한국어 문장 임베딩의 언어적 속성 입증 평가

안애림○, 고병일, 이다니엘, 한경은, 신명철, 남지순†

카카오엔터프라이즈, 한국외국어대학교†

{eileen.aa, kobi.k, daniel.e, grace.one, index.sh}@kakaoenterprise.com, namjs@hufs.ac.kr†

A Probing Task on Linguistic Properties of Korean Sentence Embedding

Aelim Ahn○, Byeongil Ko, Daniel Lee, Gyoungeun Han, Myeongcheol Shin, Jeesun Nam†

KakaoEnterprise, Hanguk University of Foreign Studies†

요 약

본 연구는 한국어 문장 임베딩(embedding)에 담겨진 언어적 속성을 평가하기 위한 프로빙 태스크(Probing Task)를 소개한다. 프로빙 태스크는 임베딩으로부터 문장의 표층적, 통사적, 의미적 속성을 구분하는 문제로 영어, 폴란드어, 러시아어 문장에 적용된 프로빙 테스크를 소개하고, 이를 기반으로하여 한국어 문장의 속성을 잘 보여주는 한국어 문장 임베딩 프로빙 태스크를 설계하였다. 언어 공통적으로 적용 가능한 6개의 프로빙 태스크와 한국어 문장의 주요 특징인 주어 생략(SubjOmission), 부정법(Negation), 경어법(Honorifics)을 추가로 고안하여 총 9개의 프로빙 태스크를 구성하였다. 각 태스크를 위한 데이터셋은 '세종 구문분석 말뭉치'를 의존구문문법(Universal Dependency Grammar) 구조로 변환한 후 자동으로 구축하였다. HuggingFace에 공개된 4개의 다국어(multilingual) 문장 인코더와 4개의 한국어 문장 인코더로부터 획득한 임베딩의 언어적 속성을 프로빙 태스크를 통해 비교 분석한 결과, 다국어 문장 인코더인 mBART가 9개의 프로빙 태스크에서 전반적으로 높은 성능을 보였다. 또한 한국어 문장 임베딩에는 표층적, 통사적 속성보다는 심층적인 의미적 속성을 더욱 잘 담고 있음을 확인할 수 있었다.

주제어: 문장 임베딩, 프로빙 태스크, sentEval

1. 서론

임베딩(embedding)이란 자연어의 단어, 구, 문장과 같은 단위를 기계가 이해할 수 있는 일련의 숫자로 변환, 특정 차원의 벡터(vector)로 추상화된 정보를 표현하는 것을 말한다.[1] 이는 딥러닝 기반의 언어처리에서 가장 기본이 되며, 실제로 임베딩을 활용한 문서 요약, 기계번역, 의미 분석 등의 분야가 큰 진전을 보이고 있다. 이처럼 임베딩의 적용으로 자연어처리 기술이 높은 수준으로 발전했음에도 불구하고, 임베딩에 담겨 있는 의미, 문법적 정보를 정확하게 이해하기란 어렵다. 특히 무한한 길이로 생성 가능하며 형태, 통사, 의미적으로 다양한 유형의 언어적 속성을 가지는 문장 정보를 유한한 차원의 벡터에 다 담아낼 수 있는지에 대해 의구심을 갖지 않을 수 없다. 많은 경우 사전 학습(pre-train)된 임베딩을 입력으로 하는 딥러닝 모델의 성능 평가나 KLUE[1]와 같은 벤치마크를 활용하여 임베딩의 품질을 간접적으로 평가하고 있다. KLUE에서 다루는 의미 추론, 문장 유사도 분류, 감성 분류, 질의응답 등과 같은 '다운스트림 태스크(downstream task)[2]' 문제의 경우 해결하는 과정이 매우 복잡하기 때문에 어떤 정보에 의해서 그 결론에 이르렀는지 알기 어렵다. 이는 과제에 따라 파인 튜닝(fine-tuning)된 모델이 얼마나 잘 작동하는지를 평가할 뿐이라서 임베딩과 같은 사전 학습 언어모델(Pre-trained Language Model)의 속성 평가로 보기는 어렵다.

