한국어 STS 척도 개선 및 모델 성능 향상

서광욱, 황은해
(Seo-KwangWook, Hwang-Eunhae)

1. Introduce

1.1 STS

NLU(Natural Language Understanding)에서 주요 과제 중 하나로 평가되는 STS 는 유사문장평가(Semantic Textual Similarity)의 약자로, 텍스트의 의미적 유사성을 판단하는 과제이다.

ㄱ. 너 설마 약속 취소할 생각을 하는 건 아니겠지?
ㄴ. 약속을 취소하는 건 진짜 그릇된 행동이야.

[표 1] 핵심내용을 공유하는 유사한 문장 예시

[표 1]의 문장 ㄱ과 문장 ㄴ은 표면적으로 상이한 어휘와 문장구조와 이루어져 있어, 기계는 다른 문장이라고 판단하기 쉽다. 하지만, 두 문장 모두 핵심 내용은 '약속을 취소하면 안 된다.'를 의미하고 있기 때문에 인간의 관점에선 두 문장을 의미상으로 유사한 문장이라고 판단할 수 있다.

1.2 STS 의 유사 척도와 평가 지표

점수	내용
5	두 문장은 중요한 내용과 중요하지 않은 내용이 동등하다.
4	두 문장은 거의 동일하다 일부 중요하지 않은 내용은 다르다.
3	두 문장은 대략 같다. 중요한 콘텐츠는 서로 비슷하지만 중요하지 않은 콘텐츠의 차이도 무시할 수 없다.
2	두 문장은 동등하지 않다.

	중요한 콘텐츠는 서로 비슷하지 않고 일부 중요하지 않은 콘텐츠만 공유한다.
1	두 문장은 동등하지 않다. 중요한 내용과 중요하지 않은 내용이 서로 비슷하지 않다. 두 문장은 주제만 공유한다.
0	두 문장은 동등하지 않다. 중요하지 않은 내용과 심지어 주제도 공유하지 않고 있다.

[표 2] 한국어로 번역한 KLUE-STS 의 라벨링 척도[1]

STS 는 유사도를 0 점과 5 점 사이의 점수를 통해 나타낸다. [표 2]는 KLUE benchmark STS 척도를 한국어로 번역한 것이다. 이에 따르면, 0 점은 두 문장이 완전히 동등하지 않은 수준을 나타내며 5 점은 두 문장이 완전히 동등한 수준을 의미한다.

STS 는 task 를 평가하기 위한 지표로서 피어슨 상관계수(Pearson Correlation coefficient)와 F1-score 를 사용한다.

$$\gamma_{XY} = \frac{\sum_i^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_i^n (X_i - \bar{X})^2}\sqrt{\sum_i^n (Y_i - \bar{Y})^2}}$$

[그림 1] 피어슨 상관계수(Pearson Correlation coefficient)

피어슨 상관계수는 변수 X, Y 사이의 상관관계가 어떻게 되는지를 나타낸 값이며, KLUE-STS 에서는 사람이 부여한 라벨 점수와 모델이 예측한 라벨 점수 사이의 상관관계를 측정하기 위해 사용한다.

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

where :

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

[그림 2] F1-score

[1] KLUE: Korean Language Understanding Evaluation(2021)

F1-score 는 threshold 3 을 기준으로 하여 3 점 미만은 '유사하지 않음', 3 점 이상은 '유사함'으로 판단한 binary-label 을 바탕으로 측정한다.

1.3 한국어 STS benchmark dataset

한국어의 STS benchmark 이라 불리는 데이터셋은 크게 2 가지로, 카카오브레인이 공개한 'KorSTS'와 KLUE 가 제공하는 NLU 데이터셋에 포함된 STS 데이터셋(본 연구에서는 해당 데이터를 'KLUE-STS'라고 칭함)이 존재한다.

1.3.1 KorSTS

KorSTS 는 영어로 된 STS-B 데이터셋의 문장들을 사용하였으며, 기계번역을 통해 한국어로 번역된 문장 쌍을 생성했다. 또한 한국어로 번역한 문장 쌍에 대한 점수를 매긴 것이 아니라 영어 원문의 라벨링 점수가 그대로 사용되었다. 이러한 방식으로 생성된 KorSTS 는 번역으로 인해 발생하는 문맥적인 차이를 라벨에 반영하지 못한다는 결점이 존재한다.

1.3.2 KLUE-STS

KLUE-STS 는 한국어로 된 말뭉치를 직접 수집하여 생성한 데이터다. 유사 문장을 만들어주는 과정에서 기계 번역을 이용하였으나, 모든 문장에 대해 한국인 작업자들이 직접 해당 문장 쌍의 의미적 유사도에 맞는 라벨 점수를 부여했다. 즉, KLUE-STS 는 번역으로 생성된 문장이 존재하지만 모든 문장에 대해 한국인이 직접 판단한 라벨 점수가 부여되었다는 점에서 한국어의 문맥적 특성이 해당 점수에 반영된 벤치마크 데이터셋이라 할 수 있다.

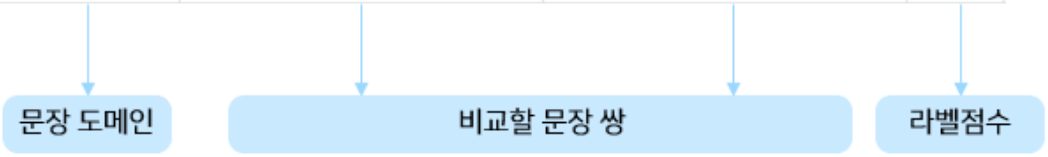
따라서, 본 연구의 목표 달성을 위한 주요 과제인 '기존 STS 벤치마크 개선'을 수행하기 위해 적합한 데이터로 한국어 문장 쌍의 의미적 유사도를 한국인 작업자들이 점수로 매긴 'KLUE-STS'를 채택하였다.

