삼성카드 공모전

‘고객 피드백 분류 모델 개발’

HJJ

김홍범 이정인 차지헌

Preprocessing

Whole Process

Text Cleaning

기호 제거

- 영어/한글/문장부호(?!.,)를 제외한 기호 제거

- 이후 복문 처리 위해 문장부호는 제거 X

Spacing & Spell Check

Soynlp의 Soyspacing 사용[1]

- 한국어 띄어쓰기 문제를 해결하기 위한 **휴리스틱 알고리즘**을 제공

- Conditional Random Field와 비교하여 가벼운 모델 사이즈와 빠른 학습이 가능

: 자체적으로 데이터에 맞게 학습하여 띄어쓰기 수행

Py-Hanspell 사용

- 코드 참고하여 자체 맞춤법 검사함수 구현

Part of speech & Stopwords

KoNLPy (Komoran,Kkma)[2], Soysnlp 사용[1]

데이터에 가장 알맞게 분석하는 분석기를 사용하도록 함

Tagging 형태소 분석, 품사 확인
- Komoran / Kkma를 단독으로 사용
- Soynlp를 진행한 결과에 Komoran, Kkma를 사용

Extract 품사 추출
- 명사, 동사, 부사, 형용사 등 의미 있는 품사만 추출해 사용

Stemming 원형 복원
- 동사의 원형을 복원

Stopwords 불용어 제거
- 불용어로 쓰일 만한 명사 및 부사를 설정하고 입력받은 문장의 단어와 비교하여 제거

Complex sentence processing

KSS 사용[3]

- 각 문장의 최종 분류 선정은 기존의 대분류에 해당하는 (칭찬), (불만), (중립)을 사용

- **Kss** 적용하여 복문 구분 후, 전처리
- 만약 구단위가 2개 이상일 시 복문을 나누어 사용

예) 상담원은 친절하지만 알맹이가 없고 잘 설명해주질 못하시네요

→ 상담원은 친절하지만
→ 알맹이가 없고
→ 잘 설명해주질 못하시네요

[1] Lovit Github: https://github.com/lovit/soynlp
[3] Likejazz Github: https://github.com/likejazz/korean-sentence-splitter

[2] Park, Eunjeong L., and Sungzoon Cho. "KoNLPy: Korean natural language processing in Python." *Annual Conference on Human and Language Technology*. Human and Language Technology, 2014.

Embedding

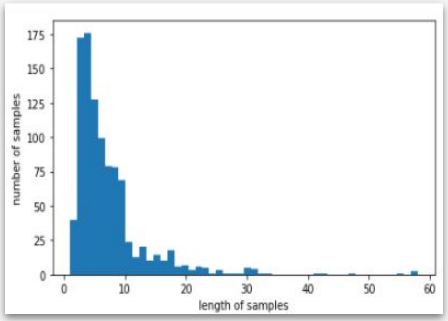
Whole Process

전처리 데이터 확인 (시각화)

시각화 통하여 전처리를 거친 결과 corpus 확인

- Vocab Frequency를 wordcloud로 시각화 (불용어 추가 등록에 이용)
- Vocab 개수, 문장 구성 Vocab 개수 확인 (Embedding, Modeling에 이용)

```
단어 총 개수(중복 허용) :  7249
단어 총 개수(중복 비허용) :  1256
문장 최대 길이 : 245
문장 평균 길이 : 25.161
문장 단어 최대 개수 : 58
문장 단어 평균 개수 : 7.249
```



Embedding

Word2Vec , GloVe, FastText 중 가장 높은 성능을 보이는 모델 선택

(참고논문 [1][2])

-이후 모델링 과정에서 이외의 임베딩 방법들 또한 사용해보며 가장 성능이 좋은 임베딩을 선택할 예정

→ **Word2Vec :**
뉴럴 네트워크 기반, 문서 데이터 셋이 벡터 공간에 높은 수준의 의미 벡터를 갖도록 효율적으로 Word 벡터 값을 추정할 수 있는 기계학습 모델
선택 이유) 효율성 측면에서 선택, 가장 유명한 모델

→ **GloVe :**
카운트 기반과 예측기반을 모두 사용하는 방법론적 모델
선택 이유) 사용자 지정 윈도우 사이즈 내에서만 학습&분석이 이루어지던 기존 Word2Vec의 단점을 보완

→ **FastText :**
형태 및 동사 정보를 학습하는 방식을 사용한 모델
선택 이유) 한국어처럼 다양한 접사가 존재하는 언어에서 높은 성능 보임

[1] Lee, Sanga, et al. "Performance analysis of Various Embedding Models Based on Hyper Parameters." *Annual Conference on Human and Language Technology*. Human and Language Technology, 2018.

[2] 이다빈, and 최성필. "대용량 텍스트 자원을 활용한 한국어 형태소 임베딩의 모델별 성능 심층 비교 분석." *한국정보과학회 학술발표논문집* (2018): 613-615.