임베딩에 담겨 있는 언어적 속성을 분석하고 이해하기 위한 시도로 [2]는 '프로빙 태스크(Probing Task)'를 고안하였다. 프로빙 태스크는 임베딩을 통해 문장이 가지는 간단한 언어적 자질을 분류하는 문제이며, 다운스트림 태스크의 복잡한 요인들에 비해 실험이 명확하고 간단하므로 결과의 해석이 쉽다. 총 10개의 프로빙 태스크를 설계하였고 SentEval[3]을 통해서 실험 문장과 툴킷(toolkit)을 공개하였다. 이는 문장 임베딩의 언어적 자질을 연구하기 위한 보편적인 방법론으로 영어뿐 아니라 다양한 언어에 적용되고 있다. [3]은 폴란드어를 대상으로 9개의 프로빙 태스크뿐 아니라 두 문장 간의 관련성(relatedness)와 함의(entailment)를 추론하는 다운스트림 태스크를 포함하여 실험하였으며, [4]은 총 14개의 프로빙 태스크를 러시아어에 적용하여 평가하였다.

본 연구에서는 한국어 문장 임베딩을 대상으로 프로빙 태스크를 수행하였다. 다른 언어에 적용된 태스크 중 한국어 특성에 적용 가능한 태스크를 선별하고, 한국어 문장의 속성을 잘 보여주는 태스크를 추가로 설계하여 총 9개의 프로빙 태스크를 적용하였다. 데이터셋은 구구조 문법(phrase-structure grammar)이 적용된 '세종 구문분

[1] https://github.com/KLUE-benchmark/KLUE

[2] 다운스트림 태스크(Downstream Task)란 해결하고자 하는 자연어처리의 구체적인 문제들을 의미하며 품사 판별, 개체명 인식, 의미역 분석 등이 있다. 반면 업스트림 태스크(Upstream Task)는 다운스트림 태스크를 하기 위해 이루어져야 하는 과제로 임베딩이 여기에 해당된다.[1]

[3] https://github.com/facebookresearch/SentEval

석 말뭉치’ [5]를 의존구문문법(Universal Depende ncy Grammar)[6]로 변환하여 자체 구축한 KUDC(Korean Universal Dependency Corpus)를 사용하였다. KUDC에 부착된 형태 및 통사 정보를 활용하여 각 프로빙 태스크에 적합한 데이터셋을 자동으로 구축하였다. 프로빙 태스크는 HuggingFace[4]에 공개된 4개의 다국어(multilingual) 문장 인코더와 4개의 한국어 문장 인코더로부터 한국어 문장 임베딩을 획득하여 실험하였다. 본 실험을 통해 인코더 간의 비교 분석이 가능하며, 임베딩이 담고 있는 정보를 파악하여 부족한 부분을 개선함으로써 다운스트림 태스크에서 더 나은 성능을 기대할 수 있을 것이다.

2. 관련 연구

임베딩에 담겨진 언어적 속성 평가에 관한 초기 연구는 [7]로 문장 임베딩을 통사적 속성에 따라 분류하는 문제로 이를 검증하였으며, 이 방법론은 프로빙 태스크로 발전하였다. 이후 [2]에서는 영어 문장 임베딩을 대상으로 언어적 속성을 평가할 수 있는 태스크를 설계하고, 여러 유형의 인코더 성능을 비교하였다. 실험 문장은 구구조 문법이 적용된 Toronto Book Copus로부터 수집하였다. 5~28개의 단어로 구성된 문장만 사용하였으며, 각 태스크의 학습셋은 100,000 문장으로 구축되었다. 인코더는 BiLSTM-last/max, Gated ConvNet로 총 10개의 태스크를 수행하였으며, 각 프로빙 태스크와 다운스트림 태스크 간의 상관관계를 밝히기도 하였다. [3]은 어순이 비교적 자유로운 폴란드어의 속성에 따라 의존 구문 정보가 부착된 4백만 쌍의 영어-폴란드어 병렬코퍼스에서 실험 문장을 추출하였다. 인코더 FASTEXT-max/mean, BERT-max/mean, COMBO -max/mean, SENT2VEC -NS/ORIG, LASER, USE를 대상으로 각 90,000 문장을 데이터셋으로 하는 9개의 프로빙 태스크를 실험하여 영어와 폴란드어 임베딩 결과를 비교하였다. [4]는 러시아 문장 임베딩을 대상으로 120,000 문장을 데이터셋으로 하는 14개의 프로빙 태스크를 실험하였다. 평가 인코더는 HuggingFace에 공개된 다국어 처리가 가능한 M-BERT, XLM-R, MiniLM, LABSE, M-BART이다. 표 1은 각 연구에서 진행한 프로빙 태스크를 비교한 표이다.

