빅데이터 최신기술

문장 생성 확률 계산

소프트웨어학부
20163162
차윤성

1. 과제 수행 내용 설명

- 프로그램 개요

  프로그램 실행 명령 옵션으로 대용량 말뭉치 파일을 입력하면, 대용량 말뭉치에서의 bigram 음절을 count 한 후 확률을 계산하려는 문장에서 시작음절과 끝 음절은 고려하지 않고, 한 음절 이후 등장하는 다음 음절에 대한 확률값을 각각 계산해 곱한다.

$$s1,s2,s3,...,sn) = P(s2|s1) \times P(s3|s2) \times ... \times P(sn|sn-1)$$

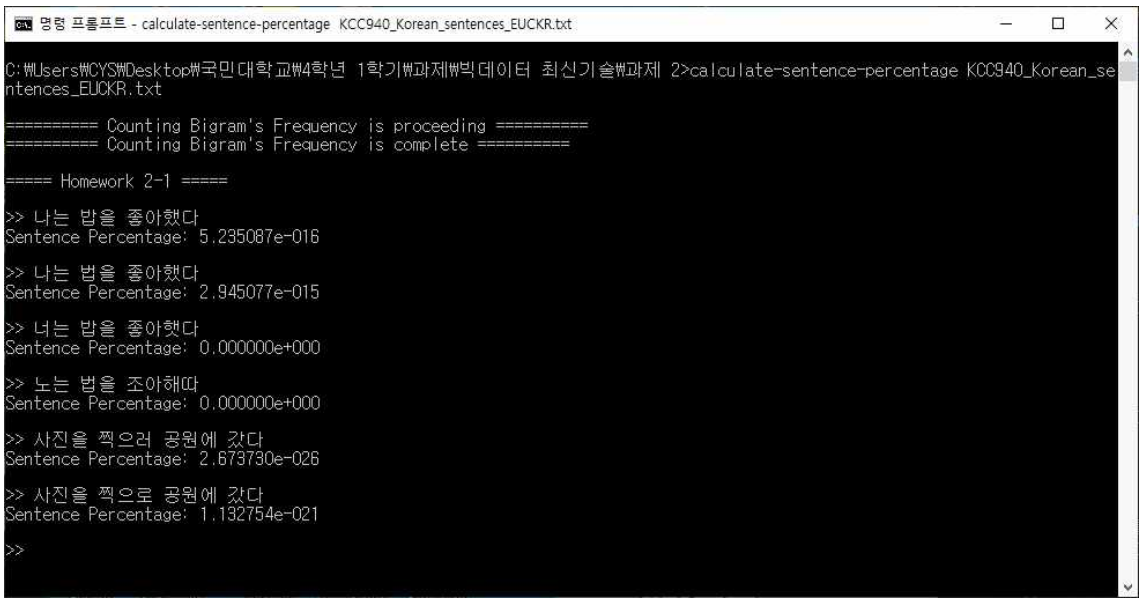
  키보드로 입력받은 문장들의 확률을 모두 구했으면 “q"를 입력해 키보드 입력을 중지하면 프로그램이 생성한 문장들의 확률을 각각 구하고, 각 케이스별로 가장 높은 확률을 가지는 문장과 해당 문장의 생성확률을 출력한다.

