어휘 유사 문장 판별을 위한 BERT모델의 학습자료 구축

정재환†, 김동준‡, 이우철‡, 이연수‡

스탠포드 대학교†, ㈜엔씨소프트‡

jaehwanj@stanford.edu, {ncdjk, darkgeo, yeonsoo}@ncsoft.com

Methodology of Developing Train Set for BERT's Sentence Similarity Classification with Lexical Mismatch

Jaehwan Jeong†, Dongjun Kim‡, Woochul Lee‡, Yeonsoo Lee‡

Stanford University†, NCSOFT Corp.‡

요 약

본 논문은 어휘가 비슷한 문장들을 효과적으로 분류하는 BERT 기반 유사 문장 분류기의 학습 자료 구성 방법을 제안한다. 기존의 유사 문장 분류기는 문장의 의미와 상관 없이 각 문장에서 출현한 어휘의 유사도를 기준으로 분류하였다. 이는 학습 자료 내의 유사 문장 쌍들이 유사하지 않은 문장 쌍들보다 어휘 유사도가 높기 때문이다. 따라서, 본 논문은 어휘 유사도가 높은 유사 의미 문장 쌍들과 어휘 유사도가 높지 않은 의미 문장 쌍들을 학습 자료에 추가하여 BERT 유사 문장 분류기를 학습하여 전체 분류 성능을 크게 향상시켰다. 이는 문장의 의미를 결정짓는 단어들과 그렇지 않은 단어들을 유사 문장 분류기가 학습하였기 때문이다. 제안하는 학습 데이터 구축 방법을 기반으로 학습된 BERT 유사 문장 분류기들의 학습된 self-attention weight들을 비교 분석하여 BERT 내부에서 어떤 변화가 발생하였는지 확인하였다.

주제어: BERT, 유사 문장 분류기, 학습 말뭉치 구성,

1. 서론

BERT (Bidirectional Encoder Representations from Transformers)[1] 는 딥러닝 기반 자연어처리 모델로, 기존 모델들과는 달리 레이어에서 단어 좌우의 문맥을 모두 학습에 반영하여 모델을 구축하고, 그 결과 여러 자연어처리 과제에서 좋은 성능을 기록한다.

유사 문장 분류는 BERT가 높은 성능을 기록하는 자연어처리 과제들 중 하나로, 영어 기반 텍스트에서의 그 정확도는 이미 여러 차례 검증된 바 있다. 대표적으로는, 유사 문장 분류 과제에서 가장 유명한 자료집인 MRPC (Microsoft Research Paraphrase Corpus)에서 89.3%의 정확도를 기록하며 현존하는 모델들 중 최고 성능을 보여준 바 있고, MRPC와 비슷한 자료 모음인 MNLI (Multi-Genre Natural Language Inference Corpus)에서도 80% 정도의 정확도를 기록하며 ESIM, DA, SPINN보다 높은 성과를 기록하였다[2].

그럼에도 불구하고, BERT 유사 문장 분류기가 문장들의 의미를 실제로 고려하여 유사도를 측정하는지에 대해서는 여전히 의문이 남는다. BERT 학습 자료의 유사 문장 쌍들은 비슷한 단어들로 이루어진 문장들로 구성되는 경향이 있어 BERT가 단순히 단어들의 겹침을 기준으로 두 문장들의 유사도를 측정하는 것인지, 아니면 두 문장의 의미들을 파악하고 측정하는 것인지를 명확히 구분하기 어렵다. 예를 들어, "어제 야구 경기 결과가 뭐야"와 "어제 야구 경기 결과 알아"와 같은 문장들을 단순히 단어들이 많이 겹친다는 사실에 근거하여 유사도를 판단한다면, 이는 의미 기반 유사 문장 분류라 할 수 없다. 또한, 위의 예시와 비슷한 문장 쌍들로 학습된 유사 문장 분류기는 어휘가 겹치지만 의미가 다른 경우 문장들의 관계 판별을 자주 실패하곤 한다.

최근 다양한 자연어 처리를 이용한 AI 기반 서비스들이 출시되고 있다. 예를 들어, 자동 응답 시스템에서 "지난주 야구 경기 결과가 뭐야" 와 "오늘 야구 경기 결과가 뭐야"라는 두 문장을 응답하기 위해서는 서로 다른 경로를 통해 응답을 생성해야 하므로, 두 문장은 유사하지 않도록 분류되어야 한다. 그러나 최근 문장 유사도 분류 모델들은 이를 분류해내지 못한다.

본 논문은 위에서 제기된 단어 오버랩 기반 유사 문장 분류 경향을 완화시키는 학습 자료 구축 방식을 제안한다. 이 방식은 여러 문장들이 의미를 기반으로 묶여 있는 자료를 기반으로 하는데, 쌍을 이루는 두 문장들의 카테고리 동일 여부로 문장이 유사한지 아닌지를 판단하여 학습 자료에 입력한다. 본 논문은 NCSOFT에서 출시한 야구 어플리케이션 PAIGE 내의 챗봇에 저장 된 예시 FAQ 발화문들을 참고하여 학습 자료를 구성하였다. PAIGE 챗봇의 FAQ 자료는 자주 입력되는 질문 5만 줄이 140개의 카테고리로 분류되어 있다. 본 논문의 학습 자료는 PAIGE 챗봇의 사용자 발화와 PAIGE 내부에서 검색된 대표 질문이 같은 의미를 지니는지를 판별하는 유사 문장 분류기에 사용된다.