2. 데이터

2.1 KLUE-STS

KLUE-STS 는 train 11,668 개, validation 519 개, test 1,037 개로 구성되어 있다. KLUE-STS 는 데이터 하나에 2 개의 문장이 한 쌍을 이루며 해당 문장 쌍에 대한 라벨이 함께 포함된 구조다. [그림 3]은 KLUE-STS 의 간략한 구조를 설명한 자료다.

guid	source	sentence1	sentence2	label
klue-sts-v1_dev_00000	airbnb-rtt	무엇보다도 호스트분들이 너무 친절하셨습니다.	무엇보다도, 호스트들은 매우 친절했습니다.	4.9
klue-sts-v1_dev_00001	airbnb-sampled	주요 관광지 모두 걸어서 이동가능합니다.	위치는 피렌체 중심가까지 걸어서 이동 가능합니다.	1.4
klue-sts-v1_dev_00002	policy-sampled	학생들의 균형 있는 영어능력을 향상시킬 수 있는 학교 수업을 유도하기 위해 2018학년도 수능부터 도입된 영어 영역 절대평가는 올해도 유지한다.	영어 영역의 경우 학생들이 한글 해석본을 암기하는 문제를 해소하기 위해 2016학년도부터 적용했던 EBS 연계 방식을 올해도 유지한다.	1.3
klue-sts-v1_dev_00003	airbnb-rtt	다만, 도로와 인접해서 거리의 소음이 들려요.	하지만, 길과 가깝기 때문에 거리의 소음을 들을 수 있습니다.	3.7
klue-sts-v1_dev_00004	paraKQC-para	형이 다시 캐나다 들어가야 하니 가족모임 일정은 바꾸지 마세요.	가족 모임 일정은 바꾸지 말도록 하십시오.	2.5



[그림 3] KLUE-STS 의 간소화된 데이터 구조

2.1.1 문장 도메인

Airbnb Reviews(AIRBNB)	에어비앤비 리뷰 데이터
Policy News(POLICY)	정책 뉴스 브리핑 자료
ParaKQC(PARAKQC)	스마트 홈 기기를 위한 발화 데이터
airbnb-rtt, policy-rtt	AIRBNB, POLICY 문장 중 papago 를 이용해 RTT 방식으로 유사한 문장 쌍을 만든 데이터

[표 3] KLUE-STS 의 문장 도메인

KLUE-STS 의 문장 도메인은 Airbnb 리뷰, policy (공식 뉴스), paraKQC(smart home queries), RTT 방식으로 생성한 문장들이 포함된다.

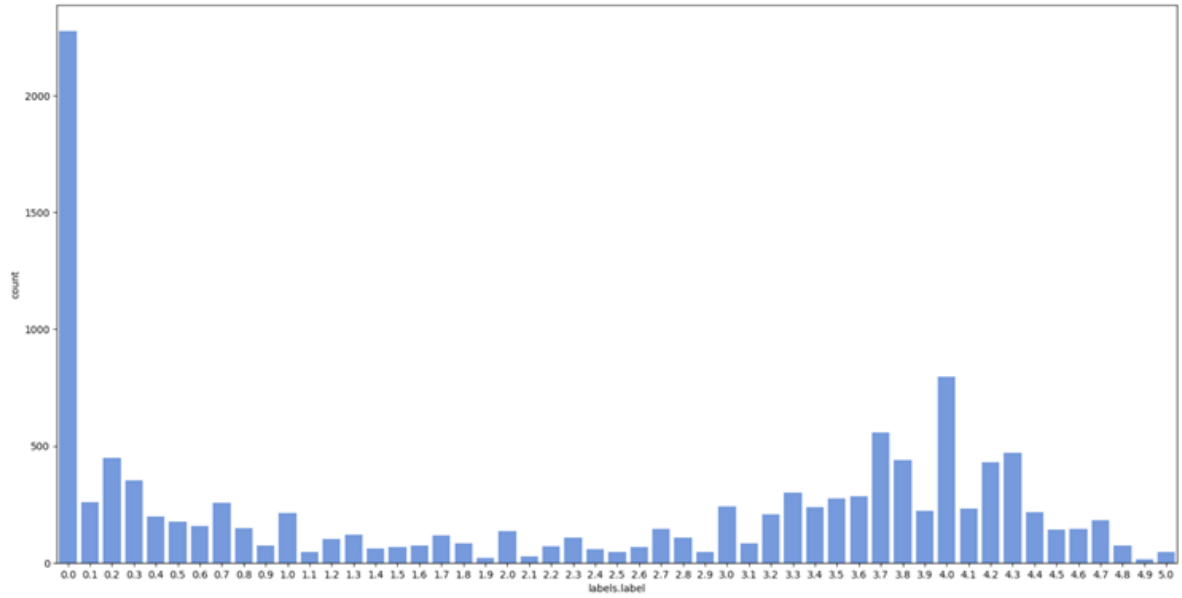
2.1.2 라벨 점수

여러 작업자가 0 점~5 점 사이의 정수로 점수를 부여한 후, 라벨 점수들의 평균을 최종 라벨 점수로 사용하였기 때문에 KLUE-STS 의 라벨 점수는 소수점이 존재하는 연속형 수치로 나타난다.

2.2 KLUE-STS 의 문제점

기존 KLUE-STS 는 크게 4 가지의 문제점을 보이고 있으며, 라벨 점수의 2.0 ~ 3.5 구간에서 해당 문제가 두드러지게 나타나고 있다. 따라서 본 연구는 KLUE-STS 데이터셋의 전체의 라벨 점수를 교정하는 것이 아니라 다음 4 가지의 근거에 따라 문제점을 교정해 줄 라벨 점수의 구간을 2.0~3.5 로 좁혔으며, 해당 구간에서 두드러지는 근본적인 문제점을 해결하려 한다.

(1) 균일하지 않은 점수 분포



[그림 4] KLUE-STS train set 의 라벨 점수 분포

[그림 4]에 따르면, KLUE-STS train set 중 1.0 점과 3.0 점 사이 데이터의 수가 타 점수대에 비해 매우 부족한 것을 확인할 수 있다. KLUE-STS 는 sampling 을 통해 데이터를 생성했기 때문에 대부분의 점수가 양극단 쪽에 치우쳐져 있으며, 가장 중간 구간의 점수인 2.0 ~ 3.0 사이에 해당하는 점수 구간의 데이터양이 다소 부족하다. 이처럼 구간별 데이터 수가 고르게 분포하지 않은 데이터로 기계를 학습시킨다면, 데이터가 부족한 구간에 대해서 잘못된 학습을 진행하여 잘못된 라벨링을 할 확률이 높아진다.

(2) 잘못된 번역으로 생성된 데이터

ㄱ. 집도 호스트님도 뭐하나 빠짐없이 베스트였습니다.
ㄴ. 집과 숙주가 최고였어요.