표 1. 언어별 실험 인코더 및 프로빙 태스크 유형

	Conneau(2018)	Krasnowska-Kiras (2019)	Mikhailov(2021)
적용 언어	영어	폴란드어	러시아어
인코더	BiLSTM-last/max, Gated ConvNet	FASTEXTmax/mean, BERT-max/mean, COMBO-max/mean, SENT2VEC-NS/ORIG, LASER, USE	BERT, XLM-R, MiniLM, LABSE, M-BART
Probing Task			
표층적 속성	Sentence Lenth, Word Content	Sentence Lenth, Word Content	Sentence Lenth, Word Content
통사적 속성	Bigram Shift, Tree Depth, Top Constituent	Tree Depth, Top Dependency, Passive	N-gram Shift, Tree Depth, Conjunction Type, Impersonal Sentence, Gapping
의미적 속성	Tense, Subject Number, Object Number, Semantic Odd Man Out(SOMO), Coordination Inversion	Tense, Subject Number, Object Number, Sentence Type	Tense, Subject Number, Object Number, Subject Gender, Object Gender, Predicate Voice, Predicate Aspect

3. 한국어 프로빙 태스크(Probing Task)

한국어 문장 임베딩 평가를 위해 기존 연구에서 고안된 프로빙 태스크 중 적용 가능한 태스크를 선별하고, 한국어 문장의 속성을 잘 반영하는 3가지 태스크를 추가로 설계하여 총 9개의 프로빙 태스크를 시행하였다.

3.1. 데이터셋

프로빙 태스크를 위한 한국어 문장은 기구축된 말뭉치에 부착된 정보를 활용하여 자동으로 추출하였다. 약 76,000여 문장으로 구성되어 있으며 형태, 통사 정보가 부착된 ‘세종 구구조 구문분석 말뭉치’를 유니버셜 의존 문법(Universal Dependency Grammar) 형태로 자동 변환하여 구축한 KUDC(Korean Universal Dependency Corpus)를 사용하였다. 어순이 비교적 자유롭고, 생략이 자주 발생하는 한국어 문장의 특성상 의존 문법을 적용하는 것이 용이하기 때문이다. 전체 문장 중에 5어절 이상 28어절 이하의 문장만 선별하여 총 56,455개의 문장으로만 구성하였다. 다른 언어에 비해 한국어 말뭉치의 양이 매우 적은 한계로 인해 각 태스크별 문장의 개수는 다르게 구축되었으며, 추후 이와 관련한 말뭉치 확장이 요구된다. 또한 문장 유형(SentType)을 구분하는 태스크를 위해 문형 정보를 부착하여 자체 구축한 약 91,000개의 ‘카카오엔터프라이즈 대화 발화 코퍼스’로 데이터셋을 구축하였다. 해당 코퍼스에서는 문장을 서법에 따라 설명법, 감탄법, 의문법, 명령법 등으로 상세히 나누고 있고, 이 정보를 활용하여 ‘평서문, 의문문, 명령문, 기타’의 유형으로 약 62,114개의 문장을 구축하였다.

3.2. 표층적 속성

문장 임베딩이 한국어 문장의 표층적인 속성을 잘 학습하고 있는지를 평가하기 위한 실험으로, 문장을 구성하는 요소들에 의해 표층적으로 보여지는 속성을 구분하는 문제를 의미한다.

3.2.1. SentLen

SentLen(sentence length)은 문장의 길이를 구분하는 문제로, 어절 개수를 기준으로 문장의 길이를 추출한 후

[4] https://huggingface.co/

기존 연구와 동일한 기준을 적용하여 전체 문장을 6개의 분류로 나누었다. 전체 데이터셋은 56,455 문장으로 구성되었다.

표 2. SentLen 분류 및 어절 수

분류	0	1	2	3	4	5
어절 수	5~8	9~12	13~16	17~20	1~25	26~28

3.2.2. WC

WC(word content)는 전체 말뭉치로부터 중빈도 어휘를 미리 선정하고, 주어진 문장을 중빈도 어휘에 따라 분류하는 문제이다. 이는 임베딩으로부터 자연어 문장에 존재하는 어휘 정보를 기억하고 있는지를 평가하는 태스크이다. 영어 문장 실험의 경우 전체 어휘 중 2001번째 빈도에 해당하는 어휘부터 1,000개를 선택하였으나, 한국어 문장은 소스 데이터셋의 양이 적어 동일한 방식을 적용할 경우 충분한 수의 문장을 수집하기 어려웠다. 따라서 전체 말뭉치의 어절 중 200 이상 600 이하의 빈도를 가지며, 두 개 이상의 어휘 중복이 없는 100개 이상의 문장을 학습셋으로 가지는 92개의 어휘를 추출할 수 있었다. 이들이 포함된 문장으로 구성된 데이터셋은 12,634개의 문장이다.

