Project 5. AWS와 Hadoop 시스템을 이용한 N-Gram 언어모델 생성

김지환 교수

Office: AS912

Tel: 715-8924

Email: kimjihwan@sogang.ac.kr

프로젝트 목표

- *N*-Gram 언어모델은 현재 단어로 부터 앞의 n-1개의 단어가 주어 졌을 때, 현재의 단어가 생성되는 확률을 계산해주는 모델이다.
- *N*-Gram 언어모델은 검색어 자동 완성, 자연어처리, 음성인식 등에 사용된다.
- 본 프로젝트 에서는 *N*-Gram 언어모델 생성에 필요한 N-gram Count를 생성한다.
  - 대용량 코퍼스가 제공되고, AWS상에서 Hadoop 시스템을 이용한 분산 처리 시스템을 통해 생성한다.
  - Python 과 AWS Elastic MapReduce (AWS EMR)을 사용한다.

프로젝트 요구사항 및 설명

- 프로젝트 목표 설정
  - 본 프로젝트에서는 대용량 코퍼스로부터 **Bigram(*N*-Gram에서 *N*=2)** 언어모델 생성에 필요한 **Bigram Count**를 생성한다.
    - **Python과 AWS Elastic MapReduce 시스템을 이용한 분산처리 시스템을 구축한다.**
- 합성
  - 입력
    - **Amazon에서 Pubic Data Set으로 공개한 Common Crawl Corpus의 2.2GB 텍스트**
      - http://aws.amazon.com/datasets/41740
      - 200TB 이상의 자료 중 2.2GB 별도 배포
    - 각 줄에 한 문장씩 저장되어있음
    - 단어는 **white spaces**로 구분

프로젝트 설명

- 합성
  - **Bigram Count 구현**
  - 출력
    - 각 줄에 단어 조합과 그 단어 조합이 나온 횟수를 **Tab**으로 구분하여 저장합니다.
    - 단어 조합은 알파벳 순으로 정렬한다.
- 제작
  - **Hadoop 및 AWS EMR 이용**
    - **Hadoop 및 AWS EMR**은 별도의 강의 자료를 통하여 설명

프로젝트 설명

- 시험
  - **AWS의 Elastic MapReduce를 이용하여 Bigram Count 생성**
    - **m3.xlarge 인스턴스 타입 3대로 구성한다.**
  - 실행을 위한 **Mapper.py, Reducer.py**는 **AWS S3** 안 자신의 버킷에 저장
  - 생성한 최종 출력은 **AWS S3** 안 자신의 버킷에 **part-***로 저장되어야 함

프로젝트 설명

- 평가
  - **part-*에서 빠진 단어 조합이 있는지, Count수가 같은지를 검사**
  - 수행 시 인스턴스 타입을 지키지 않거나, 인스턴스 개수를 초과하여 계산하였을 경우 감점
    - 추가적으로 **mapper, reducer**의 수행 시간 기준 성능을 평가할 수 있음

프로젝트 설명

- 환경구성
  - **AWS Elastic MapReduce**
- 제출물
  - **Mapper.py**
  - **Reducer.py**
  - **AWS EMR에서 수행완료한 캡쳐화면 (.png)**
    - 클러스터 생성시 클러스터명에 본인 학번 포함하여 작성
      - 생성시 Security 부분의 visible to all iam user 체크해제!!
  - **S3 링크 – 링크만 제출**

제출 방법

- **sp학번_proj5라는 이름의 디렉터리를 만들고, 이 디렉터리에 제출파일을 넣어서 디렉터리를 tar로 압축하여 한 파일로 만들어 메일로 보내시기 바랍니다.**
- **제출 파일은 sp{학번}_proj5.tar 입니다.**
- **제출 기한: 2018년 5월 28일 23:59**
- **제출 주소 : ied2017c3@gmail.com**

  메일제목 형식 : [SP숙제 #5] 학번 이름

주의 사항은 이전 프로젝트와 같습니다.