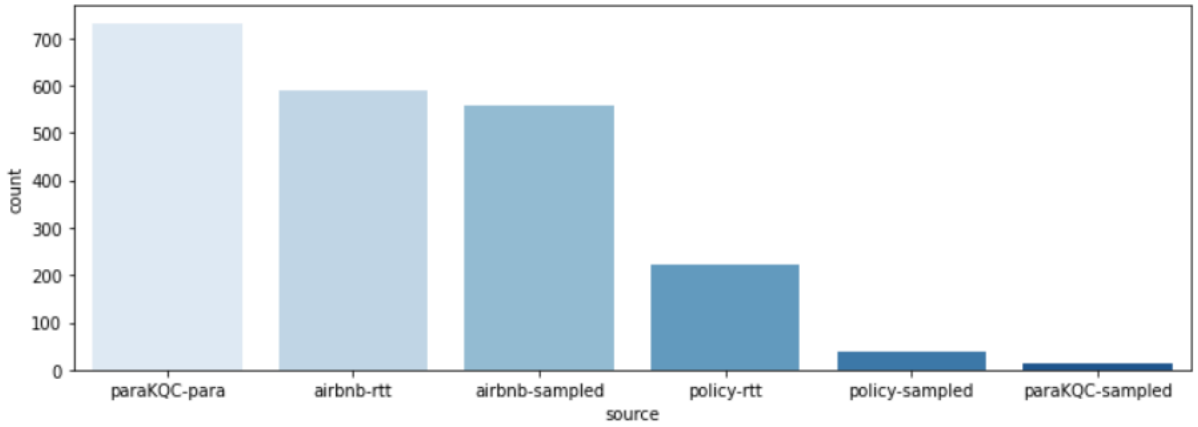
[표 4] 문맥상 이해하기 힘든 단어의 등장 (klue-sts-v1_train_09477, airbnb-rtt, 2.8 점)

ㄱ. 지난 2005 년 노무현 전 대통령이 친환경 수소경제 구현을 위한 마스터플랜을 수립한 이후 14 여년 만이다.
ㄴ. 2005 년에 겨우 14 년 이후에 대한 종합 계획을 노무현 전 대통령은 녹색 수소 경제.

[표 5] 주성분이 빠진 비문 (klue-sts-v1_train_00577, policy-rtt, 2.5 점)

[표 4]와 [표 5]는 KLUE-STS train set 에서 각각 2.8 점과 2.5 점을 받은 문장이다. [표 4]의 경우 문장 ㄴ의 '숙주'라는 단어가 문맥상 어울리지 않아 해당 문장의 내용을 이해하기 어려움에도 불구하고, 2 점 후반대의 점수를 부여받았다. [표 5]의 문장 ㄴ은 문장의 주성분인 서술어가 빠져있으며, 문장성분의 호응이 잘못된 비문이지만 2 점 중반대의 점수를 부여받았다. 기본적으로 KLUE-STS 의 데이터 구축 가이드라인을 살펴보면 오역이 존재하는 문장의 경우 번역임을 감안하여 작업자들이 자체적으로 의미를 재해석한 후, 점수를 부여하도록 유도하고 있다. 이는

KLUE-STS 중 RTT(Round-Trip Translation) 방식을 이용해 생성된 데이터(airbnb-rtt, policy-rtt)가 가진 본질적인 오역 문제를 해결하지 못한 채 점수를 부여하므로 라벨 점수의 신뢰성을 떨어뜨린다. 그뿐만 아니라, [표 4]와 [표 5]와 같은 유형의 비문들은 기계의 학습 과정과 작업자들의 라벨링 과정에서 큰 혼란을 야기하는 핵심적 원인이다.



[그림 5] KLUE-STS train 의 라벨 점수 2.0 ~ 3.5 구간 내 source 별 비중

또한, [그림 5]에 따르면 KLUE-STS train 중 라벨 점수 2.0 ~ 3.5 구간에서 airbnb-rtt 데이터가 두 번째로 큰 비중을 차지하고 있다는 점에서 rtt 로 생성된 데이터가 지닌 본질적 오역 문제가 라벨 점수 2.0 ~ 3.5 구간에 다소 큰 영향을 미치고 있음을 유추할 수 있다.

(3) 주관적 판단이 개입되는 점수 척도

기존 KLUE-STS 의 라벨 척도 [표 2]에 따르면, 라벨 점수를 부여하는 작업자가 주관적인 판단을 바탕으로 라벨링 작업을 하게 된다. 특히 2 점의 '두 문장은 동등하지 않다. 중요한 콘텐츠는 서로 비슷하지 않고 일부 중요하지 않은 콘텐츠만 공유한다.'와 3 점의 '두 문장은 대략 같다. 중요한 콘텐츠는 서로 비슷하지만 중요하지 않은 콘텐츠의 차이도 무시할 수 없다.' 같은 내용은 작업자마다 다른 기준으로 문장을 해석하는 결과를 낳는다. 결국 모호한 척도로 인해 문장 쌍에 부여되는 라벨 점수의 편차가 커지게 된다. 실제 KLUE-STS 내 데이터 중 유사한 유형 혹은 유사한 점수대를 받아야 하는 문장임에도 서로 다른 bin 에 속하는 점수를 부여받은 데이터들이 다수 발견되었는데, 특히 STS task 가 binary-label 의 threshold 로 삼는 3 점 부근에서 가장 많이 발견되었다.

(4) 오류율이 가장 높은 2.0 ~ 3.5 구간

bin	TP	TN	FP	FN	accuracy
1.3 – 1.9	0	37	6	0	0.860465
2.0 – 2.5	0	31	24	0	0.563636
2.6 – 3.0	9	8	34	2	0.320755
3.1 – 3.5	48	0	0	7	0.872727
3.6 – 4.0	53	0	0	2	0.963636
4.1 – 4.5	55	0	0	0	1.0
4.6 – 5.0	44	0	0	0	1.0

[표 6] 기존 KLUE-STS 로 학습한 모델의 정확도

[표 6]의 기존 KLUE-STS 데이터셋으로 학습을 진행한 모델의 성능을 살펴보면 2.0 ~ 2.5 구간의 accuracy 는 약 0.56 이며, 2.6 ~ 3.0 구간의 accuracy 는 약 0.32 를 보인다. 해당 구간들은 타 구간에 비해 매우 낮은 정답률을 보이는 것으로 나타났다. 선행연구에 따르면 기계가 예측한 문장 쌍의 유사도 점수와 인간 작업자가 판단한 유사도 점수가 불일치하는 문장이 2.4 점과 3 점 사이에서 거의 분포하는 결과가 도출되는데, 이는 3 점 주변 문장 쌍들의 쌍방 함의 관계를 판단하기 어려운 유사 문장의 경계이기 때문이라고 밝힌 바 있다.

