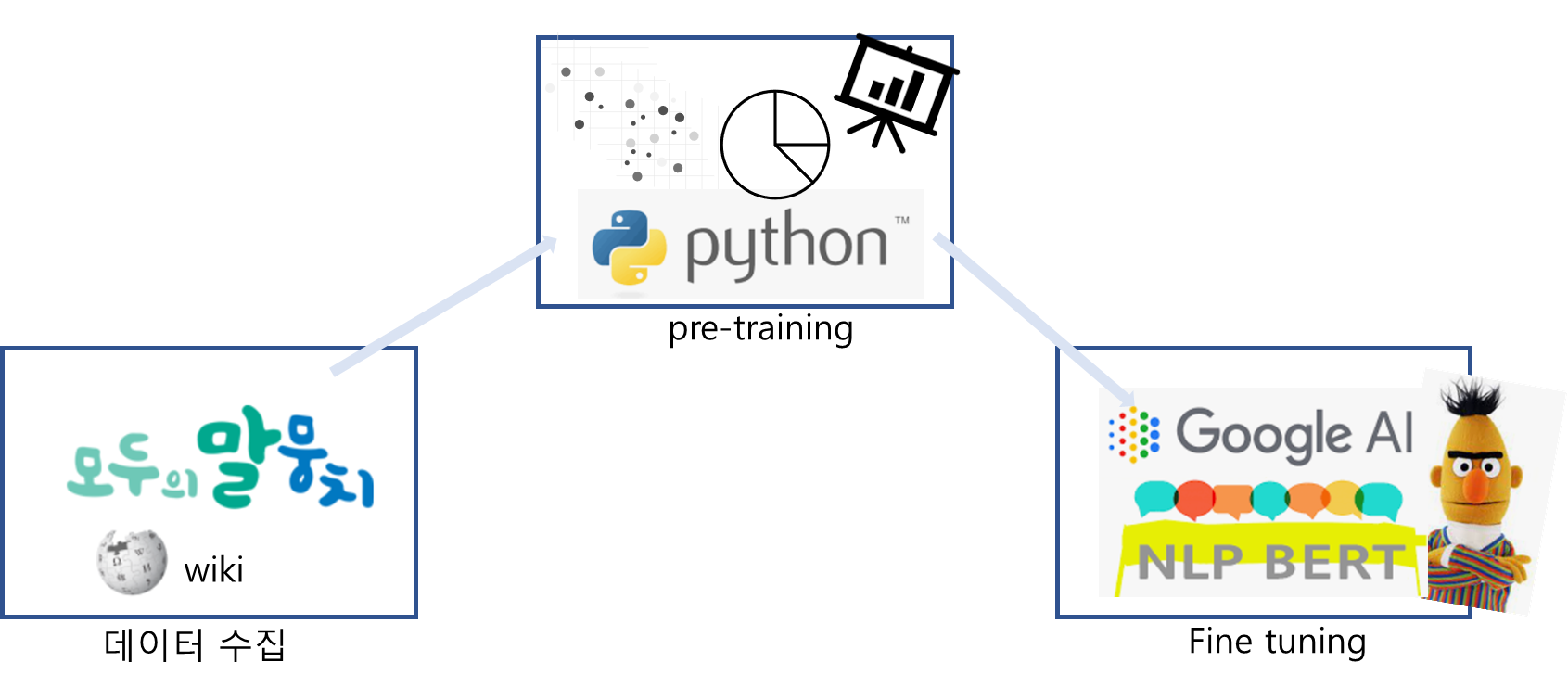
# **4.2 개발 환경**

개발 언어: Python(자연어 처리)

개발 도구: PyCharm, Visual Studio(자연어 처리),

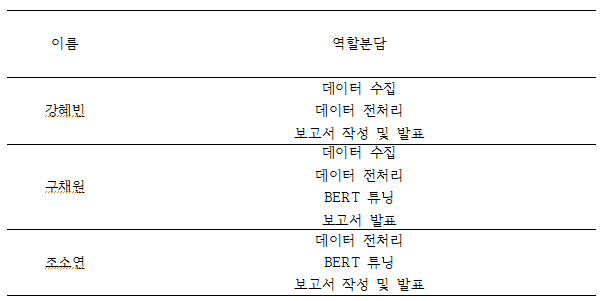
대상 시스템: Windows PC(서버)

# **4.3 시스템 구성도**



# **5. 추진 체계 및 일정**

# **6. 구성원 역할 분담**



# **7.참고문헌**

[1] <https://docs.likejazz.com/bert/>

[2] <https://wikidocs.net/115055>

[3]<https://scienceon.kisti.re.kr/commons/util/originalView.do?cn=CFKO201408355727273&oCn=NPAP12013371&dbt=CFKO&journal=NPRO00371501>

[4]<https://monologg.kr/>

[5]임준호, 김현기 (2020) 사전학습 언어모델의 토큰 단위 문맥 표현을 이용한 한국어 의존 구문분석 (정보과학회논문지 제 48권 제 1호)