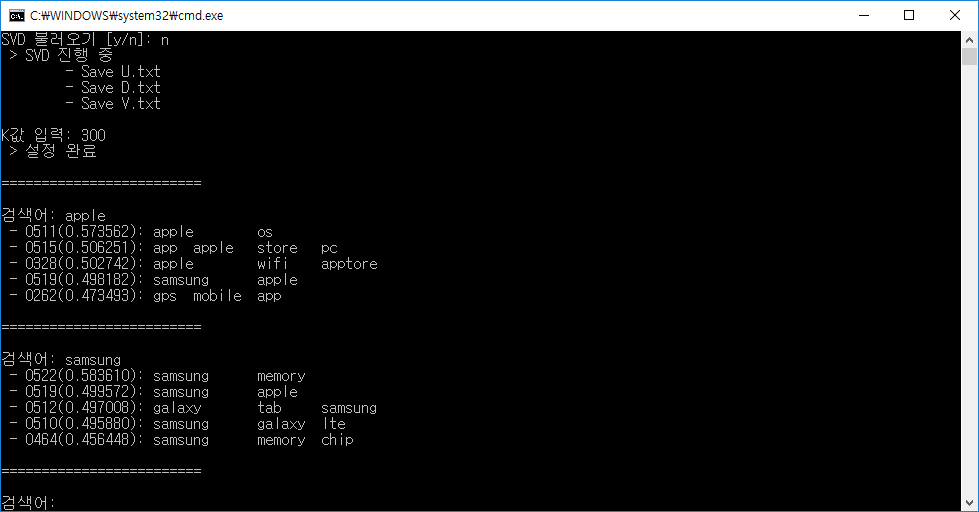
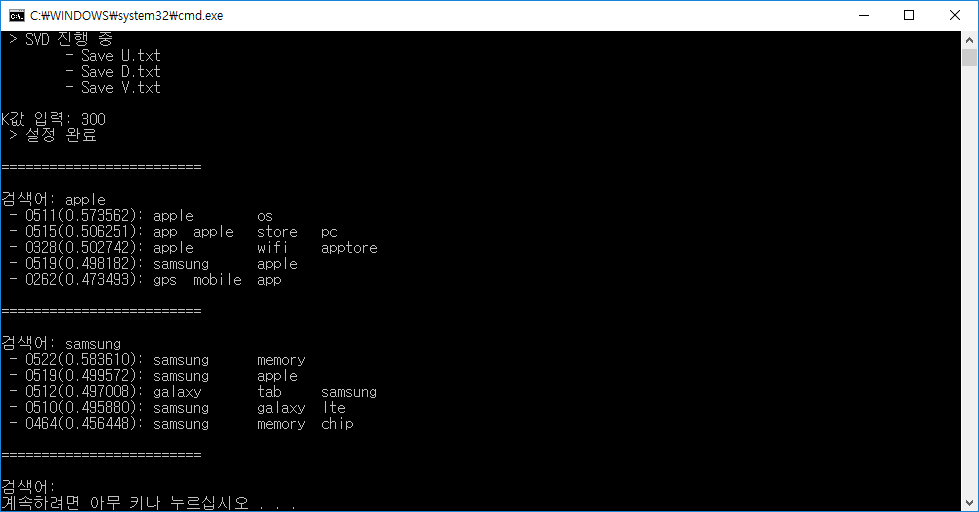
***수치 컴퓨팅 및 GPU 프로그래밍 – 숙제 3***

***LSI(Latent Semantic Indexing)***

120170420 신광수

* 사용자 인터페이스 설명
  + 시작하면 사용자는 기존에 생성한 SVD의 결과로 만들어진 U D V 행렬을 불러올 것인 지 정한다.
  + 
  + Y 또는 N을 사용하여 입력하며 그 외의 입력은 무시된다.
  + Y를 선택할 시, U.txt, D.tvt, V.tvt 파일로부터 결과값을 읽어온다.

(단, Y 선택 시 U, D, V 파일이 없는 상태는 가정하지 않는다.)