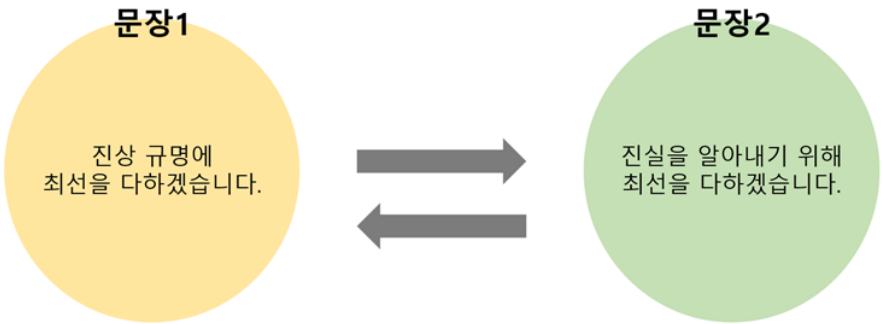
위와 같은 4 가지의 근거에 기반해 본 연구는 KLUE-STS 의 근본적인 문제점을 안고 있는 핵심 구간을 2.0 ~ 3.5 점이라고 파악했으며, 이에 따라 STS 척도 표준화를 실시해줄 구간을 2.0 ~ 3.5 점으로 설정했다.[2]

3. STS 라벨 척도 재정립

3.1 함의 관계

본 연구의 쌍을 이루는 두 문장 간 유사성을 판단하는 주요 기준은 내용 간 '함의' 여부이다. '함의'는 크게 상호함의, 일방함의, 비함의로 구분 짓는다.

(1) 상호함의

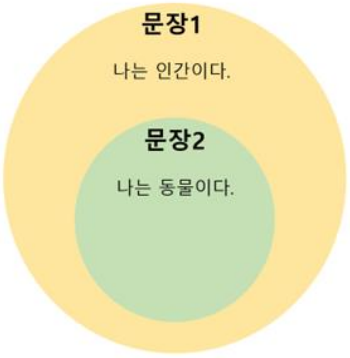


[그림 6] 상호함의

[2] 유사 문장 말뭉치 분석을 통한 유사도 인식에 관한 연구 (2021, 어문연구학회)

예시의 문장 1 과 문장 2 의 내용처럼 서로 같은 의미를 완전히 공유하고 있는 문장은 두 문장은 '상호함의' 관계에 있다고 본다.

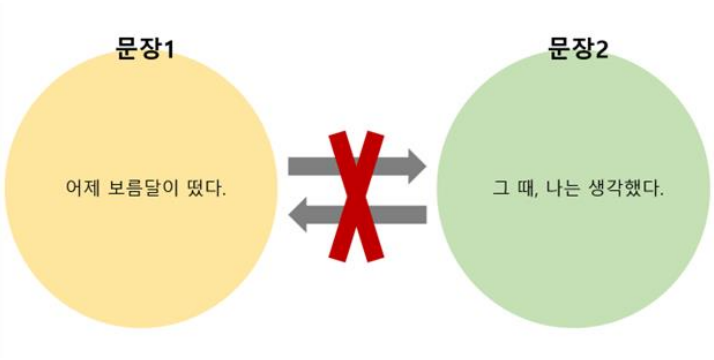
(2) 일방함의



[그림 7] 일방함의

'일방함의'는 하위어가 상위어를 함의하지만, 그 역은 성립하지 않는 관계다. 문장 1 의 '인간'은 문장 2 의 '동물'을 함의한다. 하지만, 문장 2 의 '동물'은 '인간'과 '인간이 아닌 움직이는 생명체'를 모두 포괄하는 명사이기 때문에 '인간'을 함의할 수 없다. 이처럼 어느 문장이 다른 문장의 내용을 함의하지만, 그 역은 성립하지 않는 경우 '일방함의' 관계에 있다고 본다.

(3) 비함의



[그림 8] 비함의

'비함의'는 두 문장의 내용이 서로 어느 한 쪽에도 함의되지 않는 관계에 있다.

3.2 핵심맥락과 보조맥락

기존 KLUE-STS 의 라벨 척도 [표 2]에 따르면, 2 점과 3 점의 판단이 가장 모호하다. 2 점과 3 점의 판단에서 가장 핵심이 되는 것은 '중요한 내용'과 '중요하지 않은 내용'을 구분하는 것이다. 본 연구에서는 문장의 의미를 살피기 위해 '핵심맥락'과 '보조맥락'으로 문장의 '중요한 내용'과 '중요하지 않은 내용'을 구분 지었다. 또한, '핵심맥락'과 '보조맥락'의 함의 관계를 따져 라벨 척도를 보다 구체화하였다.

핵심맥락과 보조맥락은 작업자의 주관에 따라 달라질 수 있으므로 명확한 구분법을 세우지는 않았다. 다만, '중요한 내용'과 '중요하지 않은 내용'으로 구분했던 기존 척도에 비해 더욱 명확하고 일관적으로 주요 내용을 구분할 수 있다는 점에서 해당 절차가 유의미하다고 판단했다. 또한 기존 KLUE-STS 라벨 척도를 크게 해치지 않는 선에서 라벨링이 가능하도록 기존 KLUE-STS 작업 가이드의 '중요 내용'을 판단하는 부분을 보완하여 맥락을 구분하였다. 예시는 아래와 같다.

문장 1	나한테 효율적으로 집안에서 환기하려면 뭐가 필요한지 알려주세요
문장 2	발코니말고 집안에서 효율적으로 환기하고 싶은데 뭐가 필요할까?

[표 7] 핵심맥락과 보조맥락을 구분하는 문장 예시

[표 7] 문장 1 의 핵심 내용과 문장 2 의 핵심내용은 '집 안에서 환기하려면 뭐가 필요한지 알려달라'이다. 문장 1 의 보조맥락은 '나한테', '효율적으로' 이며, 문장 2 의 보조맥락은 '발코니 말고', '효율적으로'라고 분류할 수 있다.

이처럼 작업자가 개별적으로 핵심맥락과 보조맥락을 판단하는 과정에서 개입된 주관성을 보완하고 보다 객관적인 라벨링이 가능하도록 맥락 간 함의 여부에 따라 점수를 부여하는 절차를 도입했다.

3.3 재정립한 STS 라벨 척도

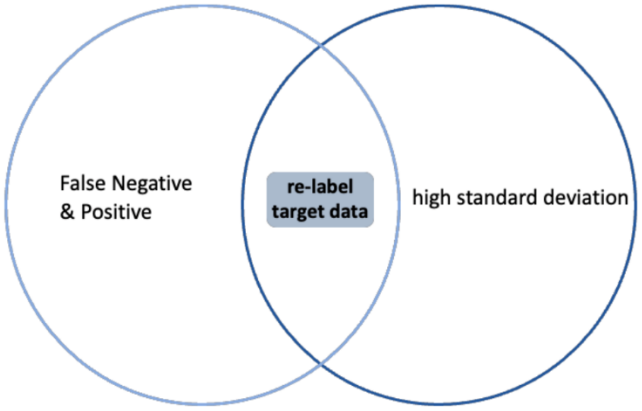
점수	새로 정립한 기준	기존 KLUE 작업 가이드라인
4		핵심 내용의 의미는 포함되어 있다. 보조 내용이 약간 포함되어 있지 않다.
3.5	핵심맥락 상호함의(2) + 보조맥락 상호함의(2)	
3	핵심맥락 상호함의(2) + 보조맥락 일방함의(1) 핵심맥락 일방함의(1) + 보조맥락 상호함의(2)	핵심 내용의 의미가 약간 달라진다. 또는 핵심 내용은 그대로지만, 보조 내용의 큰 의미 차이가 있다.
2.5	핵심맥락 상호함의(2) + 보조맥락 비함의(0) 핵심맥락 일방함의(1) + 보조맥락 일방함의(1) 핵심맥락 일방함의(1) + 보조맥락 비함의(0) 핵심맥락 비함의(0) + 병렬 정보 많은경우	
2	핵심맥락 비함의(0)	핵심 내용은 달라진다. 하지만, 핵심 내용 주제 일부/핵심 내용 수식어에서 공통 주제를 많이 공유한다.

