ByT5: Towards a token-free future with pre-trained byte-to-byte models

TACL 2022 | Tag: Tokenizer-free, mt5

0. Summary

- Byte 시퀀스를 입력으로 받아(Token이 없음) NLP 파이프라인을 단순하게 하는 mT5의 변형 version

1. Introduction & Background

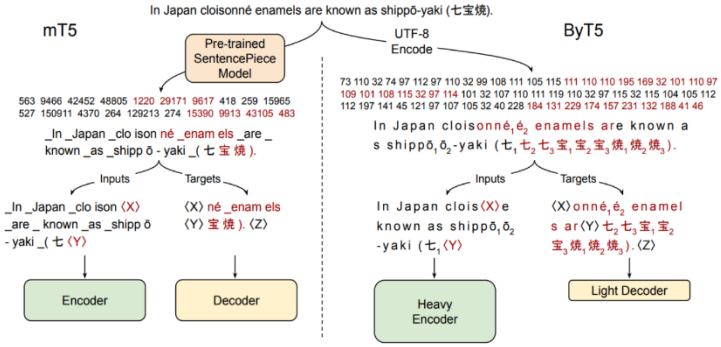
> 기존 연구

- 기존 pretrained 된 language model은 Subword tokenizer를 사용했음
- **Subword tokenizer의 문제점**: 모델마다 다른 tokenizer를 사용함, 언어별 특성 고려 필요, data에 noise나 오타 있을 수 있음

> Token-Free model

- Byte sequence를 직접 모델에 공급하여 처리
- **장점**: 학습된 어휘에 의존하지 않음 (모든 언어 처리 가능), noise에 더 강력
- **단점**: byte sequence는 token sequence보다 김, 모델 계산 비용은 sequence 길이에 따라 확장됨

2. ByT5 Design



> mT5와 달라진 점

- **Text를 utf-8 encoding**하여 Byte 단위로 모델에 공급
- mT5는 corrupted span length가 평균적으로 3개이지만, **ByT5는 평균 20개**
- encoder의 깊이는 decoder의 3배 (**Heavy Encoder**)

> Data efficiency (압축률)

- 언어별 **UTF-8 byte 길이 / tokenized 길이 평균**: 4.1
- Fixed input sequence length를 고려할 때, 모델은 pretraining동안 4배 적은 text에 노출되는 것

3. Result

> English Classification Tasks (GLUE, SuperGLUE)

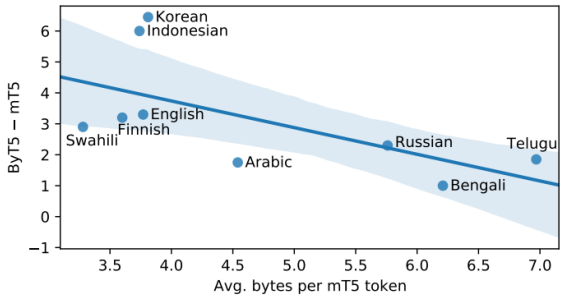
- Small, Base 모델에서는 mT5<ByT5
- Large, XL, XXL에서는 mT5>ByT5이지만 1%p 정도 차이
- **작은 모델에서 성능 향상 요인:** dense parameter의 증가 - mT5는 특정 토큰이 있을 때에만 예시에 접근하지만 ByT5는 모든 예시에서 활성화된 "dense" parameter를 사용, 더 효율적인 parameter를 사용하는 것으로 예상됨.

> English Generation Task

- XSum, TweetQA, DROP 모두 ByT5의 성능이 더 좋음

> Cross-lingual Benchmark

- 여러 언어에 대한 표현을 학습하는 Task
- 2 classification, 3 extractive QA, 1 structured prediction task (NER)로 구성됨



- 단어의 **압축률이 높을수록 ByT5의 성능이 낮은 추세**

> Word-Level task

- *Transliteration*(음역, 라틴문자로 외국어 음을 표현하는 것, ex: 가→ga), *Grapheme-to-Phoneme*(단어→음소), *Morphological inflection*(형태학적 굴절, ex: eat+PAST → ate)
- 3가지 모두 큰 폭으로 ByT5의 성능이 더 높음
- 이러한 작업은 NLP 모델 평가 시 종종 간과됨을 강조

> Synthetic noise

- TweetQA와 같은 messy한 text에도 잘 동작했음
- 더 심한 noise를 추가했을 때 mT5에 비하여 성능 하락의 폭 적음

> Speed Comparisons

- 모든 모델(small, base…)에서 byT5는 mT5에 비하여 최대 1.2배의 연산 필요
- Pretraining time 33% 더 소요, inference time 최악의 경우 10배 더 소요

📢 ***끝!!***

🐢질문 있으시면 정민지에게 DM 주세요!!