(1) a. 해가 뉘엿뉘엿 기울 무렵이 **되었다**. /되었다
b. 남편 일 **너무** 걱정은 마십시오. /너무
c. 승주의 가장 큰 고민은 **역시** 군대문제이다. /역시
d. 나는 **눈을** 동그랗게 떴다. /눈을

3.3. 통사적 속성

문장 임베딩이 한국어 문장의 통사적인 속성을 잘 학습하고 있는지를 평가하기 위한 실험이다.

3.3.1. TreeDepth

TreeDepth(tree depth)는 문장의 계층 구조를 추론하여 통사 구조의 최대 깊이를 분류하는 문제이다. 언어에 따라 통사 분석 문법에 차이가 있지만, 통사 구조의 깊이를 추출하는 방법론은 동일하다. 루트 노드(root node)에서 하위 노드로 내려갈 때마다 깊이가 하나씩 늘어나고 단말 노드(leaf node)까지의 최대 깊이를 계산하는 방식이다.

그림 1. 한국어 의존 구문 분석의 예

표 3 Tree Depth 계산의 예

```
1 빼놓는다 - 2 얼을 - 3 관광객들의
1 빼놓는다 - 2 설치돼 - 3 조명등이 - 4 화려한
1 빼놓는다 - 2 설치돼 - 3 조명등이 - 4 개의 - 5 수천
1 빼놓는다 - 2 설치돼 - 3 밤마다
1 빼놓는다 - 2 설치돼 - 3 공원에는 - 4 열리는 - 5 축제가 - 6 눈
```

그림 1의 문장에서 루트 노드 '빼놓는다'로부터 하위 노드를 따라 단말 노드까지 가는 경로는 표 3과 같고, 이때 하위 노드의 개수를 세면 통사 구조의 깊이가 된다. 위 경로 중 최대 깊이는 6이며, 해당 문장의 속성으로 6이 부착된 데이터셋이 구축된다. 통사 구조의 깊이가 너무 짧거나 긴 경우 사실상 문장의 길이와 연관성이 높기 때문에 4 이상 9 이하의 깊이에 해당하는 문장만 실험 문장으로 사용하였고 총 46,628개 문장이 구축되었다.

3.3.2. TopDeps

TopDeps(top dependency sequence)는 최상위 구성 성분인 루트 노드의 하위노드를 예측하는 문제이다. 이는 보통 문장 술어의 논항에 해당하며, 각 논항의 문장 성분으로 최상위 구성 성분 태그열을 구축하게 된다. 그림 1 문장의 루트 노드 '빼놓는다'의 의존소는 '설치돼/advcl'과 '얼을/obj'로 TopDeps는 'advcl_obj'가 된다. 이렇게 추출된 TopDeps 중에서 고빈도 19개 태그열을 추출하고, 나머지는 OTHER로 묶어 구축한다. 추가로 OTHER의 문장 수가 다른 클라스의 문장 수보다 월등하게 많은 것을 막기 위해서 전체 문장 수의 5%로 제한하였다. 이는 영어의 구구조 문법을 기반으로한 TopCont 실험과 동일한 방법을 적용한 것이다. 전체 실험 문장은 16,689 문장이며, 표 4의 태그열로 분류된다.

표 4. 한국어 TopDeps

nsubj, nsubj_obl, advcl_nsubj, acl, obl_nsubj, nsubj_acl, nsubj_advcl_obj, nsubj_obj, advcl, advcl_obj, nsubj_advcl, compound, nsubj_obl_obj, advcl_nsubj_obl, xcomp, nsubj_advcl_obl, advcl_obl, nsubj_advmod, advmod_nsubj, OTHER

3.3.3. SubjOmission

한국어 문장이 가지는 중요한 통사적 속성 중 하나는 문장 성분의 생략이 자유롭다는 것이다. 특히 주어의 생략이 가장 자주 발생하는데 본 연구에서는 이러한 특성을 반영하여 SubjOmission(subject omission) 태스크를 고안하였다. 문장의 통사 구조를 추론하여 문장의 주어 생략이 발생하는지를 판단하는 문제이다. 루트 노드에 해당하는 술어의 주어를 기준으로 생략 여부를 판단하며, 전체 22,267개의 문장으로 데이터셋을 구축하였다.

(2) a. **마녀는** 지금 빗자루를 타고 하강하고 있다. /주어있음
b. 그리고 거울을 향해 잔뜩 얼굴을 찌푸렸다. /주어생략

그 외 다른 연구에서 진행한 문장 내 인접어들 간의 순서 변화를 예측하는 실험(BShift)의 경우 문장 내 어휘 순서가 비교적 자유로운 한국어에서 인접어들 간의 순서가 변경되어도 통사적으로 정상적인 문장으로 받아들여지는 경우가 많기 때문에 제외하였다. 또한 문장의 태를 예측하는 문제(Passive)의 경우 영어에서는 수동을 예측하는 통사 패턴이 존재하지만 한국어는 통사적인 구성 뿐만 아니라 사동 접사와 결합하여 새롭게 파생된 단어를 사용하기 때문에 통사적 속성을 평가하는 실험으로 적절하지 않아 제외하였다.