[표 8] 새로 정립한 STS 라벨 척도와 기존 KLUE 작업 가이드라인의 비교

[표 8]의 2 번째 열 '새로 정립한 기준'은 본 연구에서 '함의' 개념과 '핵심맥락', '보조맥락'을 사용하여 새로 정립한 STS 라벨의 척도이며, 기존 KLUE 작업 가이드라인과 비교한 것이다.

3.4 재 라벨링이 필요한 데이터 구간 선정

계획 초기에는 2.0-3.5 구간에 해당하는 데이터들에 대해 작업자를 고용하여 전부 라벨링 후 평균을 통해 점수를 정규화하는 프로세스를 구상하였다. 하지만 변경된 점수 척도가 기존의 작업방식보다 세부적인 전달 사항이 많아 기준을 이해시키는 것에 상당한 시간이 소요되고 완성된 데이터가 프로젝트의 마감일까지 완성될 수 없었기 때문에 해당 방식이 적절하지 않다고 판단했다. 따라서 두 명의 인턴이 직접 라벨링을 진행하는 방식으로 선회하였다. 적은 수의 인원과 시간 투입, 그리고 효과적인 라벨 교정을 위해 정립한 기준을 바탕으로 KLUE-STS 데이터셋의 train set 중 모델의 성능에 큰 영향을 미치는 데이터들을 선별하는 과정을 거친 후 해당 데이터에 대해서만 라벨링을 진행했다.



[그림 9]

잘못된 데이터의 선별은 active learning 에서 활용되는 방식을 응용했다. Active learning 은 모델이 부족한 라벨데이터를 통해 학습한 결과를 바탕으로 각 데이터의 예측 불확실성을 측정한 후 이를 통해 모델이 성능을 높이기 위한 데이터를 추가적으로 작업자에게 요청하는 방식이다. 작업자는 해당 데이터에 대해 annotation 을 진행해 모델에 입력으로 들어갈 데이터를 업데이트해 줌으로써 성능을 향상시킬 수 있다. 해당 방식을 재라벨 대상을 선정하는 과정에 응용하여 STS 모델의 유사도 binary 판단에서 오류가 발생하는 데이터와 예측 불확실성이 높은 데이터의 교집합을 train set 중 재라벨을 진행할 데이터로 추출하였다.

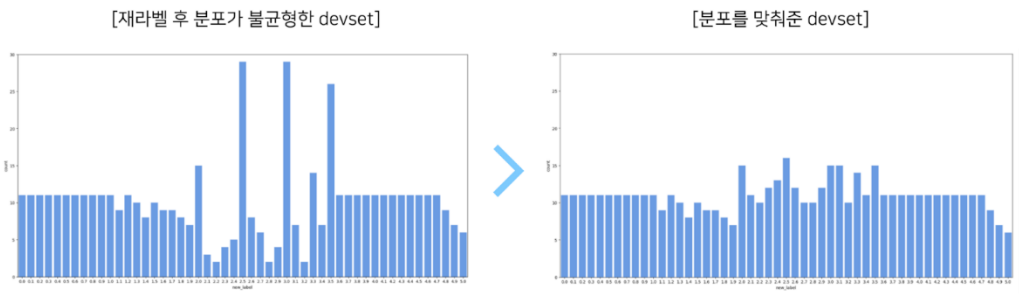
[그림 10]

구체적인 추출방식은 [그림 10]과 같다. 우선 KLUE train set 의 2.0-3.5 점수구간에 해당하는 데이터들을 작은 폴드 단위로 나누어 학습과 테스트를 반복하는 교차검증을 진행한 후 유사도 이진분류 판단에 오류가 발생한 데이터들을 수집했다. 이후 각 데이터에 학습한 모델이 가지는 예측 불확실성을 측정하기 위해 모델 앙상블을 이용했다. 같은 데이터로 학습한 서로 다른 9 개의 모델을 생성하여 각 모델이 하나의 데이터에 대해 예측하는 9 개 라벨의 표준편차를 계산하고 일정 표준편차 이상의 데이터를 예측 불확실성이 높은 데이터로 간주하여 추출하였다. 이에 따라 모델이 자주 틀리면서 예측 불확실성도 높은 412 개의 재라벨 대상 데이터를 train set 에서 선정하였다.

KLUE-devset 의 경우 모델 실험 결과를 평가하는 test 데이터로 활용되기 때문에 추출과정을 거치지 않고 train set 과 동일한 점수 구간에 해당하는 150 여 개의 전체 데이터를 재라벨링 하여 확실한 성능측정 벤치마크를 만드는것에 집중했다. 이에따라 train set 과 dev set 을 합쳐 총 560 여 개의 데이터에 대해 두 명의 인턴이 작업자로 참여하여 재라벨을 진행했다. 모든 대상 데이터를 두명의 작업자가 동일하게 4 가지 점수로 라벨링 하였으며 모든 데이터에 대해 토의를 거쳐 통일된 하나의 라벨로 확정하였다. 판단의 모호성이 여전히 존재하는 소수의 데이터에 대해서는 두 작업자 라벨값의 평균을 기입했다.

3.5 불균형한 dev set 의 라벨분포를 교정한 새로운 데이터셋 생성

새롭게 정립한 기준에 따라 라벨링이 이루어진 데이터들은 평균값으로 정규화되지 않고 두 명의 작업자에 의해 4 가지 점수로 이산화되었기 때문에 필연적으로 기존 데이터셋의 분포에 변화를 초래했다. Train set 의 경우 기존 라벨분포가 이미 불균형한 상태였고 전체 데이터 수에 비해 적은 수의 데이터만 변경되었기 때문에 분포에 따른 train set 자체의 교정은 필요하지 않았다. 하지만 devset 의 경우 새로운 라벨의 성능검증을 위한 test set 으로 사용되고 기존 라벨이 정확한 분포가 맞춰져 있었기 때문에 라벨 분포를 맞춰줘야 할 필요성이 있었다.



