ecoccon

DB 데이터 정규화

최 창 규

CONTENTS

01

주소 정규화

01. 주소 정규화

1) ADDR list

기존의 주소는 같은 서울특별시라도 서울시, 서울특별시, 서울 등 표현이 제 각각이어서 정규화가 되어있지 않아 이를 구분할 때 케이스가 복잡해지는 문제가 있었다.

그래서 처음에는 이렇게 표현의 정규화가 되어있지 않은 시, 도의 표현을 일치시키는 것을 시행했다.

01. 주소 정규화

1) ADDR list

```
addr_list['서울특별시']=('서울특별시', '서울시', '서울')
addr_list['경기도']=('경기도', '경기')
addr_list['강원도']=('강원도', '강원')
addr_list['충청북도']=('충청북도', '충북')
addr_list['충청남도']=('충청남도', '충남')
addr_list['경상북도']=('경상북도', '경북')
addr_list['경상남도']=('경상남도', '경남')
addr_list['전라북도']=('전라북도', '전북')
addr_list['전라남도']=('전라남도', '전남')
addr_list['부산광역시']=('부산', '부산광역시', '부산시')
addr_list['대구광역시']=('대구', '대구광역시', '대구시')
addr_list['인천광역시']=('인천', '인천광역시', '인천시')
addr_list['광주광역시']=('광주', '광주광역시', '광주시')
addr_list['대전광역시']=('대전', '대전광역시', '대전시')
addr_list['울산광역시']=('울산', '울산광역시', '울산시')
addr_list['세종특별자치시']=('세종', '세종시', '세종특별시', '세종특별자치시')
addr_list['제주특별자치도']=('제주', '제주도', '제주특별도', '제주특별자치도')


sub_addr_list['충청북도']=('제천시', '충주시', '단양군', '음성군', '진천군', '증평군', '괴산군', '청주시', '보은군', '옥천군', '영동군')
sub_addr_list['충청남도']=('논산시', '계룡시', '서산시', '태안군', '공주시', '부여군', '천안시', '홍성군', '예산군', '아산시', '서천시',
                          '당진시', '보령시', '청양군', '금산군')
sub_addr_list['경상북도']=('포항시', '경주시', '김천시', '안동시', '구미시', '영주시', '영천시', '상주시', '문경시', '경산시', '군위군',
                          '의성군', '청송군', '영양군', '영덕군', '청도군', '고령군', '성주군', '칠곡군', '예천군', '봉화군', '울진군', '울릉군')
sub_addr_list['경상남도']=('창원시', '김해시', '진주시', '양산시', '거제시', '통영시', '사천시', '밀양시', '함안군', '거창군', '창녕군', '고성군',
                          '하동군', '합천군', '남해군', '함양군', '산청군', '의령군')
sub_addr_list['전라북도']=('전주시', '익산시', '군산시', '정읍시', '김제시', '남원시', '완주군', '고창군', '부안군', '임실군', '순창군', '진안군',
                          '무주군', '장수군')
sub_addr_list['전라남도']=('목포시', '여수시', '순천시', '나주시', '광양시', '담양군', '곡성군', '구례군', '고흥군', '보성군', '화순군', '장흥군',
                          '강진군', '해남군', '영암군', '무안군', '함평군', '영광군', '장성군', '완도군', '진도군', '신안군')
```

01. 주소 정규화

1) ADDR list

각 시, 도 이름당 나올 수 있는 경우의 케이스들을 집합화 해서 표준표현을 지정했다.

충청도나, 경상도 등 남, 북이 구분되지 않는 도의 경우는 그 뒤에 따라오는 시나 군을 보고 정할 수 있다.

01. 주소 정규화

2) 띄어쓰기가 안되어있는 경우

간간히 주소에 띄어쓰기가 안 되어 있어서 시, 구, 동 등을 구분하기 힘든 케이스들이 있었다. 처음에는 다양한 경우마다 다르게 코딩하는 것을 생각했으나 너무나도 많은 경우의 수가 존재하는 것 같아서 포기했다.

그러던 중 꽤 품질이 좋은 띄어쓰기 모듈을 발견을 했는데 굉장히 쓸모 있었다.

```
In[2]: from pykospacing import spacing
  ...: print(spacing('경기도성남시분당구정자동불정로6'))
  ...:
C:\Program Files\JetBrains\PyCharm Community Edition 2018.1.4\helpers\pydev\_
  module = self._system_import(name, *args, **kwargs)
Using TensorFlow backend.
Backend Qt5Agg is interactive backend. Turning interactive mode on.
WARNING:tensorflow:From C:\ProgramData\Anaconda3\lib\site-packages\tensorflow
Instructions for updating:
`NHWC` for data_format is deprecated, use `NWC` instead
경기도 성남시 분당구 정자동 불정로6
```

하지만 띄어쓰기가 완벽한 것은 아니기 때문에 모든 경우에 다 쓰기보다는 띄어쓰기로 구분이 되어있지 않다고 판단이 되는 것들에게만 적용하는 것이 옳다.

출처 : https://github.com/haven-jeon/PyKoSpacing

01. 주소 정규화

3)크롤링

띄어쓰기까지 시행한 뒤에도 해당 주소가 3마디 이하이거나 오,탈자가 존재해서 인식이 불가능 하다면 웹사이트에 레코드의 위도, 경도 좌표를 이용해서 해당 위치의 주소를 받아와서 대체한다. (http://mygeoposition.com/)

이 경우는 웹 드라이버를 사용해서 속도가 느리기 때문에 최후에 사용해야 할 것 같다. 또한 레코드의 좌표가 정말로 유효한지에 대해서도 확신이 없기 때문에 정확한 값을 보장하기는 힘들다.

그 밖에 좌표도 존재하지 않는 레코드라면 해당 레코드의 주소는 전화번호를 이용해서 후후나 114등에서 크롤링해서 얻어오는 방법도 있다.

02

전화번호 정규화

02. 전화번호 정규화

전화번호의 경우 사이사이에 띄어쓰기나 -,), . 등 의미 없는 기호들이 들어가는 경우가 있는데 이것을 제거하는 것만으로도 개선효과가 좋을 것으로 기대된다.

그 밖의 전화번호의 유효성은 후후나 114에 직접 검색해서 판단하는 것이 바람직할 것으로 보인다.