한국어 BERT 모델 소개

순서

1. 공개된 한국어 BERT
2. 한국어 BERT 모델 구축 시 고려해야 하는 점
3. BERT를 넣고 돌릴 수 있는 한국어 downstream task

---

1. 공개된 한국어 BERT

- BERT를 개인이 직접 pre-training하기엔 너무 오래 걸리고 비싸다.
- 그러므로 이미 공개된 모델을 다운 받아서 우리가 풀고자 하는 task에 fine tuning하면 좋다.
- 우리가 사용할 수 있는 모델 위주로 정리해 봄. ~~(이름 헷갈림 주의)~~

BERT-base multilingual cased

(https://github.com/google-research/bert) (google)

- 104 개의 언어 처리를 위한 모델
- spec
  - Hidden size = 768, Layers = 12, Heads = 12, vocab_size = 119547
- Wordpiece tokenizer
- 이후 나온 한국어 특화 모델들에 비해 한국어 처리에서는 성능이 떨어짐.

KLUE BERT base

(https://huggingface.co/klue/bert-base)

(https://klue-benchmark.com/)

- KLUE 논문을 쓴 팀의 github에 공개된 모델

    - KLUE: 영어의 GLUE 처럼 한국어 자연어처리 모델 간 성능 비교를 위해 구축된 평가 셋
- 모두의말뭉치, CC-100, 나무위키, 뉴스 크롤링, 청와대 국민청원 크롤링 데이터로 pre-train
- morpheme-based subword tokenization
    - 형태소 단위로 초벌 토크나이징 후 BPE 알고리즘 적용
- spec
    - Hidden size = 768, Layers = 12, Heads = 12, vocab_size = 32000

Kortok

(https://github.com/kakaobrain/kortok)

(https://arxiv.org/abs/2010.02534)

- kakaobrain
- tokenization 방식에 따른 다양한 BERT 모델을 공개함.
- 한국어 위키 + 나무위키로 pre-train
- 다양한 토크나이징 방식 중에서 mecab + sentencepiece tokenization 방식의 모델이 성능이 가장 뛰어나다고 보고함.
- spec
    - Hidden size = 768, Layers = 12, Heads = 12, vocab_size = 32000, 64000

KoBERT

(https://github.com/SKTBrain/KoBERT)

- SKT
- 한국어 위키로 pre-train
- spec
    - Hidden size = 3072, Layers = 12, Heads = 12, vocab_size = 8002

KR-BERT

(https://github.com/snunlp/KR-BERT)

(https://arxiv.org/pdf/2008.03979.pdf)

- BERT 구축 시 시간, 비용 너무 많이 드니까 비교적 성능을 잘 내면서 더 작은 모델이 필요하다.
- 한국어 위키, 뉴스 기사로 pre-train
- 자체 제작 Bidirectional WordPiece tokenizaer 사용 (다른 BERT 모델들은 대부분 wordpiece 또는 sentencepiece 사용)
- 한글을 음절 단위, 자모 단위로 분해하는 tokenizing 방식 채택 → 성능 향상
- spec
  - vocab size = 16424(character), 12367(sub-character)

KcBERT

()

- 구어체에 강인하도록 네이버 뉴스 댓글 데이터로 pre-train
- spec
  - Hidden size = 768, Layers = 12, Heads = 12, vocab_size = 30000

---

2. 한국어 BERT 모델 구축 시 고려해야 하는 점

1. 영어보다 한정된 데이터의 양
   - pre-training 시 대부분 한국어 위키, 나무위키 사용
   - 최근에는 모두의 말뭉치(https://corpus.korean.go.kr/), AIHUB를 활용할 수 있음.
2. 토크나이징
   - 영어처럼 단순히 공백을 기준으로 단어의 경계를 나누면 성능이 별로 좋지 않다. 왜냐하면 **한국어 표기법에서는 띄어쓰기 단위가 단어의 경계가 반드시 일치하지는 않기 때문이다.** 게다가 유저들이 띄어쓰기를 잘 지키지도 않는다는 문제도 있다.
   - 한국어 같은 이단아들을 처리하기 위해 고빈도의 subword pair에서 각 subword를 하나의 토큰으로 처리하는 wordpiece, sentencepiece가 제안되었다. 그러나 이들

도 간혹 조사를 잘 분리해내지 못한다는 한계가 있다.