[그림 11]

[그림 11]에서 보이는 것처럼 점수의 이산화로 인해 불균형해진 dev set 의 점수별 분포를 균일화하기 위해 과 포함된 라벨은 train set 으로 내보내고 과소 포함된 라벨은 train set 에서 dev set 으로 가져오는 방식으로 새로운 dev set 을 제작하였다. 이때 기존 KLUE 에서 dev set 을 제작할 때 사용한 방식과 동일하게 STS 모델이 단순 워드 오버랩기반으로 학습하는 경향성을

반영하지 않도록 3.0 미만의 낮은 유사도 점수일수록 워드 오버랩이 높은 데이터를, 반대로 3.0 이상의 높은 유사도 점수일수록 워드 오버랩이 낮은 데이터를 우선적으로 dev set 에 포함시켰다.

4. 데이터 유효성 검증과 모델 성능향상 방안모색

4.1 실험에 사용할 모델 선정

Model trained with	**Pearson R**	**F1 score**	**MSE**
Bi-encoder / *KLUE-RoBERTa-base*			
original klue-trainset	87.60	80.03	-
new trainset	88.08	80.65	-
Cross-encoder / *KLUE-RoBERTa-base*			
original klue-trainset	90.55	81.38	0.41
new trainset	**91.14**	**82.98**	**0.37**

[표 9]

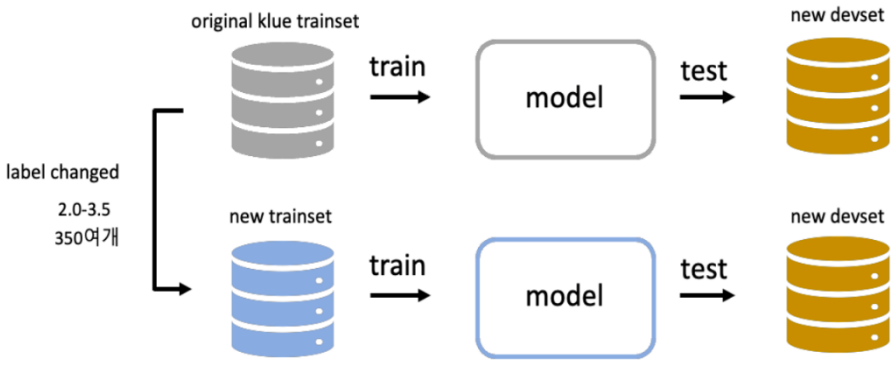
BERT 를 기반으로 한 문장유사도 평가 모델의 학습에 사용되는 방식은 크게 cross-encoding 과 bi-encoding 두 가지 방식으로 나뉜다. 모델의 성능 자체는 cross-encoding 방식이 더 뛰어나며 [표 9]처럼 실제 실험 결과에서도 cross-encoder 를 통한 학습이 bi-encoder 의 성능을 큰 폭으로 상회하는 것으로 나타났다. bi-encoding 방식은 연산이 가볍고 실제 시스템으로 활용되는 과정에서 많이 사용되지만 새로운 데이터셋을 제작하고 데이터의 변화에 따른 성능을 면밀하게 평가하는 task 에서는 cross-encoding 방식을 사용하여 각 평가지표의 정확한 측정을 하는 것이 더 적합하다고 판단했다.

Model trained with	**Pearson R**	**F1 score**	**MSE**
KLUE-BERT-base			
original klue-trainset	88.32	81.09	0.44
new trainset	89.62	80.30	0.41
KoELECTRa-base			
original klue-trainset	88.72	81.12	0.40
new trainset	90.54	**83.12**	**0.37**
KLUE-RoBERTa-base			
original klue-trainset	90.55	81.38	0.41
new trainset	**91.14**	82.98	**0.37**

[표 10]

데이터 학습에 사용할 사전학습모델은 BERT, ELECTRA, RoBERTa 세 가지 모델에 대해 성능을 비교 후 가장 좋은 성능을 보이는 모델을 선택했다. [표 9]를 보면 F1-score 에서는 ElECTRA 가 소폭 앞서긴 하지만 RoBERTa 와의 차이 폭이 크지 않고 STS 모델의 메인 평가지표인 피어슨 계수에서 RoBERTa 가 더 큰 폭으로 앞서기 때문에 RoBERTa 를 실험에 사용할 사전학습모델로 선택했다.

4.2 변경된 데이터의 유효성 검증



[그림 12]

기존 데이터에서 재 라벨을 통해 변화된 데이터가 얼마나 유효한가에 대한 검증 실험을 진행했다. 우선 변경된 데이터들이 새로운 기준에 맞게 제대로 라벨링 되었다면 새로운 데이터로 학습한 모델이 기존 데이터로 학습한 모델보다 새로운 테스트데이터에 대해 더 좋은 성능을 보일 것이라는 판단 하에 실험을 설계했다. 특히 선별한 데이터가 적은 수만 변경하더라도 모델에게

큰 영향을 미칠 것이라고 예상되는 데이터였기 때문에 타겟 구간으로 잡은 2.0-3.5 구간의 성능변화가 더 두드러질 것이라고 예상했다.

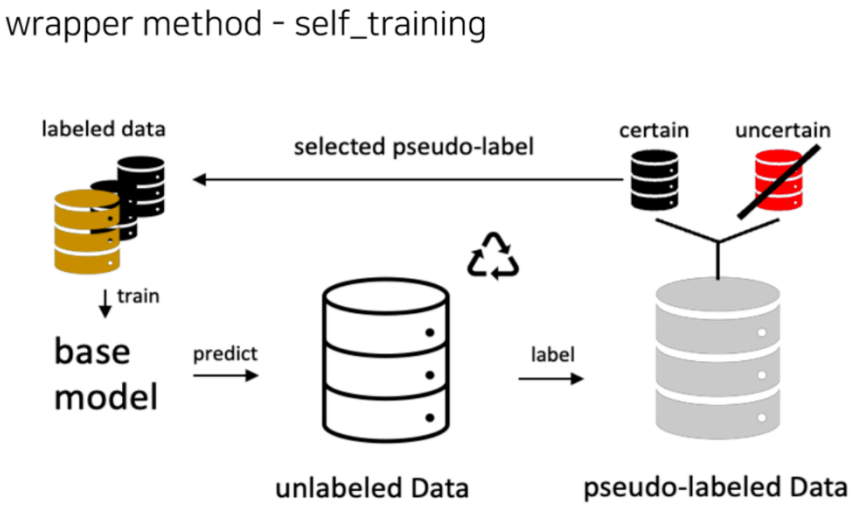
num_epochs	3
batch_size	32
learning_rate	5e-5
evaluation step	len(train) * 0.1
warmup_steps	len(train) * num_epochs / train_batch_size * 0.1
results	average of 10 random seed

[표 11]

평가지표는 KLUE leaderboard 에서 공식적으로 사용되는 피어슨 계수와 F1-score 에 100 을 곱한 값을 사용했고 추가로 타겟구간의 성능변화를 확인하기 위해 MSE 와 구간별 상관계수, acc 또한 평가지표로 활용하였다. 실험을 위해 모델 학습에 사용되는 하이퍼파라미터들을 고정하여 각 학습 단계가 동일한 환경에서 이루어지도록 모델학습을 진행했다. 모든 결과 수치는 10 개의 랜덤시드를 통해 가중치를 학습한 모델 결괏값의 평균을 기입했다. 원활한 실험을 위해 우분투 서버 환경의 8G 램 GPU 두 장을 사용하여 학습을 진행했다.