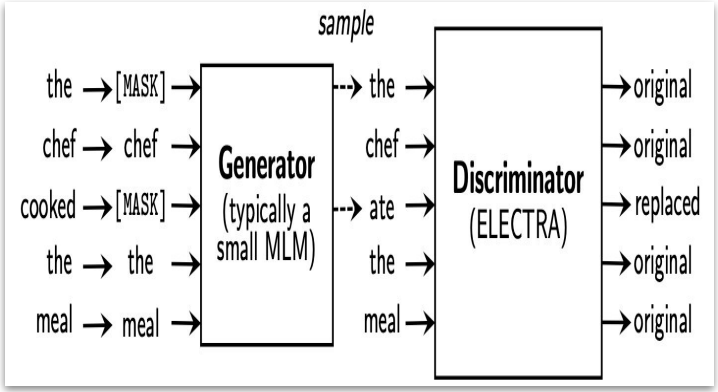
Modeling

Whole Process

Modeling (단어 / 문장 기반 모델)

CNN, Bi-LSTM, KoElectra, KoBERT 중 가장 높은 성능을 보이는 모델 선택

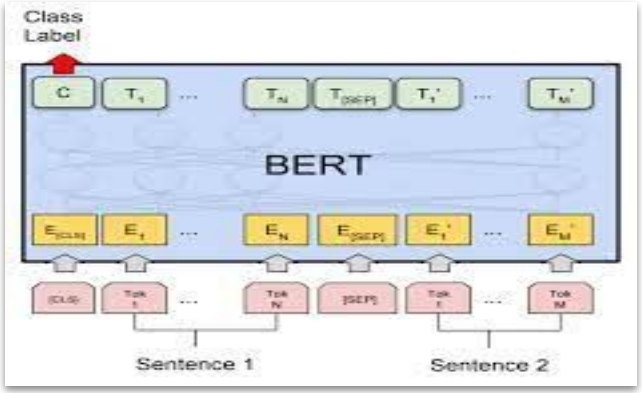
KoElectra[1]



- **Generator에서 나온 token을 보고 Discriminator에서 real, fake를 판별**하는 방법으로 학습함
- **모든 Input에 대해 학습** 가능 / BERT에 비해 성능 우수함
- 감성 분석에 우수한 성능을 보임

[1]Monologg Github : https://github.com/monologg/KoELECTRA

KoBERT[2]



- 문장 전체가 주어진 후 **빈칸[mask]**를 예측하는 방식으로 학습함 **프리트레인, 파인 튜닝**을 통해 성능 증대 가능함
- 감성 분석에 우수한 성능을 보임

[2] SKTBrain Github : https://github.com/SKTBrain/KoBERT

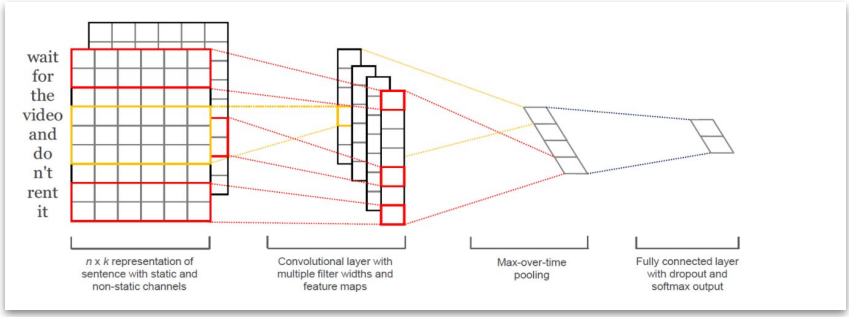
Modeling

Whole Process

Modeling (단어 / 문장 기반 모델)

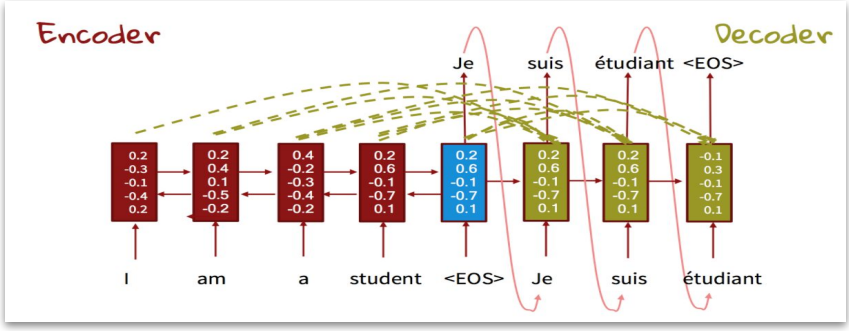
CNN, Bi-LSTM, KoElectra, KoBERT 중 가장 높은 성능을 보이는 모델 선택

CNN[1]



- **순차적인 데이터**를 보존한다는 점에서 자연어 처리에 강점을 보임
- **라벨링된 데이터**가 있을 시 우수한 성능을 보임

Bi-LSTM[1]



- 출력값에 대한 **손실을 최소화**하는 과정에서 모든 파라미터를 동시에 학습하는 **종단간 학습 가능함**
- **단어와 구(Phrase)간 유사성**을 입력벡터에 내재화하여 성능 개선 가능함

Post-processing : 칭찬/불만/중립 으로 분류된 결과 데이터에 대하여, **중·소분류 라벨링** 진행

→ TF-IDF 등의 단어 빈도 기반 중요도를 각 중·소분류에 대해 계산하여 라벨링

[1]김나랑. "CNN 과 Bidirectional LSTM 을 활용한 부산시 민원 자동 분류 연구." *전산회계연구* 17.2 (2019): 81-98.