본 논문에서는 두 문장간 유사한 어휘가 출현하더라도, 문장의 의미를 파악하여 유사하지 않은 문장을 분

류해내도록 BERT를 학습시킬 수 있는 학습 데이터를 생성하는 방법론을 제안한다. Word2Vec를 통해 문장 임베딩 값들을 구하고, 유사한 문장 쌍들로 학습 자료를 구성하는 것이다. [3] 문장 임베딩은 Word2Vec를 통해 구해진 어휘들의 평균을 통해 구하였다. 따라서 유사도가 높은 문장들은 문장에 포함된 어휘가 중복되는 경향이 있다. 문장 내 어휘가 유사하지만 문장의 의미는 유사하지 않은 문장 쌍들을 BERT에 학습시켜 이 유사 문장 분류기가 문장의 의미를 결정짓는 단어들과 그렇지 않은 단어들을 학습할 수 있도록 하는 것이 본 논문의 목표이다. 또한, 본 논문에서 제안한 새로운 학습 말뭉치 생성 방법론이 BERT의 학습에 가져오는 변화를 분석하기 위하여 새로운 말뭉치로 학습된 BERT모델 내의 self-attention을 분석하였다.

2. 관련 연구

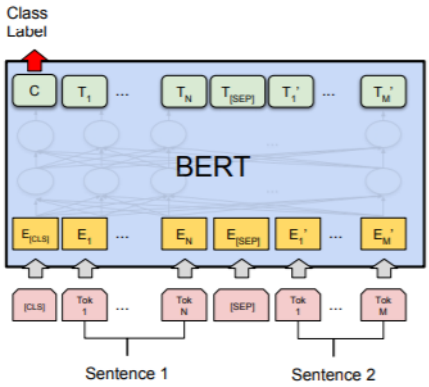
학습 자료가 유사 문장 분류기의 성능에 끼치는 영향에 대해 이미 많은 연구가 진행되고 있다. 예를 들면, 언어적 추론이 요구되는 유사 문장 분류 과제에서는 동의어, 유의어, 그리고 포함 관계에 놓여 있는 어휘들이 대량으로 학습되어야 분류기가 문장들을 의미를 고려하여 분류할 수 있다. “남자가 와인을 마시고 있어” 와 “남자가 샴페인을 마시고 있어” 라는 문장을 비교할 때는 샴페인이 와인의 일종임을 고려해야 하지만, 일반 자료로 학습된 유사 문장 분류기는 문장들의 어휘가 비슷하여 유사하다 판단한다[5]. 더 나아가, 기존의 문장 분류기는 학습 자료에 없는 반의어들을 서로 다르다고 인식하지 못하고, 문장의 핵심 의미를 바꾸는 문장 내 사소한 변화들 역시 인식하지 못한다[6].

한편, 기존의 유사 문장 분류기들이 위의 오류에 취약한 만큼, 분류기에 여러 오류들을 학습 시키면 그 성능은 높아진다[7]. MNLI 학습 자료의 문장 쌍들의 경우 두 문장들의 어휘가 일정 수준 이상으로 겹칠 시에 89.2%의 확률로 의미가 유사하였고, 유사 문장 분류기는 이 경향을 학습하여 어휘가 유사하면 의미가 유사하다 판별하게 된다. [2] 따라서 동일 BERT 모델을 어휘가 유사하지만 의미는 다른 문장들에 대해 실험을 진행한 결과 기존 MNLI 평가 자료에 대한 성능보다 60% 하락한 수치를 기록했다. 반면, MNLI 학습 자료 양의 8%를 여러 휴리스틱에 어긋나는 30개 종류의 예시들로 구성하여 학습을 진행한 결과, 동일 평가 데이터에 대해 거의 100%에 달하는 정확도가 나왔다. 이는 단순히 학습 자료의 구성 방식 변화로도 BERT의 성능이 상향될 수 있음을 보여주는 좋은 예시이다.

학습 자료의 변화나 파인 튜닝등 BERT 유사 문장 분류기에 생기는 변화가 모델의 학습에 미치는 실질적인 영향을 분석하는 방법 역시 지속적으로 연구되고 있다. 현재 사용되는 여러 방식들 중 하나로는 BERT의 레이어별 attention weights 들을 분석하는 것을 꼽을 수 있다. [4] BERT의 기본 모델은 총 12개의 레이어와 12개의 attention head로 구성되어 있으며, 학습된 BERT 모델은 12 * 12 차원의 self-attention 가중치 행렬로 표현될 수 있다. 이 행렬의 12 개의 열은 해당하는 head의 self-attention weights들의 합을 정해진 문장 길이에 맞춰 정규화 시켜 구성되고, 두 BERT 모델의 비교는 각각의 행렬의 코사인 유사도를 비교하며 이뤄진다. 예를 들어, 파인 튜닝된 BERT 모델과 기존 BERT 모델을 위의 방식으로 비교한 결과 11번째 레이어와 12번째 레이어의 코사인 유사도가 낮다면, 파인 튜닝이 BERT의 최종 두 레이어에 큰 영향을 끼쳤음을 알 수 있다[4].

