**CNE-Join과 N-Join-Pair에 대한 설명 및 사용법**

1. **본 프로그램의 특징**
   1. 본 프로그램은 테이블 데이터셋을 대상으로 함
   2. CNE-Join은 조인 가능한 테이블을 탐색하기 위해 컬럼명과 컬럼값의 유사성을 동시에 고려함
   3. CNE-Join은 컬럼명 임베딩과 컬럼값에 대한 개체명 인식 및 포괄성 계산을 통해 컬럼간의 유사성을 알아냄
   4. CNE-Join은 최종적으로 조인 가능한 테이블쌍과 조인의 기준이 되는 컬럼쌍을 사용자에게 제공함
   5. N-Join-Pair는 CNE-Join의 결과를 활용해 3개 이상의 조인 가능한 테이블 조합을 생성함
   6. N-Join-Pair는 생성된 3개 이상의 조인 가능한 테이블 조합에 대해 조인이 잘 된 순으로 랭킹하여 사용자에게 제공함
2. **주요 기능**
   1. 다수의 테이블을 입력 받아 각 테이블에 대해 컬럼명 임베딩, 개체명 인식 및 포괄성 계산을 수행하여 조인 가능한 테이블쌍과 조인 기준이 되는 컬럼쌍을 찾음
   2. 조인 가능한 테이블쌍을 조합해 3개 이상의 조인 가능한 테이블 조합을 생성하고 이를 랭킹하여 사용자에게 제공함
3. **사용 방법**
   1. Anaconda, Jupyter Notebook, Python 설치 필요
   2. Anaconda 프롬프트를 실행하여 pip install -r requirements.txt 를 입력
   3. requirements.txt에 있는 모든 패키지를 설치한 후 Anaconda 프롬프트에 jupyter notebook을 입력하여 Jupyter Notebook 실행
   4. Jupyter Notebook이 실행되면 CNE-Join 및 비교기법 실험코드.ipynb 실행
   5. CNE-Join 및 비교기법 실험코드.ipynb 소스 파일은 크게 3개의 파트로 작성되어 있음
      1. <Part 1> : 필요 라이브러리 Import 및 개체명 인식기 구축 코드
      2. <Part 2> : CNE-Join을 실행하기 위해 필요한 함수 정의 및 테이블 데이터셋을 불러와서 전처리하고 CNE-Join을 실행하는 코드
      3. <Part 3> : CNE-Join과 성능을 비교할 비교 기법의 코드
   6. CNE-Join 및 비교기법 실험코드.ipynb <Part 2> 까지 코드를 실행하면 조인 가능한 테이블쌍 리스트를 얻을 수 있음
   7. 조인 가능한 테이블쌍 리스트를 Table Pairs Combinator.ipynb 소스 파일의 입력으로 Table Pairs Combinator.ipynb를 실행
   8. Table Pairs Combinator.ipynb를 실행 완료하면 조인 가능한 3개 이상의 테이블 조합이 만들어지고 이는 csv파일의 형태로 저장됨
   9. 저장된 csv파일을 활용해 N-Join-Pair 코드.ipynb 소스 파일을 실행
   10. N-Join-Pair 코드.ipynb 소스 파일은 크게 4개의 파트로 작성되어 있음
       1. <Part 1> : 필요한 라이브러리를 Import하는 코드
       2. <Part 2> : 조인 가능한 3개 이상의 테이블 조합 csv파일 불러오기 및 전처리 코드
       3. <Part 3> : N-Join-Pair를 실행하기 위해 필요한 함수 정의 및 N-Join-Pair를 실행하는 코드
       4. <Part 4> : 조인 테이블과 조인에 사용된 소스 테이블의 TF-IDF 값 계산 및 TF-IDF overlapping을 활용하여 N-Join-Pair의 성능평가를 진행하는 코드
   11. N-Join-Pair 코드.ipynb 파일을 실행 완료하면 랭킹이 매겨진 조인 가능한 3개 이상의 테이블 조합을 얻을 수 있음