3.4. 의미적 속성

문장 임베딩이 한국어 문장의 의미적 속성을 잘 학습하고 있는가를 평가하기 위한 실험으로, 문장 내 요소들의 의미적인 속성과 관련이 있다.

3.4.1. Tense

Tense는 주절의 서술어 시제를 예측하는 문제로 한국어 문장은 '-었-/-았-'과 같은 문법적 요소에 의해 과거 시제를 구분한다. 한국어 문법에서는 현재 시제에 미래의 일까지 포함하기 때문에 비과거 시제라고 하기도 한다.[8] 따라서 '과거'와 '비과거'의 이분법 분류로 실험하였으며, 데이터는 총 56,454개 문장으로 구성되었다.

(3) a. 그게 언제나 그가 하는 식이지. /비과거
b. 논 뜰에서 개 아버지를 만났어. /과거

3.4.2. SentType

SentType(sentence type)은 문장의 유형을 예측하는 문제이다. 한국어에서는 종결어미에 따라서 문장의 유형이 나뉘는 특성 때문에 의미적 속성으로 볼 수 있다. 한국어의 문장형은 평서문, 의문문, 명령문, 청유문의 네 가지 형태이며, 그 외에 감탄문, 허락문, 약속문 등을 더 나누는 경우도 있으나 이들은 명시적으로 나뉠 만큼 어미 체계를 갖추고 있지는 않다.[8] 따라서 SentType 분류 클래스는 의문문, 명령문, 기타(평서문, 청유문, 감탄문 등)로 분류하였다. 데이터셋을 위한 문장 추출은 '카카오엔터프라이즈 대화 발화 코퍼스'의 '서법' 분류를 기반으로 이루어졌으며 총 62,114 문장으로 이루어졌다.

(4) a. 판교에 맛집 있으면 좀 알려줘. /명령문
b. 그럼 재즈는 어때요? /의문문
c. 이젠 정말 슬프지도 않다. /기타

3.4.3. Negation

본 연구에서는 문장의 의미를 부정문과 긍정문으로 나누는 Negation 태스크를 추가 고안하였다. 복문의 경우 주절을 기준으로 판단하며, KUDC에 부정의 의미를 가지는 관계 태그를 활용하여 자동으로 추출하였다. 한국어에서는 '안, 못'과 같은 부정 부사와 '않다, 말다, 못하다'와 같은 보조용언 또는 서술어에 의해 부정의 의미를 드러낸다. 뿐만 아니라 '아니다, 없다, 모르다'와 같은 특수 부정사에 의해 부정의 의미가 첨가되기도 해서[8], 이들을 포함하는 문장을 부정문으로 나누었다. 전체 문장은 총 10,200개이다.

(5) a. 아이는 한참 동안 아무 말도 **못** 했다. 부정
b. 어제와 오늘과 내일은 따로 있는 것이 **아니다**. 부정
c. 절을 하는 이도, 받는 이도 **없었다**. 부정
d. 그는 다급한 목소리로 옆자리 짝에게 묻는다. 긍정

3.4.4. Honorifics

언어마다 중시하는 속성이 큰 차이를 보이는데, 주어, 목적어의 수(SubjNum, ObjNum) 나 주어, 목적어의 성(SubjGender, ObjGender)을 중시하는 언어가 있는 반면에 한국어의 경우에는 이러한 속성은 중요하지 않다. 한국어에서 중요하게 여기는 속성 중 하나는 경어법이다. 한국어의 경어법은 정교하게 세분되어 있고 매우 체계적으로 발달해 있다.[8] 본 연구에서는 이러한 특성을 반영하여 문장을 경어 여부에 따라 나누는 Honorifics 태스크를 추가하였다. 경어법은 (6)과 같이 존대의 대상에 따라서 주체경어법, 객체경어법으로 나뉘어지고, '-시-/-으시-, '-께서'등과 같은 문법 형태소나 '잡수시다, 주무시다, 편찮으시다 등'의 특수 높임말에 의해 실현된다. 또한 '상대경어법'중 청자의 신분을 높이는 '해요체, 합쇼체'가 적용된 문장은 경어 문장으로 분류하였다. 본 연구에서 '경어 문장'과 '비경어 문장'의 이분법 실험으로 진행되었으며, 데이터 셋으로는 9,554개 문장이 구축되었다.