  - 예: `철수가` 라는 조합이 코퍼스에 많다면, wordpiece, sentencepiece는 `철수가` 를 한 단어로 취급함. 그러나 `철수` 는 명사, `가` 는 조사로, 한 단어로 취급해서는 안 됨.

- 그래서 최근에는 **형태소 분석기(mecab)로 초벌 tokenizing을 한 뒤, wordpiece나 sentencepiece로 2차 tokenizing**을 하는 더 섬세한 tokenizing 방식이 제안됨. (https://arxiv.org/abs/2010.02534)
  - 이렇게 하면 형태소 분석기를 통해 조사, 어미 등의 문법형태소가 잘 분리되므로, 위에서 언급한 문제가 발생하지 않음.
  - 또한, 토큰을 자르는 단위가 더 세밀할수록 wordpiece, sentencepiece 학습 시 포함되는 vocabulary(vocab.txt)에 유의미한 형태소들을 집어넣을 수 있음.
    - 예를 들어, 조사를 잘 쪼개지 못한다면 `철수가` , `철수는` , `철수를` 등이 하나의 vocab으로 포함되어버려서, 정작 vocab에 포함되어야 할 다른 단어들이 들어갈 자리가 없어져버림.
  - mecab으로 한 뒤 왜 또 sentencepiece나 wordpiece를 하나?
    - mecab만 하면 신조어, 희귀 단어 등이 OOV(Out of vocabulary)가 됨. → OOV가 많아지면 성능 저하. 이를 예방하기 위해 OOV에 강한 sentencepiece, wordpiece를 추가로 적용. (Park et al. 2020)
- `뱃사람` 의 `ㅅ` , `춥다` 의 `ㅂ` 등과 같은 분석을 위해 자모 단위로까지 쪼개기도 함. (KR-BERT)

---

3. 한국어 downstream task

- **kornli**(https://github.com/kakaobrain/KorNLUDatasets)
  - Natual Language Inference
  - 구성: premise, hypothesis의 쌍으로 구성됨. 이들 쌍의 관계가 entailment인지, contradiction인지, neutral인지를 분류하는 태스크임.
  - train: multinli train set + snli_1.0 train set = 942,854 line, dev: 2490 line, test: 5010 line
- **Korsts** (https://github.com/kakaobrain/KorNLUDatasets)

    - Semectic Textual similarity
    - 구성: 두 문장 쌍, label(의미적 유사도 0.0~5.0)
    - train: 5749, dev: 1500, test: 1379
- **NSMC**(https://github.com/e9t/nsmc)
    - naver movie review dataset
    - 영화 평점이 1(긍정), 0(부정)으로 레이블됨.
    - train: 150000, test: 50000
- **PAWS-X** (https://github.com/google-research-datasets/paws/tree/master/pawsx) PAWS-X: A Cross-lingual Adversarial Dataset for Paraphrase Identification
    - 한국어를 포함하여 6개의 언어의 paraphrase identification dataset.
    - label: 두 문장이 서로 동일한 의미(1), 서로 다른 의미(0)
    - (한국어) train: 49401, dev: 2000 , test: 2000
        - 공식 repo에서 소개하는 데이터 양: 49,401 / 1965 / 1972 (데이터셋을 보면 'NS'라고 적힌 행이 몇 개 있음. NS 행 제거하면 이 숫자 나옴.)
- **KorQuad(**https://korquad.github.io/)
    - 한국어 질의 응답
    - training: 60407, dev: 5774
- kornli, korsts, nsmc, paws-x, korquad: kortok github에 공개된 코드에 BERT 모델을 넣어서 구동 가능

- **KLUE**(https://arxiv.org/pdf/2105.09680.pdf)
    - github에 공개된 코드에 BERT 모델을 넣어서 구동 가능
    - Dependency parsing(DP)
        - 단어 간 의존 관계 찾기 (head, dependents)
    - RE (Relation Extraction)
        - 텍스트 내의 entity 간의 의미적 관계, subject entity와 object entity 간의 관계
    - yant(younhap news agency dataset for topic classification
        - 뉴스 기사 제목을 통해 토픽을 예측함

- STS (Semantic Textual Similarity)
  - korsts와 목적 동일
- NLI (Natural Language Inference)
  - kornli와 목적 동일
- WoS (Wizard of Seoul)
  - Multi domain dialogue state tracking
  - 멀티턴 대화에서의 Dialogue states 예측
- NER(Named Entity Recognition
  - 개체명 인식
- Machine Reading Comprehension (MRC)
  - 질의응답