3. 자연어 처리 모델 구조

3.1 BERT 유사 문장 분류기 구조



[그림 1] BERT 유사 문장 분류기[1]

본 논문은 구글에서 발표한 BERT 기반의 유사 문장 분류기 모델을 사용한다. BERT는 언어의 표현을 많은 양의 일반 도메인 말뭉치로 학습하였으며, 해당 pre-train모델을 각 task에 맞춰서 파인 튜닝하여 다양한 task에서 바로 사용할 수 있는 범용성이 뛰어난 모델이다. 다수의 자연어 분류 문제에서 가장 뛰어난 성능을 보이고 있으며, 제안하는 방법론 역시 두 문장 사이의 유사도 분류 문제로써 BERT 모델을 사용하여 실험을 진행하였다.

3.2 Word2Vec 문장 유사도 계산기

본 논문이 사용한 Word2Vec 문장 유사도 계산기는 야구 기사 23M 문장과 mlbpark의 야구 관련 글 및 댓글 31M 문장들을 기반으로 1024차원의 Word2Vec모델을 학습시켰으며, 하나의 단어를 그 문맥에 나타나는 단어들을 기반으로 임베딩하는 skip-gram 방법을 사용하였다[3]. Word2Vec 문장 유사도 계산기는 주어진 두 문장들을 구성하는 단어 임베딩 값들의 평균으로 각각의 문장 임베딩 값들을 구한 후, 두 문장 임베딩들의 코사인 유사도를 통하여 측정한다.

Word2Vec 문장 유사도는 문장을 이루는 단어들의 평균 임베딩을 기반으로 하기에, 각 문장에 포함된 어휘의 유사도가 높게 반영된다. 따라서, Word2Vec 문장 유사도가 높지만 의미가 유사하지 않은 문장들은 BERT에게 어휘 유사도와 문장 의미 유사도가 다르다는 좋은 예시가 된다.

4. 유사 문장 판단을 위한 학습 자료 구성 방법

본 논문은 어휘가 유사한 문장 쌍들로 문장 분류기 학습 자료를 구성하는 방법으로 Random 방법, Remove 방법, Neighbor 방법, Matrix 방법 총 네 가지를 제안

한다. 학습 자료 구성 방법을 연구하는 초기 단계에서는 유사한 의미의 문장 쌍들은 어휘가 겹치는 문장들로, 그리고 유사하지 않은 의미 문장 쌍들은 어휘가 겹치지 않는 문장들로 구성하였으나, 이 방법은 유사 문장 분류기의 성능을 오히려 하락시켰다. 어휘가 유사하지 않는 문장들로 의미가 다른 문장 쌍들을 형성한 결과 학습 자료 내부에서 어휘가 유사하면 의미가 유사하다는 경향이 전보다 더 두드러져, 유사 문장 분류기가 문장들의 어휘 유사도에 더욱 크게 의존하게 되었다. 따라서, 어휘가 유사하지만 의미가 다른 문장 쌍들을 중점적으로 학습 자료에 추가하기로 결정하였다. 이러한 문장 쌍들은 BERT가 단순히 두 문장이 같은 어휘를 사용하는지 여부를 학습하기보다, 전체 문장의 의미를 결정짓는 어휘를 학습하는 것을 돕는다. 또한, 공통된 단어와 의미를 지는 문장 쌍들 역시 학습 자료에 추가해 BERT가 문장들의 어휘 유사도와 의미 유사도가 상반된다는 경향을 지닌다는 것을 학습하는 것을 방지하고, 또 문장의 의미를 결정 짓는 다양한 요소들을 BERT에 학습시킨다.

4.1 FAQ 학습 자료

본 연구의 학습 자료는 [표 1]의 방식처럼 140개의 항목으로 분류된 예시 PAIGE FAQ 문장들 5 만개 의 문장들로 구성된다. 한 쌍을 이루는 두 문장은 같은 카테고리에 속하면 유사 문장 쌍, 다른 카테고리에 속하면 유사하지 않은 문장 쌍으로 분류되어 학습 자료에 추가된다. 학습 데이터는 유사 문장 쌍과 유사하지 않은 문장 쌍을 각각 10만개 포함하여 총 20만 쌍으로 구성하였다.