- 세부 구조

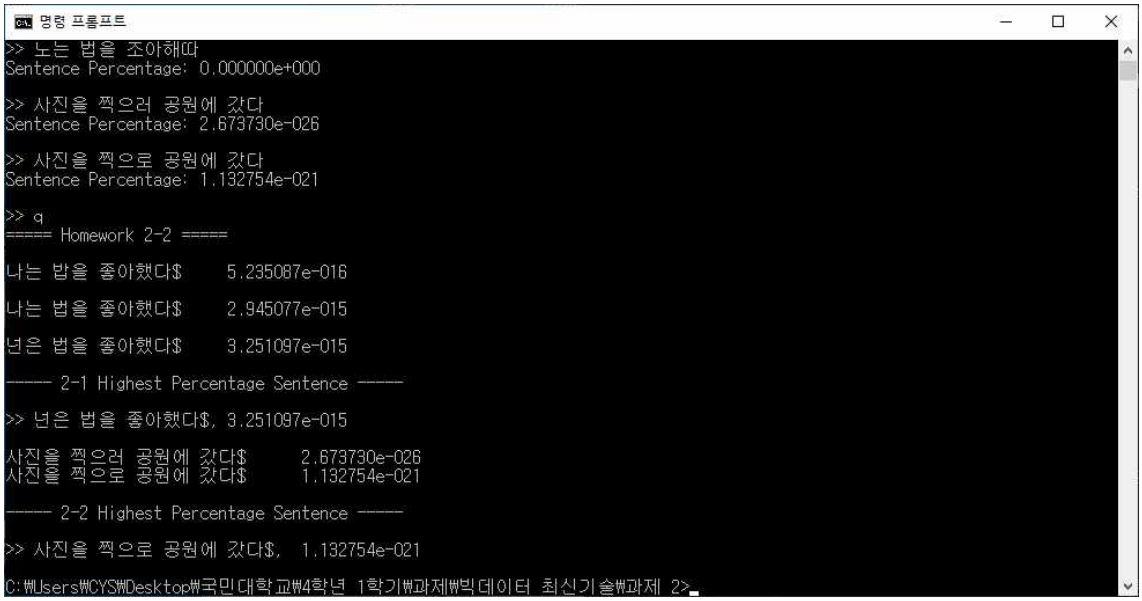
  각 음절의 생성확률을 계속해서 곱해나가면 확률값이 상당히 작아지기 때문에 %f 포맷으로 확률값을 출력하면 0.0으로 수렴한 값이 출력된다. 때문에 %f 포맷이 아닌 %e 포맷을 사용해 확률값을 출력하였다. 또한 어절단위의 공백(‘_’)을 고려해야 하기 때문에 “가_”의 경우에는 “가”에 해당하는 인덱스로 cntBlank를 count 해주었고, “_가”의 경우에는 “가”에 해당하는 인덱스로 cntBlankStart를 count 해주었다. 이 외의 “가나”와 같이 한글 음절이 연속해서 나오는 경우에는 bigramCnt를 count 해주었다.
  해당 과정에서 대용량 말뭉치 중 “ 가?"\r\n ”의 경우는 처리해주지 못해 KCC940_Korean_sentences_EUCKR.txt 말뭉치와 gtlee.txt 말뭉치는 정상적으로 bigram 음절을 count 해주었지만 해당 문자열이 포함된 RAW2169-CORE.txt의 경우는 counting 과정에서 runtime error가 발생해 프로그램이 종료되는 문제가 발생하였다.