(6) a. 선생님이 부르**시**는구나. /주체경어법
b. 선생님께 편지를 **드렸다**. /객체경어법
c. 당신 생각이 맞**아요**. /상대경어법(해요체)
d. 당신 생각이 맞**습니다**. /상대경어법(합쇼체)

[2]는 문장 내 명사 또는 동사를 다른 단어로 치환하고, 문장의 의미가 적절한지를 판단하는 실험인 SOMO(semantic odd man out)[5]와 대등절 구문에서 문장의 순서를 바꾼 후 문장의 의미가 적절한지를 판단하는 CoordInv(coordination inversion)[6] 태스크를 실험했다. 이러한 프로빙 태스크도 한국어에도 적용이 가능할 것으로 보이지만, 본 연구는 기구축된 말뭉치를 활용하여 실험 문장을 자동으로 구축하는 방식으로 이루어졌기 때문에 적용 가능한 말뭉치의 부재로 제외되었다. 위 실험은 그럴듯한 구조의 문장에서 의미적인 오류를 알아차려야 하거나, 담화의 화용론적인 요소를 이해하는 등의 매우 높은 수준의 의미적인 이해를 평가하는 실험으로 차후 추가되어야 할 것으로 보인다.

표 5 언어별 프로빙 태스크 비교

	영어	폴란드어	러시아어	한국어
문장 길이	SentLen	SentLen	SentLen	SentLen
어휘 구분	WC	WC	WC	WC
통사구조 깊이	TreeDepth	TreeDepth	TreeDepth	TreeDepth

[5] Conneau(2018)의 SOMO(semantic odd man out)실험은 무작위로 문장 내 명사와 동사 중 한 어휘를 선택하여 다른 어휘로 치환한다. 의미 파악 없이 인접어들 간의 빈도로 문장이 적절한지를 판단하는 것을 막기 위해 치환하는 어휘는 인접어들과 유사한 빈도를 가지는 어휘로 선택하였다.

[6] CoordInv(Coordination Inversion)은 일의 순서와 같은 화용적인 의미를 이해를 요구하는 실험이다. 예를 들어 '나는 늦잠을 잤고 지각을 했다'의 대등절의 순서를 바꾸면 '나는 지각을 했고, 늦잠을 잤다'는 문장이 되고 이는 화용적으로 적절하지 않은 문장이다.

서술어의 논항	TopConst	TopDeps	-	TopDeps
문장 성분 생략	-	-	Gapping	Subj -Omission
어휘순서변화	BShift	-	NShift	-
접속사 유형	-	-	ConjType	-
비인칭 주어	-	-	Impersonal Sent	-
시제	Tense	Tense	PT	Tense
주어의 수	SubjNum	SubjNum	SubjNum	-
목적어의 수	ObjNum	ObjNum	ObjNum	-
주어의 성	-	-	SubjGender	-
목적어의 성	-	-	ObjGender	-
어휘 대치	SOMO	-	-	-
대등절 순서	CoordInv	-	-	-
태(수동/능동)	-	Passive	PV	-
상(완료/진행)	-	-	PA	-
문장 유형	-	SentType	-	SentType
부정/긍정	-	-	-	Negation
경어법	-	-	-	Honorifics

4. 실험

4.1. 인코더

본 연구에서는 HuggingFace에 공개된 4개의 다국어(multilingual) 문장 인코더와 4개의 한국어 문장 인코더를 실험하였다.

4.1.1. 다국어 문장 인코더

- **XLM-RoBERTa(XLM-R)**[10]는 2.5TB의 커먼 크롤(Common Crawl) 데이터를 이용하여 100개 언어에 대해 Masked Language Model(MLM)로 학습한 언어 모델이다.
- **mBART**[11]는 Seq2Seq 스키마와 Text Infilling와 같은 노이즈 함수를 적용한 BART의 다국어 모델이다. 커먼 크롤 데이터를 이용하였으며 25개 언어를 지원한다. 해당 실험에서는 인코더(encoder)의 마지막 레이어의 평균값을 추출(mean pooling)하여 사용하였다.
- **MiniLM**[12]은 XLM-R모델에 Self-Attention Distillation 방법을 적용하여 모델 사이즈를 가볍게 한 언어 모델이다.
- **laBSE**[13]은 Masked Language Model(MLM)과 Translation Language Model(TLM) 방식의 다국어 BERT로 초기화된 듀얼 인코더를 이용해 학습된 문장 임베딩 모델이다.