[표 1]: FAQ 학습 자료 분류 예시

No.	카테고리	문장
1	경기 일반	LG잘했어?
2	경기 일반	{사용자 응원 팀} 어땠어?
3	경기 일반	{사용자 응원 팀} 잘했어?
…	…	…
886	[실책] 선수	어제 LG 경기 실책한 애
887	[실책] 선수	LG 실책있는 선수
888	[실책] 선수	LG 실책한게 누구야?
…	…	…

4.2 FAQ 유사 문장 분류 평가 자료

본 연구의 평가 자료는 유사 문장 쌍 780개과 유사하지 않은 문장 쌍 1091개로 이뤄져 총 1871개의 문장 쌍으로 구성된다. 평가 자료의 문장 쌍들은 PAIGE 챗봇이 빈번히 틀리는 문장들로 구성되어, [표 2]의 예시처럼 어휘가 유사한 문장 쌍들이 많이 포함되어 있다. 따라서, 본 평가 자료에서의 성능 향상은 BERT가 어휘 유사도에 덜 의존하여 유사 문장들을 분류함을 의미한다.

[표 2] FAQ 평가 자료 예시

문장 1	문장 2	유사도 정답
1번 타자	넥센 선발 1번 타자	1
10시에 퇴근해	언제 퇴근하니?	0
16일 일정 좀 알려줘봐	오늘 한화 일정 좀 알려줘.	0
1986년 선동열 기록	오늘 김하성 기록	0
1번타자	넥센 선발 1번타자	1
20일 경기결과	kt 오늘 경기결과	0

4.3 학습 자료 구성 방법: Random

NCSOFT에서 보유한 PAIGE FAQ 데이터는 약 5만개의 문장으로 이루어져 있다. 여기서 무작위로 두 문장을 뽑는다면 대략 12억 쌍의 학습 데이터를 만들 수 있다. 그러나 모델에 학습시킬 수 있는 데이터의 양은 한도가 있다. 따라서 전체 문장 쌍 후보중 무작위로 20만건을 추출하여 학습하였다. 문장간 유사 여부는 문장이 같은 카테고리에 속해있는지 여부이다. 이 방식으로 구성된 학습 자료는 유사한 문장 쌍 10만 개, 유사하지 않은 문장 쌍 10만 개로 총 20만 쌍을 사용하였다.

[표 3] Random 학습 자료 내부의 문장 쌍 유사도 평균

유사한 문장 쌍들의 유사도 평균	유사하지 않은 문장 쌍들의 유사도 평균
0.2873	0.1174

4.4 학습 자료 구성 방법: Remove

Remove 는 Random 을 기반으로 25 만 개의 문장 쌍을 생성한 뒤, Word2Vec 유사도가 낮은 유사 문장 쌍과 높은 문장 쌍 각각 2.5 만개, 총 5 만개를 학습 자료에서 삭제하여 20 만개 문장 쌍 크기의 학습 자료를 만든다. 이 과정에서 제거되는 문장 쌍들은 “오늘 경기 결과 알려줘” 와 “포인트는 어떻게 사용하는 거야?“ 등 유사도가 현저하게 낮아 BERT 가 유사 문장들을 분류하는데 큰 도움을 주지 않는 예시들이다. [표 3]과 [표 4]는 생성한 학습 데이터 내 문장 쌍의 유사도의 평균을 측정한 것이다.

[표 4] Remove 학습 자료 내부의 문장 쌍 유사도 평균

	유사한 문장 쌍들의 유사도 평균	유사하지 않은 문장 쌍들의 유사도 평균
제거 전	0.2873	0.1175
제거 후	0.3479	0.1553

4.5 학습 자료 구성 방법: Neighbor

Neighbor 는 Random 과 Remove 가 1 차적으로 FAQ 자료 문장들을 무작위로 매칭시킨다는 한계를 극복하기 위해서 고안되었다. Neighbor 의 목적은 무작위로 구성된 학습자료에서 유사도가 낮은 문장 쌍들을 제거하기 보다 처음부터 유사한 문장들을 쌍으로 매칭시키는 것이다. 하지만, 5 만 개의 FAQ 학습 자료 문장들로 구현 가능한 모든 쌍들의 유사도들을 계산하는 것은 많은 시간과 컴퓨팅 능력을 요구한다.

반면, Neighbor 방식은 FAQ 자료 문장들을 1차원 상수로 임베딩하여 비슷한 어휘의 문장들을 편리하게 찾을 수 있다. 주어진 FAQ 자료 5만 문장들의 1024차원 임베딩 벡터를 구한 후 5만 문장들의 임베딩 벡터들의 평균을 산출하였다. 각 문장들의 상수 임베딩 값은 문장 임베딩 벡터와 전체 평균 임베딩 벡터의 코사인 유사도 값으로 계산되는데, 이는 바로 어휘가 비슷한 문장들은 전체 평균 임베딩 벡터와 비슷한 코사인 유사도 값을 지니기 때문이다. 5만개의 문장들을 상수 임베딩 값을 기준으로 정렬하여 비슷한 어휘의 문장들을 서로와 인접하게 위치하게 한 후 이웃하는 문장들을 위주로 학습 자료에 추가하였다. 다. 매칭되는 두 문장들의 간격을 조금씩 늘려가며 유사 문장 쌍 10 만개와 유사하지 않은 문장 쌍 10만개, 총 20만 개의 문장 쌍을 학습 자료에 추가였다.