2. 실행화면 스크린샷

1) KCC940_Korean_sentences_EUCKR.txt

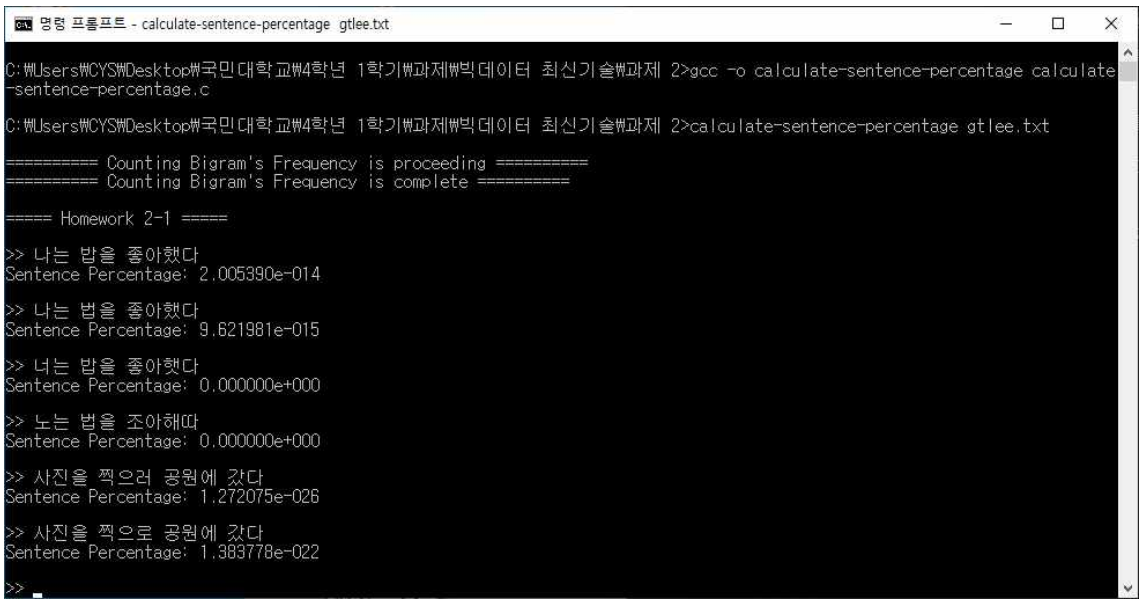


- ‘너는 밥을 좋아햇다’와 ‘노는 법을 조아해따’의 경우 각각 ‘아햇’&‘햇다’ 와 ‘해따’의 출현 확률이 0.0이기 때문에 생성확률이 0.0인 것이 확인되었다.

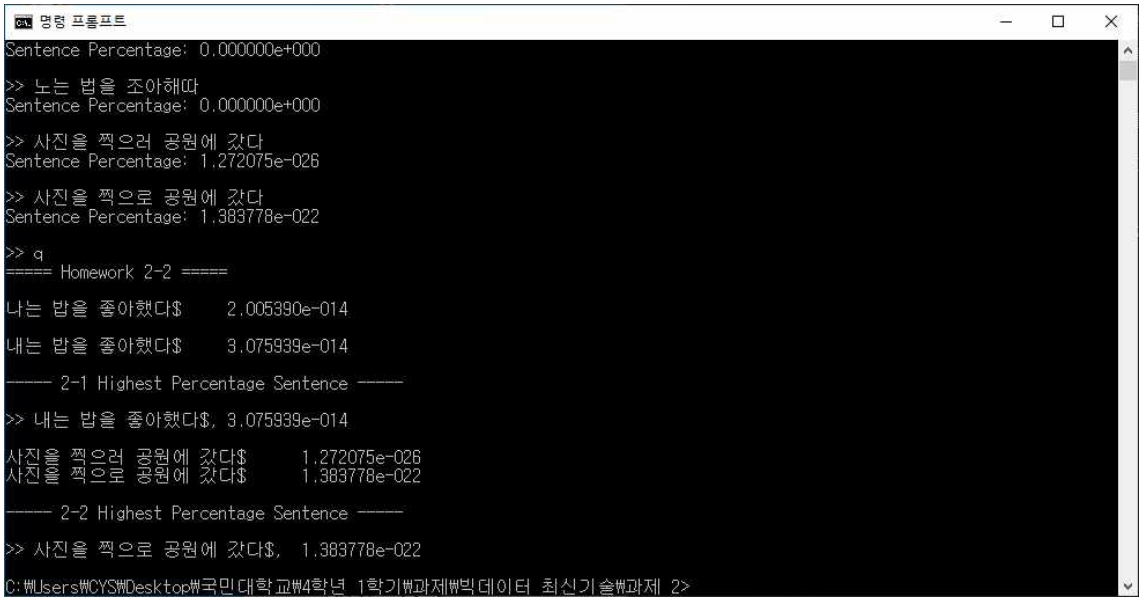


- 프로그램 내부에서 생성된 문장 중 '나는 밥을 좋아했다'와 '나는 법을 좋아했다', '넌은 법을 좋아했다'가 각각 가장 높은 확률의 문장으로 대체되었지만 이 중 '넌은 법을 좋아했다'가 가장 높은 확률의 문장이 되었다.
- 마찬가지로 '사진을 찍으러 공원에 갔다'와 '사진을 찍으로 공원에 갔다'가 각각 가장 높은 확률의 문장으로 대체되었지만 이 중 '사진을 찍으로 공원에 갔다'가 가장 높은 확률의 문장이 되었다.

2) gtlee.txt



- '너는 밥을 좋아햇다'와 '노는 법을 조아해따'의 경우는 KCC940_Korean_sentences_EUCKR.txt와 같은 이치로 생성 확률이 0.0이 되었고, 각 문장의 생성 확률이 KCC940_Korean_sentences_EUCKR.txt와는 차이가 있음을 확인할 수 있었다.



- 첫 번재 문장과 두 번째 문장 모두 가장 높은 생성 확률을 가지는 문장이 KCC940_Korean_sentences_EUCKR.txt와는 다른 문장으로 결정되는 것을 확인할 수 있었다.