CP2
누비랩(Nuvi-Lab) 식단추천 프로젝트

AI10기 임규현
6/3/21

목차

1. 프로젝트 개요

2. 프로젝트 수행 절차 및 방법

3. 프로젝트 수행 결과

4. 프로젝트 회고 및 자체 평가 의견

1. 프로젝트 개요

- **영양사**분들의 식단추천에 도움을 주기 위한 프로젝트
  - 식단을 짤 때 고려하는 여러가지 요소들이 존재 (메뉴 대분류, 고기 종류, 색감 등)
  - AI 라벨링을 통해 식단 추천에 대한 도움 및 인사이트 제공이 목표
  - 예) 메뉴에 고기가 골고루 들어가는 걸 돕기 위해 고기 종류별로 AI 라벨링 진행
- 급식 데이터 음식 메뉴 분류 문제로 접근
  - '고기' 카테고리를 기준으로 분류 모델 개발
- 기대효과
  - 영양사분들의 식단추천 과정 개선 / 최적화

2. 프로젝트 수행 절차 및 방법 - 1

- 총 진행기간
  a. 5/9(월)~6월/3(금) (총 4주)
- Week 1 (5/9 ~ 5/13)
  a. 사전기획
    i. 프로젝트 기획 및 주제 선정
- Week 2 (5/16 ~ 5/20)
  a. EDA (전처리) 진행 및 모델 선정
- Week 3 (5/23 ~ 5/27)
  a. 모델 선정 및 개발
- Week 4 (5/30 ~ 6/3)
  a. 모델 개발 완료
  b. 데이터 시각화 진행
  c. 최종보고 및 PPT 자료 준비

3. 프로젝트 수행 결과 - 1

- 탐색적 분석 및 전처리
  a. 데이터 소개
     i. NEIS 푸드 데이터 (급식)
        1. https://open.neis.go.kr/portal/data/service/selectServicePage.do?page=1&rows=10&sortColumn=&sortDirection=&infId=OPEN17320190722180924242823&infSeq=2
     ii. 메뉴 66만개
  b. EDA
     i. NEIS 푸드 데이터를 살펴 본 후 각 고기 카테고리에 대한 워드 라이브러리 구성
     ii. Tokenization: KoBERT
     iii. 중복메뉴 제거 (66만개 -> 5.5만개로 축소)
     iv. Word Library: 각 고기 종류에 대한 단어 라이브러리 구성
        1. 예). 돼지고기: 제육, 돈육, 돈까스 등

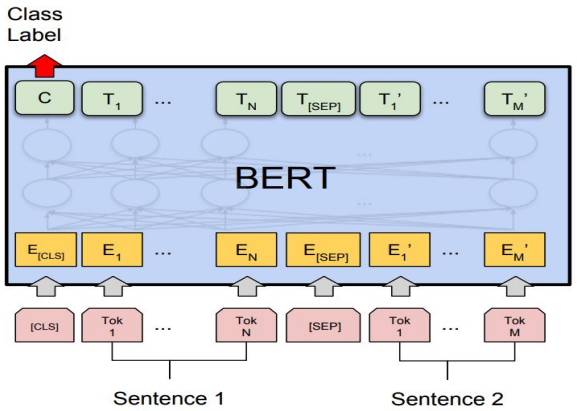
```
# 고기별 단어 리스트 정리하기
beef = ['소고기','쇠고기','함박','불고기']
# 함박스테이크 제외? -> 함박스테이크, 함박스테이크소스 등..
pork = ['제육','돼지','돈육','돈갈비','목살','돈까스','두루치기','돈코츠','돈사태','고추장불고기','스팸','삼겹살']
chicken  = ['탄두리','치킨','닭','장각']
duck = ['오리','오리훈제'] #오리엔탈은 제외?
fish = ['고등어','연어','생선','동태','오징어','새우','게','조개','굴','전복','홍합','참치','어묵','해물','가자미','멸치','주꾸미','쭈꾸미'] #굴소스 제외?
```

4. 프로젝트 수행 결과 - 2

- Modeling - 시도는 하였으나 적용하지 못했던 모델들
  a. Autokeras
     i. **고려했던 이유:** AutoML 프로그램. 편의성 UP
     ii. **장점:** 내장된 TextClassfier. 별도의 Tokenization (X)
        1. 스스로 신경망 검색 가능 + 텍스트든 이미지든 다양한 형식 처리 가능
     iii. **단점:** 특수한 데이터나 데이터가 적을 때 성능이 떨어짐
        1. 한국어 텍스트를 분류 불가
  b. Naive Bayes Classifier
     i. **고려했던 이유**: 텍스트 분류를 위해 전통적으로 사용되는 분류기
     ii. KoNLPy를 통해 토큰화 - 한국어 말뭉치를 제공하는 패키지
     iii. **결과**: 토큰화까지는 문제가 없었으나 모델에 train / test 데이터를 넣는 과정에서 에러 발생