[표 5] 상수 임베딩 기반 FAQ 자료 문장 정렬 예시

카테고리	문장	1차원 임베딩
경기결과_경기결과	전반전 개막 경기 전체 결과 보여줘	0.4031241
인물_선수_출전	오늘 조수행 나와?	0.4031330
숫자_기록_스탯	최근 이범호의 성적을 알고 싶어.	0.4031339
인물속성_역할_포지션	오늘 이형종은 포지션이 뭐야?	0.4031345
인물속성_역할_출전순서	오늘 이승엽이 몇번으로 나오는 거야?	0.4031729
[선수] [연봉] [특정랭킹] [특정그룹]	우리 구단 연봉 랭킹 1위는 누구야?	0.4031767
숫자_기록_스탯	장원준의 지난 경기 투구 내용은 어땠어?	0.4032034

[표 5]에서는 “전반전 개막 경기 전체 결과 보여줘” 와 “오늘 조수행 나와?” 등의 문장들이 유사하지 않은 문장 쌍으로, 그리고 “최근 이범호의 성적을 알고 싶어” 와 “장원준의 지난 경기 투구 내용은 어땠어?” 등의 문장들이 유사한 문장 쌍으로 추가된다. 이 문장들은 이전의 구성 방식 아래에서 만들어진 문장 쌍들보다 어휘와 의미 둘 다 더 겹침을 알 수 있다. 다음은 학습 자료들에 속해 있는 문장 쌍들의 평균이다

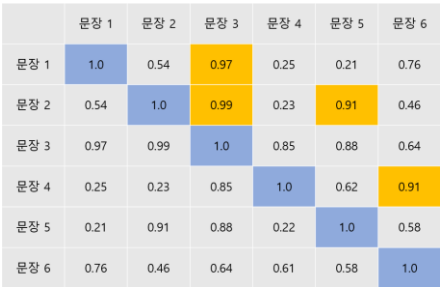
[표 6] Neighbor 학습 자료 내부 문장 쌍 유사도 평균

유사한 문장 쌍들의 유사도 평균	유사하지 않은 문장 쌍들의 유사도 평균
0.5791	0.1527

유사한 문장 쌍과 유사하지 않은 문장 쌍의 전반적인 유사도가 Random 과 Remove 학습자료의 유사도 보다 더 높아졌음을 확인할 수 있다.

4.6 학습 자료 구성 방법: Matrix

문장 쌍 간 유사도가 높은 데이터를 추가하는 것이 현재 분류 문제에서 효과가 있음이 확인되어, 결국 전체 학습 데이터간 문장 유사도를 구해보았다. Matrix는 Neighbor처럼 유사도가 높은 문장 쌍들로 학습 자료를 구성하지만, 문장들의 개별 상수 임베딩 대신 5만개의 FAQ 자료 문장들로 구상 가능한 모든 문장 쌍들을 고려하였다. 이를 계산하기 위해서는 문장들의 임베딩 벡터들로 형성된 5만 * 1024 크기의 행렬을 구한 후 그 행렬을 전치 행렬과 곱하여 5만 * 5만 크기의 행렬을 만든다. 이 때 행렬의 각 원소는 해당하는 행과 열의 두 문장들의 유사도가 된다. 이 5만 * 5만 행렬은 대칭 행렬이기에 [그림 2]의 파란색 대각선 위의 원소들만 고려했다. 학습 자료에는 [그림 2]의 노란색 원소들처럼 일정 한도의 유사도를 넘는 문장 쌍들만 추가하고, 유사 문장 쌍 10만개와 유사하지 않은 문장 쌍 10만개가 모두 구해질 때까지 유사도 한도를 1.0에서 0.25씩 낮추며 학습자료에 문장 쌍들을 추가하였다.



[그림 2] Matrix방식에서 유사도 행렬 예시

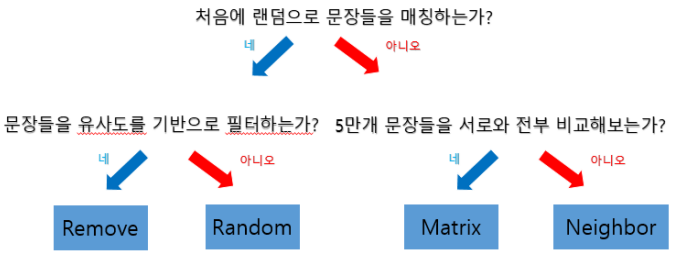
[표 7] Matrix 학습 자료 내부 문장 쌍 유사도 분포도

코사인 유사도	유사 문장 개수	x 유사 문장 개수
0.975 ~ 1.00	7904	9960
0.950 ~ 0.975	6183	5292
0.925 ~ 0.950	13938	6779
0.900 ~ 0.925	32881	9690
0.875 ~ 0.900	39094	13156
0.850 ~ 0.875	0	21330
0.825 ~ 0.850	0	33793