4.3 모델 성능 개선

4.3.1 semi-supervised learning

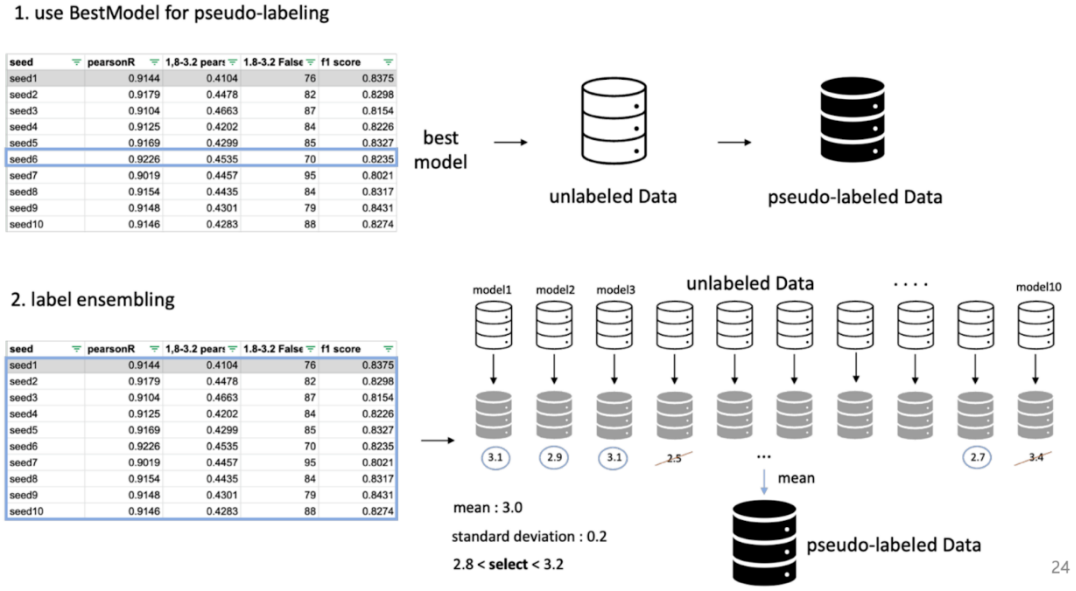


[그림 13]

라벨링 된 데이터의 유효성 검증을 마친 후 추가로 STS 모델의 성능을 개선하기 위해 다양한 방안을 적용해보았다. 그중 유의미한 결과를 얻은 self-training 을 통한 준지도학습기법을 활용한 성능 비교 실험을 진행했다. self-training 은 모델이 학습한 결과를 사용해 라벨이 되어있지 않은 새로운 데이터에 대해 pseudo-label 을 생성하여 부족한 학습 데이터를 증강하고 기존 라벨과 함께 재학습을 진행해 모델의 성능을 높이는 방식이라고 요약할 수 있다. 해당 방식을 라벨 불균형이라는 문제를 가지고 있는 STS 데이터에 적용하여 분포가 부족한 라벨 구간의 데이터를 증강함으로써 학습한 모델의 성능향상이 이뤄질 것이라는 가설을 세우고 실험을 진행했다.

다만 self-training 방식은 잘못된 pseudo-label 로 생긴 이상치에 민감하기 때문에 추가적인 라벨 검증작업이 필수적이다. 이러한 문제 때문에 classification task 에서는 학습이 이뤄진 모델의 soft-max 출력값을 바탕으로 확실성이 높은 pseudo-label 만 선별하는 과정이 존재하는데, 이 과정을 regression task 인 STS 모델에 적용하기 위해 새로운 라벨 선별 방식 또한 시도했다.

4.3.2 pseudo-label selection



[그림 14]

이상치에 민감한 self-training 방식의 단점을 보완하기 위해 불확실성이 높은 라벨을 선별하여 제거 후 평균값을 라벨로 사용하는 방식을 사용했다. 라벨 선별과정을 거친 학습 과정과 그렇지 않은 방식을 비교하기 위해 증강이 이뤄지기 전 데이터에 대해 10 개의 개별 시드로 학습된 모델 중 가장 높은 성능지표를 보이는 하나의 모델로 라벨링 한 결과와 10 개의 각 모델이 출력하는 예측값에서 표준편차에 따라 이상치들을 제거해준 뒤 평균값을 사용한 결과를 비교하여 실험했다.

4.3.3 증강 구간설정

bin	TP	TN	FP	FN	accuracy
1.3 - 1.7	0	44	2	0	0.956521
1.8 - 2.2	0	39	12	0	0.764706
2.3 - 2.7	0	33	30	0	0.523809
2.8 - 3.2	25	3	19	15	0.451613
3.3 - 3.7	58	0	0	4	0.935484
3.8 - 4.2	54	0	0	1	0.981818
4.3 - 4.7	55	0	0	0	1.0
4.8 - 5.0	22	0	0	0	1.0

[표 12]

학습에 사용된 데이터셋의 점수분포와 점수대 구간별 모델 성능을 고려하여 데이터 증강이 적용될 구간을 선택했다. 전체 train set 의 점수 분포 중 데이터의 수가 부족한 구간은 1.0-3.2 구간이었고 이 중 학습시 유사도 binary 판단에 대한 정확도가 특히 낮은 구간인 1.8-3.2 구간이 데이터 증강이 필요한 구간이라고 판단했다.

해당 구간에 pseudo-label 로 추가해줄 라벨이 되어있지 않은 데이터는 KorSTS 의 test set, ExoBrain 의 paraphase dataset(paraKAIST) 두 가지를 사용했다. KorSTS 는 사람의 번역검수가 이루어진 test set 의 1000 여 개의 데이터만 사용했고 paraKAIST 는 2000 여개의 전체 문장 쌍 데이터를 전부 사용했다. 3000 여 개의 데이터에 기존 데이터로 학습된 모델을 통해 pseudo-labeling 을 진행했고 이들 중 1.8-3.2 로 판단된 데이터들을 추출하여 각각 480 여 개, 360 여 개의 데이터를 추가적으로 학습시킬 수 있는 형태로 가공했다.