* + N를 선택할 시, SVD를 실행하고 U D V 행렬을 파일로 저장한다.
  + 그 이후에는 이 행렬들의 차원을 줄일 K를 설정한다.
  + 
  + K에 따라 검색 속도 및 정확도가 다소 차이 날 수 있다.
  + K를 입력한 이후에는 설정 완료라는 창이 뜨며, 그 이후부터 검색어를 입력하면 된다.
  + 
  + 검색 결과를 score 별로 최대 5개를 보여주며, 만약 프로그램을 종료하고 싶다면, 검색어에 아무것도 입력하지 않고 엔터를 치면 종료된다.
  + 
* 문서 추출 S/W가 제대로 작동하고 있는가? 그렇다는 사실을 어떻게 보일 수 있는가?
  + K의 값을 100으로 하고, 임의의 검색어를 10개 정도 뽑아 실험을 한 결과 대체로 검색어가 들어가거나 유사한 검색어가 있는 문서를 추출하고 있음을 알 수 있었다.
  + 
  + 
* LSI 방법의 구현 과정에서 SVD 문제를 풀기 위하여 어떤 공개 코드를 어떻게 사용했는가?
  + 이를 위해 netlib 에서 slatec(LINPACK) 라이브러리 중 하나인 dsvdc.f를 사용하였다. 이는 Fortran 코드로 되어있는 오픈 소스이며, 이 코드는 행렬 A에 대해 SVD한 결과를 출력하여준다. 이 함수의 파라미터는 다음과 같다.
  + SUBROUTINE DSVDC (X, LDX, N, P, S, E, U, LDU, V, LDV, WORK, JOB, INFO)
  + X는 입력 행렬이며, LDX는 X 행렬의 차원, N은 X행렬의 행, P는 X행렬의 열 값을 의미한다. LDU는 U 행렬의 차원, LDV는 V행렬의 차원, WORK는 작업을 위한 행렬 공간을 의미하며, JOB은 어떤 방식으로 SVD를 실행할 것인지 정하는 변수이다. 이 함수를 통해 나오는 결과값인 S는 X의 singular value들을, E는 전체가 0이여야 하지만 만약 에러가 있을 시 정보를 제공하며, U는 right singular vector, V는 left singular vector의 값을 가지고 있다.
  + 이 결과는 Fortran 코드이므로 모두 Column-major로 구성되어 있다.
* 서로 다른 크기의 K값에 대해 속도 및 정확성 등의 관점에서 본 S/W의 특성이 어떻게 변하는가?
  + K를 [50, 100, 500, 1000] 로 바꾸며 세 개의 검색어 [3g, nvidia, linux] 에 대해 검색을 진행하였다. 그 때의 평균 속도 및 추출되는 문서의 상태는 다음과 같았다.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 50 | | | | | 100 | | | | |
|  | 3g | nvidia | linux | 평균 |  | 3g | nvidia | linux | 평균 |
| 수행속도 (ms) | 0.164 | 0.172 | 0.19 | 0.175333 | 수행속도 (ms) | 0.291 | 0.347 | 0.348 | 0.328667 |
| top5 내 오류 개수 | 2 | 0 | 2 | 1.333333 | top5 내 오류 개수 | 0 | 0 | 0 | 0 |
| 500 | | | | | 1000 | | | | |
|  | 3g | nvidia | linux | 평균 |  | 3g | nvidia | linux | 평균 |
| 수행속도 (ms) | 1.391 | 1.481 | 1.486 | 1.452667 | 수행속도 (ms) | 2.942 | 2.805 | 3.003 | 2.916667 |
| top5 내 오류 개수 | 0 | 0 | 0 | 0 | top5 내 오류 개수 | 0 | 0 | 0 | 0 |

* + K의 값에 따라 정확도는 큰 차이를 보이지 않았으나 수행 속도는 점차 증가하는 것을 볼 수 있었다. 이는 행렬의 곱셈에 필요한 수가 늘어남에 따라 당연한 결과라고 볼 수 있다.
  + K의 값을 줄여나간다는 것은 더 높은 수준으로 A 행렬을 근사하는 것이라 볼 수 있으므로 정확도가 점차 떨어지는 것을 볼 수 있었다.
  + 본 실험을 통해 일정 수준으로 K를 늘려나가는 것은 큰 의미가 없음을 알 수 있다.
  + 다만 신기한 것은 K의 값에 따라 검색되는 문서가 달라진다는 점이었다. 이 점은 SVD 행렬을 근사함에 따라 다소 score에 변화가 있었고, 이에 따라 top 5가 변화하는 것으로 생각한다.
  + 
* 자신이 문서 추출과 관련하여 본 S/W의 질을 높이기 위하여 어떤 특별한 노력을 기울였는가?
  + SVD 값은 한 번 계산하면 바뀌지 않으므로 계산한 값을 저장하여 사용할 수 있도록 프로그램을 수정하였다.
  + 특별한 것은 하지 않았지만, 속도를 높이기 위해 미리 document\_collection\_matrix라는 함수를 통해 V\*D 를 미리 계산하여 속도를 높이고자 하였다.