목표 : meums\_company 의 CO\_NAME을 통해 브랜드를 추출하고 뽑은 브랜드를 통해 카테고리 수정

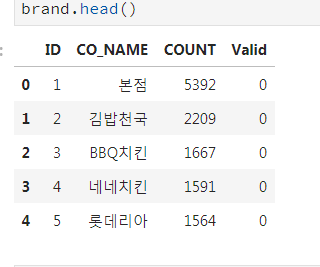
브랜드란? -> 지점을 가지고 있는 상점 (지점 개수 기준은 추후에 세우기로)

단계1 = 지점 리스트 만들기

1-1.

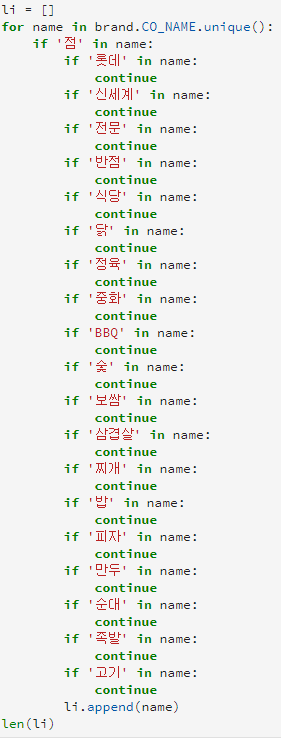
먼저 CO\_NAME을 띄어쓰기 단위로 나눈 뒤 각 단어의 등장 빈도를 셉니다.

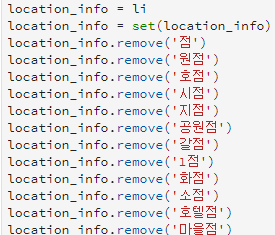
단어와 단어의 등장 횟수 pair를 brand라는 이름의 table로 eums-poi 에 저장합니다.



그 중 점으로 끝나는 모든 단어를 가져온 뒤 지역+점 으로 이루어진 단어를 따로 리스트화 합니다. (본점,수원점, 교대점, 홍대점 등등)

문제점(하단 그림참조)





점으로 끝나는 모든 단어 이기 때문에 백화점, 정육점, 전문점, 홍콩반점 등등 지점명이 아닌 점을 가져오게 됩니다.

sol) 직접 제거합니다.

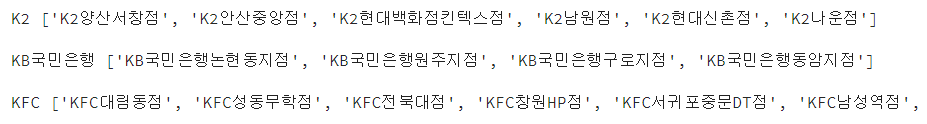
1-2

이 지역 지점 리스트들을 통해 co\_name\_r에서 상점이름은 상점명+지역점으로 구분하고

예) CU교대점 => CU 와 교대점

지점을 가진 상점명을 dictionary화 합니다.

예){CU : CU교대점, CU홍대점, CU수원점, , , , }



이렇게 만들어진 상점을 브랜드라고 가정합니다.

단계2 => 브랜드끼리 묶어서 진짜 브랜드를 만든다

현재의 문제점은 CU와 씨유로 묶이는 것이 존재한다는 것입니다.

예){CU : CU교대점, CU홍대점, CU수원점……} 과 {씨유:씨유교대점, 씨유고속버스터미널점,…..}

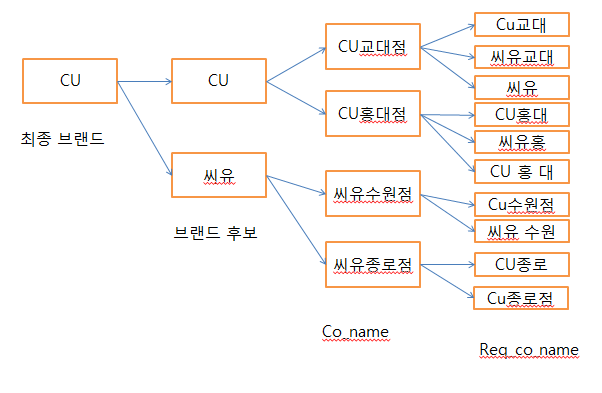
여기서 사람은 CU와 씨유가 같은 브랜드 임을 알지만 우리의 목적은 자동으로 브랜드 리스트를 생성하는 것 이기 때문에 컴퓨터가 CU와 씨유가 같은 브랜드라는 계산을 수행할 수 있게 해줘야 합니다.

이때 req\_mapp이라는 테이블을 참조해 비교 연산을 수행합니다.

Req\_mapp은 자주와 어플을 통해 들어온 문자 정보가 들어있는 table 입니다.

이 테이블에 req\_co\_name이라는 문자를 통해 들어온 상점명이 존재하고,

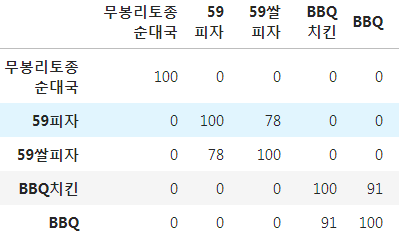
이를 한 상점에 대한 document로 여깁니다.



브랜드 후보 CU에 대한 document = {CU교대, 씨유교대, 씨유, CU홍대, 씨유홍….},

씨유에 대한 document = {Cu 수원점, 씨유 수원, CU종로, CU종로점….} 으로 구성됩니다.

CU document 와 씨유 document를 비교하여 유사도가 높으면 최종 브랜드 CU로 묶습니다.



Ex)유사도 비교를 통해 BBQ와 BBQ치킨의 유사도 91%, 59피자와 59쌀피자의 유사도 78%를 밝혀냄