

접수번호

AI0288

「통계데이터 인공지능 활용대회」 2차 심사 설명자료

1) 구동 환경 및 정보

1-1) 구현에 사용한 프로그래밍 언어: Python

1-2) GPU, CPU 등 하드웨어 정보

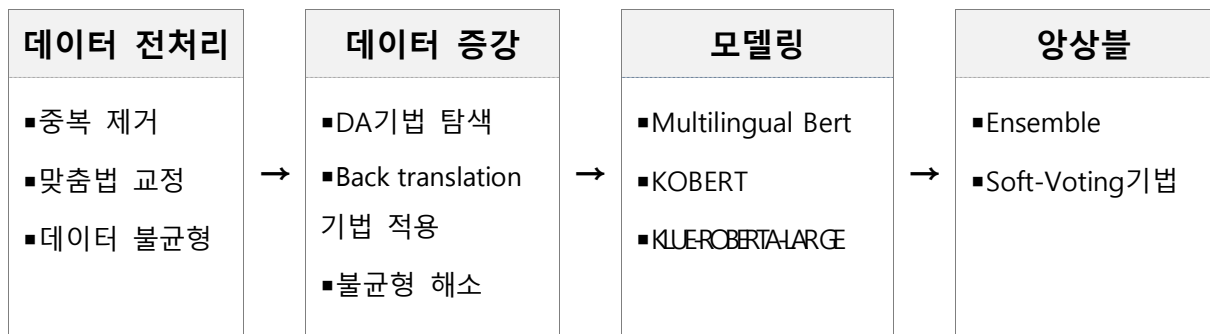
- 클라우드 : Google Colab Pro 사용

1-3) 학습 및 테스트에 소요되는 시간

- KOBERT: 6시간 35분
- KLUE-ROBERTA-LARGE(v1~v4): 7시간 36분 ~ 9시간 39분

1-4) 추가 데이터셋 사용: [KSIC 한국표준산업분류\(10차\) 기준표](#)

2) 알고리즘 개요



2-1) 데이터 전처리 : EDA & Text Preprocessing

- (중복 제거) 제공된 학습 데이터 100만 건에서 약 36만 건이 중복데이터로 확인되어 제거하였습니다.
- (맞춤법 교정) 학습 데이터에 존재하는 상당히 많은 맞춤법, 띄어쓰기 오류들을 아래와 같이 처리하였습니다.
 - re library를 사용해 특수문자를 제거
 - py-hanspell library를 사용해 맞춤법과 띄어쓰기를 교정
- (데이터 불균형) 학습 데이터에서 예측 소분류 값(Label)의 분포를 확인해본 결과, 클래스 불균형이 심함을 파악하였습니다.

Label Name	Train 건 수	Train Ratio(%)
561(음식점업)	62051	9.73%
961(미용, 욕탕 서비스업)	22877	3.5%
(중략)		
51(석탄 광업)	2	0.0003%
61(철 광업)	2	0.0003%

2-2) 데이터 증강 : Data Augmentation

- (데이터 증강) 아래의 DA^{Data Augmentation}기법을 사용했을 때, 생성된 문장들이 기존 Label의 성질을 잘 따르고, 모델 성능이 향상된다는 논문*을 참고하였습니다.
 - RS(Random Swap): 문장내 임의의 두 단어의 위치를 교체
 - RI(Random Insertion): 문장내 임의의 단어를 삽입
 - RD(Random Deletion): 문장내 임의의 단어 삭제
 - SR(Synonym Replacement): 문장내 특정 단어를 유의어로 교체

*참고 논문: Jason Wei, Kai Zou(2019), [*EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks, Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing , pages 6382-6388*](#)

- (Oversampling) 4가지 기법 중 RI와 SR의 경우 추가된 단어들로 인해 noise가 생기고 부정확하게 동의어 대체가 이루어지다 보니 오히려 성능이 저하되는 현상이 나타나서, RD와 RS 두가지 방법을 사용하여 약 10만건의 추가 데이터를 생성하고 클래스 불균형을 완화하였습니다.
- 추가적으로 Back translation 기법을 사용해 데이터를 증강하였습니다. 해당과정에서 google translator나 kakao에서 발표한 Pororo를 사용하려고 했으나, 일일 용량제한이 있거나 실행시간이 지나치게 오래걸리거나 번역기의 성능이 papago를 따라가지 못한다고 판단해 네이버 Papago를 크롤링해 사용하였습니다. 또한 Papago를 사용한 이유중 하나는 해당 데이터는 '...'과 같은 특수 문자들이 사용되는 일종의 규칙이 존재한 것을 발견했는데, 이러한 문장 스타일에 대한 규칙을 보존하면서 번역하는 모습을 볼 수 있었습니다. 최종적으로 한글을 영어로 번역한 뒤, 다시 한글로 재번역하여 데이터를 증강하였습니다.

2-3) 모델링 : Modeling

- 데이터 내에 다양한 형태의 텍스트들이 존재하여 한국어 특화 PLM^{Pre-trained Model}, multilingual PLM 등 다양한 모델들을 활용하였습니다.
- Huggingface*에서 제공하는 multilingual bert모델**을 사용해보았으나, score가 낮게 나와 제외하였습니다.

*자료 출처: <https://huggingface.co/bert-base-multilingual-case>

**참고논문: Jacob Devlin et al(2019). [BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of NAACL.](#)

- SKT brain에서 제공하는 Kobert모델*은 bert모델에서 한국어 위키 500만개의 문장과 5400만개의 단어를 학습시킨 한국어 특화 모델입니다. 해당 모델은 multilingual bert보다 월등히 높은 score를 보여주었고, 해당 모델을 baseline으로 설정하고 tokenizer 및 hyper-parameter tuning을 진행하였습니다.

*자료 출처: <https://github.com/SKTBrain/KoBERT>

- 해당 방법으로 어느정도 만족할 만한 score가 나왔지만 더 이상 추가적인 향상은 어려웠습니다.
- Kobert 모델은 8002개의 vocab size를 갖고 있는데, vocab size가 작다 보니 다수의 유의한 단어들을 catch하지 못한다는 점과 한국어 위키를 기반으로 학습했다 보니 학습 데이터 text와는 그 내용이 많이 다르다는 단점이 있었습니다.
- 추가적인 향상을 위해서 Klue-roberta-large모델*을 활용하였습니다. Klue**는 최초의 한국어 자연어 이해 벤치마크로 이에 대한 논문과 github을 참고하였습니다. Klue-roberta-large는 더 많은 한국어 문장 방식 데이터를 기반으로 학습하였으며 vocab size도 32000개로 충분했습니다. Klue-roberta-large의 weight는 유지하고자 일부 Layer는 얼리고 미세 조정하는 방식으로 4개의 모델(v1~v4)을 만들었습니다.

*자료 출처: <https://github.com/KLUE-benchmark/KLUE>

**참고논문: 박성준 외 31명(2021), [「KLUE: Korean Language Understanding Evaluation\(2021\)」](#)

2-4) 앙상블 : Ensemble

- 보다 정확한 예측을 도출하기 위해서 Kobert와 Klue-roberta-large(v1~v4)를 결합하였습니다.
- 결합방식으로 Voting방식(Hard, Soft)을 여러 가지로 시도하였는데, 각 모델들의 logit 값에 가중치를 부여해 soft voting을 한 후, 클래스 별 확률이 가장 높은 label을 최종 label로 선정하는 방식이 score가 가장 높았습니다