[표 8] Matrix 학습 자료 내부 문장 쌍 유사도 평균

유사한 문장 쌍들의 유사도 평균	유사하지 않은 문장 쌍들의 유사도 평균
0.9152	0.8850

Random, Remove, Neighbor 학습 자료의 유사도 평균이 0.60을 넘지 않았다는 점을 고려한다면 Matrix 의 학습 자료들은 타 방식들보다 월등히 높은 문장 쌍 유사도를 지니고 있음을 알 수 있다. 실제로, Matrix 학습 자료의 문장들은 “포인트 사용 방법이 뭐야” 와 “포인트 사용 방법 알려줘” 가 유사 문장 쌍으로 매칭되고, “오늘 선발 투수 알려줘” 와 “어제 선불 투수 알려줘”가 유사하지 않은 문장 쌍으로 매칭되는 등 어휘 유사도가 매우 높은 문장 쌍들로 구성되어 있다.



[그림 3] 각 학습 데이터 생성 방식의 결정 트리

5. 실험 방법 및 결과

5.1 BERT 유사 문장 분류기 hyper parameter

BERT 유사 문장 분류기는 구글에서 공개한 다중 언어를 통해 미리 학습된 모델인 multi_cased_L-12_H-768_A-12 모델을 기반으로 진행되었다. 최대 sequence length 는 50, 최대 학습 batch size 는 32, learning rate 는 2e-5, 그리고 train epochs 의 숫자는 10이라는 값들 아래에서 실험이 진행되었다

5.2 실험 결과

실험 결과 Random에서 Matrix 순으로 정확도가 증가함을 볼 수 있는데, 이 순서는 학습 자료의 문장 쌍 유사도 증가 순서와 동일하다. 평가 자료에 대한 정확도의 증가는 주로 False Positive 개수의 감소에 인해 발생하였는데, False Positive가 감소하는 과정에서 BERT가 추가로 맞히게 된 True Negative 문장 쌍들은 어휘가 유사하여 예전의 문장 분류기가 유사하다 판별하였던 쌍들이다. 이는 BERT 유사 문장 분류기 어휘 유사도가 높은 문장들을 통해 평가 자료의 단어들이 겹쳐도 문장의 의미가 상이할 수 있음을 학습하였기 때문이다.

[표 9] 평가 자료에 대한 BERT의 정확도 분석

학습자료	False Positive 개수	False Negative 개수	Accuracy
Random	593	76	0.6424
Remove	547	79	0.6654
Neighbor	474	109	0.6884
Matrix	185	265	**0.7595**

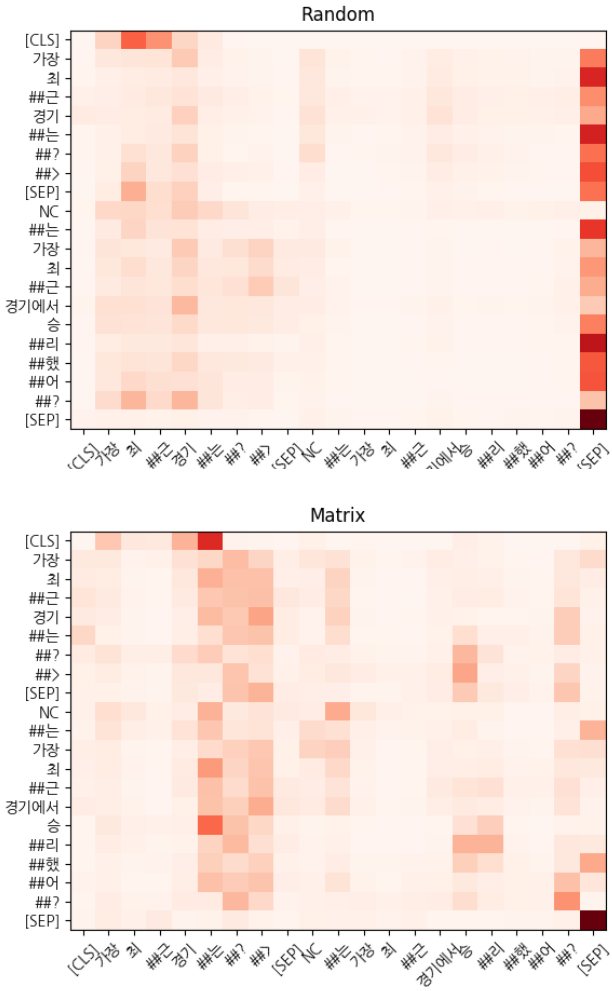
* False Positive: BERT가 유사하다 판단하였으나 실제로는 유사하지 않은 평가 자료 문장 쌍
* False Negative: BERT가 유사하지 않다 판단하였으나 실제로는 유사한 평가 자료 문장 쌍

[표 10] 평가 자료에 대한 BERT분류기 성능

학습자료	Recall	Precision	F1
Random	0.9026	0.5428	0.6779
Remove	0.8987	0.5617	0.6913
Neighbor	0.8603	0.5860	0.6971
Matrix	0.6594	**0.7350**	0.6951