4.1.2. 한국어 문장 인코더

- **KcBERT**[7]은 구어체에 가까우며 잘 정제되지 않은 1.1억 개의 댓글 데이터셋을 사용하여 BERT모델을 학습한 한국어 언어 모델이다.
- **KoELECTRA**[8]는 34GB 크기의 뉴스, 위키, 신문, 문어, 구어, 메신저, 웹 등의 한국어 문서를 활용하여 ELECTRA 모델을 학습한 한국어 언어 모델이다.
- **KLUE-RoBERTa**[9]는 KLUE Benchmark에서 제공한 사전 학습 언어 모델로 모두의 말뭉치, 나무 위키, 뉴스, 청원 등 약 62GB의 한국어 문서를 가지고 RoBERTa를 학습하였다.
- **KoBART**[10]는 Text Infilling 노이즈 함수를 사용하여 40GB 이상의 한국어 문서를 학습한 한국어 인코더-디코더 모델이다. mBART와 동일하게 인코더의 마지막 레이어의 평균값을 추출하여 사용하였다.

4.2. 실험 방법

프로빙 태스크 실험은 [9]가 개발한 SentEval 툴킷을 사용한다. 이는 프로빙 태스크를 통해 임베딩 벡터를 평가할 수 있는 유틸리티로 데이터셋 문장을 실험 인코더를 통해 벡터로 변환하고, 그 벡터와 문장에 라벨링된 속성값을 가지고 Multi-Layer Perceptron(MLP)를 올린 분류기(classifier)를 학습시킨다. 프로빙 태스크에 따라 학습된 분류기에 임베딩 벡터를 입력으로 넣으면 속성값을 결과로 출력하여 평가하는 방식이다. 속성값이 잘 출력된다면 임베딩 벡터에 문장이 가지고 있던 언어적 속성이 잘 담겨져 있다고 판단할 수 있다.

4.3. 실험 결과

SentEval을 통한 프로빙 태스크 실험 결과는 표 6과 같다. 성능 평가 지표는 일반적인 정확도(accuracy)가 아니라 균형 정확도 점수(balanced accuracy score)[11]를 사용하였다. 각 실험의 클래스별 데이터셋이 동일한 수로 구축되지 않아 발생하는 데이터 불균형 문제를 해결하기 위함이다. 각 태스크마다 가장 높은 성능을 보이는 결과는 음영을 주었다.

표 6 프로빙 태스크 결과

	XLM-R	mBART	miniLM	laBSE	KcBERT	koELECTRA	KLUE-RoBERTa	KoBART
SentLen	43.57	**84.45**	31.55	53.69	69.93	55.32	59.94	71.96
WC	4.89	53.23	6.47	46.86	40.25	20.65	27.13	**62.31**
TreeDepth	25.04	**36.32**	23.26	30.34	30.21	30.21	30.50	34.39
TopDeps	5.00	28.19	7.75	**28.24**	11.84	5.07	9.99	22.50
SubjOmission	57.87	**76.27**	65.64	70.33	74.67	64.31	73.59	75.06
Tense	54.51	90.88	73.40	91.76	90.18	76.44	**91.93**	89.00
SentType	67.86	89.06	65.29	90.90	90.99	83.32	**93.62**	85.69
Negation	52.75	**86.86**	73.10	86.27	84.01	58.46	76.53	82.71
Honorifics	62.13	95.36	79.54	78.02	86.22	75.05	**97.50**	91.51
평균	**41.51**	**71.18**	**47.33**	**64.04**	**64.25**	**52.09**	**62.30**	**68.34**

실험 인코더 중 가장 좋은 성능을 보인 인코더는 mBART로, 대부분의 태스크에서 고른 결과를 보여주었다.

[7] https://github.com/Beomi/KcBERT
[8] https://github.com/monologg/KoELECTRA
[9] https://github.com/KLUE-benchmark/KLUE
[10] https://github.com/SKT-AI/KoBART
[11] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.balanced_accuracy_score.html

한국어 문장을 대상으로 하는 실험이었음에도 다국어 문장 인코더인 mBART의 성능이 가장 좋았다. 한국어 문장 인코더 중 모든 태스크에 대해 고르게 좋은 성능을 보인 인코더는 koBART이며 전체적으로 BART계열의 인코더의 실험 결과가 좋음을 알 수 있다. KLUE-RoBERTa의 경우 표층적, 통사적 속성 관련 프로빙 태스크의 결과는 좋지 않지만, 의미적 속성 관련 프로빙 태스크에서는 아주 좋은 결과를 보여주었다.