4. 프로젝트 수행 결과 - 3

- Modeling
  a. KoBERT (모델 개요)
     i. **장점**: 한국어에 특화된 모델. 토큰화 + 다중분류 동시에 진행 가능
     ii. SKT 에서 공개한 모델
     iii. BERT 구조를 따름 + 한국어 버전으로 만듦
     iv. 한국어 위키에서 5백만 개의 문장과 5400만개의 단어를 학습



SKTBrain/**KoBERT**

Korean BERT pre-trained cased (KoBERT)

5 Contributors · 3 Issues · 627 Stars · 159 Forks

4. 프로젝트 수행 결과 - 4

- Modeling - KoBERT (계속)
  - 1st Case
    - 고기 종류 1~5까지 라벨링 된 데이터로 진행 (라벨 0 제외)
  - 라벨 0은 고기가 아님
  - 총 데이터 수 ~ 1만개
  - 결과물: 메뉴 이름을 넣으면 메뉴 이름 입력 시 고기 종류 반환
  - 문제: 0으로 분류돼야 하는 '고기가 아닌 메뉴'들도 고기로 분류
  - Epoch 5 기준 train acc. = 0.999, test acc. = 0.998
  - Predict() 함수 선언
    - 모델에 메뉴 입력 -> 라벨링 결과 출력
    - 0 (고기 없음) 으로 라벨돼야 되는 메뉴들도 1~5 로 라벨링이 되는 오류가 있었음

4. 프로젝트 수행 결과 - 5

- Modeling KoBERT (계속)
  - 2nd case
    - 라벨링이 0~5 까지 된 데이터 (중복제거 O, 증폭 X, 5.5만개 중 1만개 샘플링)
  - Epoch 4 기준 train acc. = 0.999, test acc. = 0.998
  - Vs. 누비푸드 데이터 (595/1855 가 고기에 대해 라벨링 완료)
    - .Predict 함수 () 을 통해 결과 확인
      - 워드 라이브러리에 있는 단어들은 쉽게 라벨링이 되었음
      - 라이브러리에 없는 단어들 → 0 (분류가 안 됨)
      - '정답'인 누비푸드 데이터에 비해서는 다소 아쉬운 점

4. 프로젝트 수행 결과 - 6

- Modeling KoBERT (계속)
  - Predict 함수를 통한 예측 vs. Nuvi_Foods 데이터와 비교 (정답)
    - 돼지 - 50% 정답률
    - 소 - 80%
    - 닭 - 100%
    - 오리 - 80%
    - 물고기 - 60%

```
메뉴 이름을 입력하세요 : 해물찜
> 해산물이 들어간 메뉴입니다.

메뉴 이름을 입력하세요 : 소고기
> 소고기가 들어간 메뉴입니다.

메뉴 이름을 입력하세요 : 바지락수제비
>아무것도 안 들어간 메뉴입니다.

메뉴 이름을 입력하세요 : 통안심치킨꼬치
> 닭고기가 들어간 메뉴입니다.

메뉴 이름을 입력하세요 : 소고기미역국
> 소고기가 들어간 메뉴입니다.

메뉴 이름을 입력하세요 : 햄듬뿍부대찌개
>아무것도 안 들어간 메뉴입니다.

메뉴 이름을 입력하세요 : 고구마치즈돈까스
> 돼지고기가 들어간 메뉴입니다.

메뉴 이름을 입력하세요 : 보리밥
>아무것도 안 들어간 메뉴입니다.
```

```
f1_score(y_true, y_pred, average=None)
```

```
array([0.33333333, 0.33333333, 0.75      , 0.75      , 0.88888889,
       1.        , 1.        , 1.        , 1.        , 1.        ])
```

4. 프로젝트 수행 결과 - 7

- Word Cloud (5.5 만개의 나이스 푸드 데이터 사용)
  - 가장 눈에 띄는 단어들은
    - **오리, 불고기, 볶음, 소스, 훈제, 치킨** 등
    - Most_common 을 통해서도 확인할 수 있음


```
counter_food .most_common(10)
```

```
[('오리', 1147),
 ('볶음', 979),
 ('불고기', 972),
 ('소스', 911),
 ('구이', 731),
 ('훈제', 666),
 ('치킨', 617),
 ('닭', 575),
 ('샐러드', 520),
 ('치즈', 447)]
```

5. 자체 평가 의견

- 누비푸드 데이터의 라벨링에 비해서 다소 아쉬운 퍼포먼스의 원인?
  - KoBERT 모델 자체의 특징 (글씨기반 학습)
  - 음식 단어들에 대한 특징 학습이 잘 이루어지지 않음
  - 고로 워드 라이브러리 외의 단어들에 대한 분류가 어려움
- 워드라이브러리 업데이트를 통한 성능 개선 가능
  - 하지만 근본적인 해결책은 아님
  - KoBERT 과는 다르게 '음식' 단어 자체에 대해 학습이 더 잘 되는 모델이 있나 확인 필요
- 모델링과 디버깅에 시간을 뺏겨서 원하는 결과물을 도출하는 데 어려운 점이 있었음