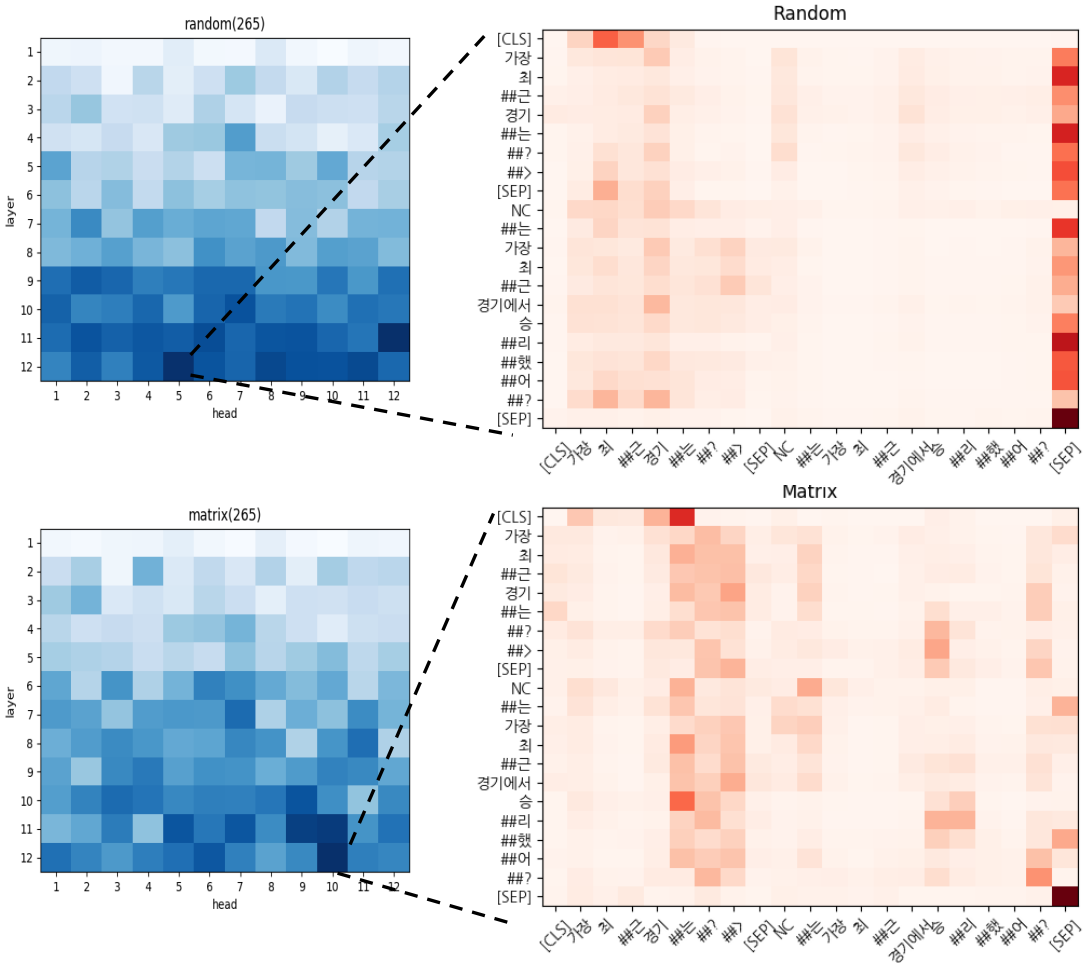
실제로 오직 Matrix에서 학습한 유사 문장 분류기만 “오늘 경기” 와 “오늘 경기 날씨”를 다른 문장으로 분류하였다. 반면, 두 문장들이 유사하다 판단하는 기준이 높아져 False Negative 의 개수는 오히려 증가하는 추세를 보이기도 했다. Recall이 계속 감소한 이유는 바로 유사 문장 분류기가 예전처럼 두 문장들을 단순한 단어 오버랩을 기반으로 무분별하게 유사하다 판별하지 않아서이고, 동일한 이유로 Precision은 증가하였다. Recall 과 Precision의 상반된 변화로 F1값 자체는 크게 변화하지 않았다.

6. 학습 자료의 변화가 BERT 유사 문장 분류기의 학습에 주는 영향



[그림 5] 평가 데이터에서 random방식(위)과 matrix방식(아래)의 self-attention weight 변화가 관측된 예시

본 논문에서는 문장의 어휘, 구조가 매우 유사한 두 문장이 실제로도 같은 의미인지 여부를 BERT모델에 성공적으로 학습시켰다. 우리는 BERT 내부에서 실제로 어떤 변화가 있었는지 확인하기 위해 BERT 내의 self-attention weight의 변화를 살펴보았다. 논문에서 사용한 BERT의 기본 버전은 총 12개의 layer, 12개의 head를 사용하는 multi-head attention 구조이다. [그림 6]은 BERT 기본 모델과 해당 모델에 random, matrix방식으로 구축한 학습 데이터를 추가로 학습시킨 모델에서 학습된 self-attention weight를 코사인 유사도를 통하여 비교하여 히트맵을 구축한 모습이다(왼쪽). 히트맵