속성별 결과를 비교해보면 대부분의 인코더가 통사적 속성 구분에 낮은 정확도를 보여주는데, 이는 의존 문법 구조로 설계된 태스크에서 통사 구조에 대한 충분한 정보가 학습되지 않은 결과로 보여진다. 이러한 결과는 폴란드어를 대상으로 한 [3]의 실험에서도 유사하게 나타났다. 표층적 속성을 구분하는 태스크인 WC는 인코더들 간에 큰 편차를 보이며, 이는 임베딩에 담겨진 표층적인 정보의 양이 매우 다름을 의미한다. 또한 다른 속성에 비해 의미적인 프로빙 태스크 성능이 매우 높게 나타나는데, 이는 대부분의 인코더가 의미적인 문제 해결을 목적으로 학습되며, 특히 실험에 사용된 최종 레이어(last layer)에 표층, 통사적 특성보다는 추상적인 의미 정보가 잘 담긴 형태로 학습된 것으로 예측할 수 있다.[14]

5. 결론

본 연구에서는 문장 임베딩에 언어적 속성이 잘 반영되어 있는지를 평가하기 위한 방법론을 소개하였다. 한국어 문장 임베딩 평가를 위해 한국어 속성을 잘 보여주는 프로빙 태스크를 설계하고, 데이터셋을 자동으로 구축하여 실험하였다. 4개의 다국어 문장 인코더와 4개의 한국어 문장 인코더를 대상으로 9개의 프로빙 태스크를 실험하였으며, mBART와 같은 다국어 문장 인코더가 전반적으로 높은 성능을 보이는 것을 확인하였다. 또한 대부분의 인코더에서 표층적, 통사적인 속성보다는 의미적 속성을 더욱 잘 담고 있음을 알 수 있었고, 이러한 시도를 통해 한국어 문장 임베딩에 담겨진 언어적 속성을 면밀하게 분석해 볼 수 있었다.

본 실험의 데이터셋은 '세종 구문분석 말뭉치'를 기반으로 구축되었는데, 해당 말뭉치는 재배포를 허용하지 않기 때문에 아쉽게도 공개하지 못하였다. 차후 공개 가능한 데이터셋의 확장을 통해 한국어 SentEval의 리더보드 개발과 임베딩 모델에 대한 평가 지표를 제공될 수 있도록 추가 연구되어야 할 것이다.

참고문헌

[1] 이기창, "한국어 임베딩", 에이콘, 2019.

[2] Alexis Conneau, Germán Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni, "What you can cram into a single W$&!#* vector: Probing sentence embeddings for linguistic properties", In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL, 2018.

[3] Katarzyna Krasnowska-Kieras and Alina Wroblewska, "Empirical linguistic study of sentence embeddings", In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, ACL, 2019.

[4] Vladislav Mikhailov, Ekaterina Taktasheva, Elina Sigdel, Ekaterina Artemova, "RuSentEval: Linguistic Source, Encoder Force!", *Proceedings of the 8th BSNLP Workshop on Balto-Slavic Natural Language Processing*, ACL, 2021.

[5] 국립국어원, "21세기세종계획", 2012.

[6] Jayeol Chun, Na-Rae Han, Jena D.Hwang, Jinho D.Choi, "Building Universal Dependency Treebanks in Korean", *the Language Resources and Evaluation Conference(IREC)*, 2018.

[7] Xing Shi, Inkit Padhi, and Kevin Knight, "Does String-Based Neural MT Learn Source Syntax? ", *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing,* ACL, 2016

[8] 이익섭, 한국어문법, 서울대학교출판문화원, 2004.

[9] Alexis Conneau and Douwe Kiela, "SentEval: An Evaluation Toolkit for Universal Sentence Representations", *Computation and Language(CL)*, 2018.

[10] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzman, Edouard Grave, Myle Ott, Luke Zettlemoyer, Veselin Stoyanov, "Unsupervised Cross-lingual Representation Learning at Scale", *In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, ACL, 2020.

[11] Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, Luke Zettlemoyer, "Multilingual Denoising Pre-training for Neural Machine Translation", *Computation and Language(CL)*, 2020.

[12] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, Ming Zhou , "MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers",*Computation and Language(CL)*, 2020.

[13] Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, Wei Wang , "Language-agnostic BERT Sentence Embedding", *Computation and Language(CL)*, 2020.

[14] Yijin Liu1 , Fandong Meng , Jinchao Zhang , Jinan Xu1 , Yufeng Chen1 and Jie Zhou, "GCDT: A Global Context Enhanced Deep Transition Architecture for Sequence Labeling", *Computation and Language(CL)*, 2019.