5. 연구 결과

5.1 변경된 데이터의 유효성 검증 실험 결과

original dataset vs **new dataset** evaluation

Model trained with	**Pearson R**	**F1 score**	**MSE**	**2.0-3.5 Pearson**	**2.0-3.5 ACC**
Cross-encoder / KLUE-RoBERTa-base					
original klue-trainset	90.55	81.38	0.41	39.42	59.36
new trainset	**91.14**	**82.98**	**0.37**	**43.73**	**62.68**

[표 13]

재 라벨링을 통해 생성한 새로운 데이터셋의 유효성 검증 실험 결과는 [표 13]과 같다. 기존 데이터셋을 라벨을 교정해준 새로운 데이터셋과 비교하여 실험했을 때 전체 라벨 구간의 피어슨 계수는 평균적으로 약 0.6 퍼센트, F1-score 는 약 1.6 퍼센트 상승하며 적은 수의 라벨데이터

변화에도 모델 성능에 유효한 변화가 있음이 확인되었다. 특히 타겟으로 잡아 라벨을 변경해준 구간인 2.0-3.5 구간의 성능이 더 큰 폭으로 상승하며 실험설계 시 제시한 가설을 만족하는 것을 확인할 수 있었다.

5.2 모델 성능개선 실험 결과

Model trained with	**Pearson R**	**F1 score**	**1.8-3.2 Pearson R**	**Data Augmentation**	**Number of Augmented data**	**Aug range**
without data augmentation						
original klue-trainset	90.55	81.38	39.45	x	x	x
new trainset	91.14	82.65	41.75	x	x	x
with data augmentation						
use best-model prediction	91.61	83.57	43.59	+ korsts-test	485	1.8-3.2
label ensemble	91.98	83.24	46.44	+ korsts-test	485	1.8-3.2
use best-model prediction	91.57	**84.04**	46.60	+ korsts-test & paraKAIST	851	1.8-3.2
label ensemble	**92.05**	83.30	**49.53**	+ korsts-test & paraKAIST	847	1.8-3.2

[표 14]

데이터 증강을 통한 모델 성능향상 실험 결과는 [표 14]과 같다. [표 14]의 위쪽 두 행은 데이터 증강을 통한 준지도학습기법을 적용하지 않은 모델의 성능 결과, 밑 쪽 네 행은 증강을 적용한 결과이다. 피어슨 계수와 F1-score, 구간별 피어슨 계수 모두 데이터 증강을 적용했을 때 상승하는 것으로 보아 self-training 을 이용한 준지도학습방식이 유의미한 성능변화를 이끌어낸다는 사실을 확인할 수 있었다. 또한 데이터 증강을 적용한 네 가지 실험 결과 내에서도 피어슨 계수의 경우 앙상블을 통해 pseudo-label 의 이상치를 제거한 결과가 단순히 최고성능을 보인 하나의 모델 라벨을 사용한 것 보다 소폭의 성능향상을 보인다. 반면 F1-score 의 경우 오히려 성능이 소폭 하락하는 결과가 도출되었다. 이는 라벨 앙상블에 사용된 모델 수가 부족하여 유사도 binary 판단에 부정적인 영향을 주는 라벨들이 많이 포함된 것으로 판단되며 추후 모델 수를 더 늘려 추가적인 실험을 진행한다면 F1-score 또한 피어슨 계수와 같이 성능향상이 이뤄질 것으로 예상된다.

6. 연구 의의와 한계

본 연구는 기존 한국어 STS task 에서 문제로 지적되던 라벨 기준의 모호성을 해결하기 위해 명확한 척도를 도입하여 기계와 인간 모두 라벨링 난이도가 높은 점수 구간에 대한 명확한 기준을 정립했다. 이 기준을 바탕으로 기존 한국어 벤치마크로 사용되는 KLUE-STS 의 데이터들을 재 라벨 하여 양질의 라벨데이터를 제작하였고 적은 수의 데이터 교정만으로도 모델에 유의미한 성능변화를 이끌어냈다. 또한 데이터 증강과 관련된 다양한 방법론을 직접 실험을 통해 적용해보며 추가적인 모델 성능향상을 이끌어내기도 했다. 이 같은 결과물들을 활용해 추가적인

라벨 데이터들을 제작할 수 있다면 기존 벤치마크의 문제점을 교정한 새로운 한국어 NLU 벤치마크로 발전할 가능성 또한 기대할 수 있다.

반면 여전히 해결하지 못한 문제들로 인한 연구의 한계도 존재한다. 우선 여러 가지 제약으로 인해 초기 계획했던 데이터 구간을 전부 재라벨링으로 교정해주지 못하고 한정적인 수의 데이터만 라벨링을 진행했다. 이 때문에 초기 예상했던 것만큼 큰 폭의 성능변화가 나타나지 못했다. 데이터 증강기법을 통해 어느 정도의 라벨 균형을 맞춰주었음에도 학습데이터는 여전히 라벨별 분포가 불균형하기 때문에 정립한 기준을 바탕으로 추가적인 라벨데이터를 제작하는 과정이 추가되어야 더 큰 성능변화를 이끌어 낼 수 있다. 또한 정립한 라벨 기준은 기존 방식보다 작업 난이도가 높기 때문에 라벨작업자들을 이해시키는데 추가적인 비용이나 시간이 소요되는 등 어려움이 많을 것으로 예상된다. 따라서 작업자들이 알아듣기 쉬운 언어로 최대한 표현을 정제하고 가이드 세션을 편성하는 등 실제 라벨링 과정에서 사용되기 위한 추가적인 노력이 필요할 것으로 예상된다.

References

Papers

Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks(2019)

Augmented SBERT: Data Augmentation Method for Improving Bi-Encoders for Pairwise Sentence Scoring Tasks(2020)

KLUE: Korean Language Understanding Evaluation(2021)

KorNLI and KorSTS: New Benchmark Datasets for Korean Natural Language Understanding(2020)

SemEval2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation

A survey on semi-supervised learning(2020)

Learning with pseudo-ensembles(2014)

Pseudo-Labels Are All You Need(2022)

2021 년 유사 문장 생성 말뭉치 연구 분석 사업

유사말뭉치 분석을 통한 유사도 인식에 관한 연구(2021)

Others

https://www.datacamp.com/tutorial/active-learning

https://blog.est.ai/2020/11/ssl/

https://github.com/UKPLab/sentence-transformers

https://github.com/Huffon/klue-transformers-tutorial/blob/master/sentence_transformers.ipynb

https://pseudo-lab.github.io/klue-baseline/docs/Semantic%20Textual%20Similarity.html

https://www.youtube.com/watch?v=WS1uVMGhlWQ

https://www.youtube.com/watch?v=aSx0jg9ZILo