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

박운상 교수

Office: K 336

Tel: 705-8936

Email: unsangpark@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 대용량 코퍼스로부터 *N*-Gram 언어모델을 AWS 상에서 Hadoop 시스템을 이용한 분산처리를 통해 빠르게 생성한다.**
- **이를 위해 Python과 AWS에서는 Elastic MapReduce (AWS EMR)을 사용한다.**

프로젝트 요구사항 및 설명

- 프로젝트 목표 설정
  - 본 프로젝트에서는 대용량 코퍼스로부터 **Bigram(*N*-Gram에서 *N*=2)** 언어모델을 **Python**과 **AWS Elastic MapReduce** 시스템을 이용한 분산처리를 통해 빠르게 생성한다.
- 합성
  - 입력
    - **Amazon**에서 **Pubic Data Set**으로 공개한 **Common Crawl Corpus**의 **2.2GB** 텍스트
      - http://aws.amazon.com/datasets/41740
      - 현재 541TB로 늘었으므로, 별도로 배포할 예정
    - 각 줄에 한 문장씩 보관되어 있음
    - 단어는 **white spaces**로 구분된다.

프로젝트 설명

- 합성
  - N-Gram 언어모델 구현
  - 출력
    - 각 줄에 단어 조합과 그 단어 조합이 나온 횟수를 Tab으로 구분하여 저장합니다.
    - 단어 조합은 알파벳 순으로 정렬한다.
- 제작
  - Hadoop 및 AWS EMR을 이용
    - Hadoop 및 AWS EMR은 별도의 강의자료를 통하여 설명

프로젝트 설명

- 시험
  - **AWS의 Elastic MapReduce를 이용하여 Bigram 언어모델을 생성한다.**
    - **R3.xlarge 인스턴스 타입의 3대로 구성됩니다.**
  - 실행을 위한 **Mapper.py, Reducer.py**는 **AWS S3** 안 자신의 버킷에 저장
  - 생성한 최종 출력은 **AWS S3** 안 자신의 버킷에 **parted-***로 저장되어야 함

프로젝트 설명

- 평가
  - **Parted-*에서 빠진 단어 조합이 있는지, Count수가 같은지를 검사**
  - **Mapper와 Reducer의 성능평가를 위해 3번을 돌려 평균을 내어 차등 점수 부여**
    - 수행 시 인스턴스 타입을 지키지 않거나, 인스턴스 개수를 초과하여 계산하였을 경우 경고

프로젝트 설명

- 환경구성
  - **AWS Elastic MapReduce**
- 제출물
  - **Mapper.py**
  - **Reducer.py**
  - **AWS EMR에서 수행하는 캡쳐화면 (.png)**
  - **S3 링크**

제출 방법

- **sp학번_proj5라는 이름의 디렉터리를 만들고, 이 디렉터리에 제출파일, Document, readme 파일을 넣어서 디렉터리를 tar로 압축하여 한 파일로 만들어 메일로 보내시기 바랍니다.**
- **제출 파일은 sp<학번>_proj5.tar 입니다.**
- **제출 주소 : sp2016proj@gmail.com**

  메일제목 형식 : [SP숙제 #5] 학번 이름

주의 사항은 이전 프로젝트와 같습니다.