[그림 6] BERT 기본 multi-lingual모델과, random, matrix방식의 학습 데이터를 추가로 학습시킨 모델을 비교하여 코사인 유사도를 나타내었다(왼쪽). 색이 진할수록 기본 모델과 다른 파라미터로 학습되었다는 의미이며, 두 방식에서 가장 진한 부분의 실제 attention matrix를 살펴보았다(오른쪽)

상 색이 진할수록 기존 모델에서 학습된 값과 유사도가 낮다는 의미이며, 기존 모델에 비해 가장 많이 추가적으로 학습이 된 Head라고 생각할 수 있다. 가장 색이 진한 head에서의 self-attention을 직접 확인하기 위해 실제 평가 데이터에 존재하는 데이터의 attention matrix를 확인하였다. [그림 5]에서는 실제 평가데이터에 존재하는 문장인 “가장 최근 경기는?>”과 “NC는 가장 최근 경기에서 승리했어?”가 같은 종류의 문장인지 분류할 때 분류기 네트워크 내부에 잡힌 attention이다. 두 문장은 어휘 출현이 유사하지만, 응답이 다른 방식으로 생성되므로 서로 다른 문장으로 분류되어야 한다. Random 방식의 학습 데이터에서 네트워크는 두 문장 중 “최근”이라는 토큰에 중점을 둔 모습을 확인할 수 있다. 이는 아마도 두 문장에 동시에 가장 먼저 출현한 어휘가 “최근”이었기 때문이라 사료된다. 이처럼 무작위 방식으로 유사 문장 분류를 하였을 때에는 앞서 여러 번 언급한 것처럼 네트워크가 유사한, 혹은 같은 어휘의 사용에 집중하는 모습을 보였으며 결국 분류를 실패하였다. 그러나 유사한 문장을 중점적으로 학습시킨 Matrix 방식의 학습 데이터에서는 첫 문장의 마지막 토큰인 “는?>”에 가장 높은 attention이 부여되며, 비교 문장에서의 마지막 토큰 “승리했어?”에 높은 attention을 부여함으로써 두 문장의 의미 차이를 성공적으로 찾아냈고, 결국 두 문장은 서로 다른 문장이라고 분류하였다.

7. 결론

본 논문은 BERT 유사 문장 분류기가 어휘와 문장 구조가 비슷한 문장들을 효과적으로 분류할 수 있는 학습 자료들을 다양한 방법으로 구성하여 실험했다. 학습 자료의 문장 쌍들을 단순히 문장 카테고리의 동일 여부로 무작위로 매칭할 시, 학습 자료의 유사 문장 쌍들은 유사하지 않은 문장 쌍들보다 높은 어휘 유사도를 지니게 되고, 분류기는 이 경향을 학습하여 문장들의 의미 유사도를 문장 내 출현하는 어휘 유사도에 의존하여 판단하게 된다. 반면 위해 어휘가 유사한 문장들을 위주로 분류기를 학습한 결과, 분류기가 어휘가 유사한 문장들도 다르다고 판단하는 능력이 향상되어 평가 자료에 대한 성능이 높아졌다.

한편, 본 논문의 학습자료로 구축된 분류기가 문장간 유사도 판별 시 어휘 유사도를 배제하는 것은 아니다. 제안하는 학습 자료 역시 어휘와 의미 둘 다 유사한 문장 쌍들을 여럿 포함하기에 어휘 유사도가 유사

문장 판별 시 반영되고, 따라서 의미가 다르지만 어휘가 매우 유사한 문장들을 여전히 유사하다 분류하는 경향 역시 존재한다. 본 논문의 의의는 기존의 유사 문장 분류기가 보이는 어휘 유사도에 대한 높은 의존도를 완화시키는 데에 있다.

참고문헌

[1] DEVLIN, Jacob, et al. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.

[2] MCCOY, Thomas, et al. Right for the Wrong Reasons: Diagnosing Syntactic Heuristics in Natural Language Inference. arXiv preprint arXiv: 1902.01007, 2019.

[3] MIKOLOV, Thomas, et al. Efficient Estimation of Word Representations in Vector Space. arXiv preprint arXiv: 1301.3781, 2013.

[4] KOVALEVA, Olga, et al. Revealing the Dark Secrets of BERT. arXiv preprint arXiv:1908.08593, 2019.

[5] GLOCKNER, Max, et al. Breaking NLI Systems with Sentences that Require Simple Lexical Inferences. arXiv preprint arXiv: 1805.02266, 2018.

[6] SANCHEZ, Ivan, et al. Behavior Snalysis of NLI Models: Uncovering the Influence of Three Factors on Robustness. arXiv preprint arXiv: 1805.04242, 2018.

[7] GURURABGAN, Suchin, et al. Annotation Artifacts in Natural Language Inference Data. arXiv preprint arXiv: 1803: 02324, 2018.