2008 Spring: Programming Methodologies

3rd Project Description

Q-Gram Full Text Indexing and Searching

1. 목표

Q-그램을 이용한 풀 텍스트 (full text) 검색엔진을 구현한다. 풀 텍스트 검색이란 문서에 포함된 모든 부분 문자열에 대한 검색을 뜻한다. 본 프로젝트는 풀 텍스트 검색이 가능하도록 문서를 색인하고 검색하는 기능을 가진 프로그램 작성을 목표로 한다.

2. 설명

1) Q-그램

Q-그램이란 Q개의 문자(character)로 이루어진 단어를 뜻한다. 다음 예를 통해 알아보자.

문자열
String
3-그램
Str, tri, rin, ing

3-그램의 경우 예와 같이 한 글자씩 뒤로 옮겨가며 (sliding) 뽑아낸 3개의 문자로 이루어진 문자열을 말한다.

2) 검색

검색이란 찾고자 하는 문자열(쿼리)가 나타나는 모든 문서를 돌려주는 기능을 말한다. 다음과 같이 세 개의 문서가 있다고 해보자.

문서1	A Q-gram in this context refers to a sequence of letters, q letters long, from a given word.
문서2	For example, for q = 2, the word Ne**lson** has the following q-grams: NE EL LS SO ON
문서3	By comparison, Nei**lson** breaks down into these q-grams (q = 2): NE EI IL LS SO ON

"lson"이란 쿼리로 검색을 한다면 검색결과는 문서2와 문서3이 될 것이다.

2) 색인

위와 같은 검색을 할 때 매번 모든 문서를 읽어 어떤 문서에 쿼리가 나타나는 지 체크한다면 많은 시간과 연산이 필요할 것이다. 만약 구글이 100억개가 넘는 문서를 사용자가 쿼리를 입력할 때마다 모두 뒤져본다면 결과를 얻는데 몇 일이 걸릴지도 모른다. 따라서 구글과 같은 검색엔진들은 일반적으로 사용자가 입력할 키워드들에 대해 어떤 문서에서 그 키워드가 나타나는지에 대한 정보를 저장하고 있다. 예를 들어 위의 예에서 키워드 "gram"은 문서1, 2, 3에서 나타나고 "lson"은 문서2, 3에서 나타난다는 정보를 미리 저장한다. 이와 같은 일을 색인 또는 역색인(inverted indexing)이라 한다.

3. 구현 내용

1) 색인 기능

문서를 읽어 3-그램을 사용하여 키워드를 생성하고 모든 키워드에 대해 키워드가 나타나는 문서의 위치를 파일에 저장한다. 다음 예를 보면서 프로세스를 이해하자.

문서1	I am a boy.
문서2	I am a girl.
문서3	Boys and girls.

각 문서에서 모든 3-그램을 생성하고 각 3-그램에 대해 문서 번호와 3-그램이 몇 번째 위치에서 나타나는지를 메모리에 저장한다. (위치를 저장하는 이유는 다음에 설명한다) 3-그램은 단어 단위로 생성하며 ‘, . !’등 특수 문자는 제외시키고 대소문자 구분을 하지 않도록 모든 알파벳을 대문자 혹은 소문자로 변환하여 3-그램을 생성한다. 또한 3글자가 되지 않는 단어는 색인하지 않는다. (3단어 이상의 단어만 검색한다고 가정한다; 단, 3글자 미만의 검색에 대해서도 구현을 한다면 추가 점수를 줄 것이다.)

(BOY:1:5), (GIR:2:5), (IRL:2:6), (BOY:3:1), (OYS:3:2), (AND:3:5), (GIR:3:8), (IRL:3:9), (RLS:3:10)

이와 같이 저장한 (3-그램, 문서번호, 위치)들을 Q-그램의 알파벳순, 다음은 문서의 번호순, 다음은 위치 순의 우선순위로 정렬한다. 그리고 각 고유한(distinct) 3-그램의 문서 번호와 위치 쌍을 파일에 저장한다. 이와 같이 키워드가 나타나는 문서번호, 위치 등의 나열을 포스팅 리스트(posting list)혹은 역색인리스트 (inverted index list)라고 부른다.

BOY	1:5 3:1
GIR	2:5 3:8
IRL	2:6 3:9

OYS	3:2
AND	2:5 3:8
RLS	3:10

위 포스팅 리스트는 파일에 저장해야 하며 각각의 파일에 저장해도 좋다. 예를 들어 위와 같은 포스팅 리스트를 각각, *BOY.pls, GIR.pls, IRL.pls, OYS.pls, AND.pls, RLS.pls*와 같은 파일이름으로 저장하도록 한다.

2) 검색 기능

쿼리로 들어온 문자열이 나타나는 모든 문서를 찾아 그 문서 번호를 출력한다. 쿼리의 길이가 Q=3를 넘으면 쿼리를 3-그램으로 자르고 그 모든 3-그램이 포함되어 있는 문서를 찾도록 한다. 다음 예를 통해 알아보자.

쿼리	Girls
쿼리의 3-그램들	GIR, IRL, RLS

Girls를 포함하는 문서를 찾고자 한다면 Girls의 모든 3-그램, GIR, IRL, RLS를 포함한 공통의 문서를 찾아야 한다. 따라서 각 3-그램의 포스팅 리스트를 저장해 놓은 파일에서 읽어와 교집합을 계산한다.

GIR	2:5 3:8
IRL	2:6 3:9
RLS	3:10

위의 예에서는 3번 문서가 답이 될 것이다. 단, 교집합과 함께 문자의 위치가 하나씩 증가하는 경우만 결과로 출력해야 한다. 위의 예를 보면 GIR은 3의 8번째, RLS은 3의 9번째, RLS는 3의 10번째에 하나씩 증가 하는 위치에서 나타나므로 GIRLS란 문자열이 문서 3에서 나타난다고 대답할 수 있다. 만약 위치를 고려하지 않는다면 Girls가 아닌 Girrrrrrrrrls를 포함하는 문서도 답을 할 것이다.

교집합을 할 때에는 각 포스팅 리스트가 문서번호, 위치에 따라 정렬되어 있으므로 앞에서부터 포인터를 하나씩 이동하며 문서번호가 같으면 위치를 체크하고 다르다면 작은 쪽의 포인터를 증가시키는 방식으로 비교하도록 한다.

3) 입출력

실행파일은 색인 프로그램과 검색 프로그램 두 가지를 컴파일 하도록 한다.

색인 실행파일에게 주어질 입력은 다음과 같다.

```
$ index inputfilelist.txt
```

텍스트형식 파일 inputfilelist.txt는 다음과 같이 한 라인에 색인 할 문서 파일명 하나를 갖는다.

```
1.txt
3.txt
10.txt
4.txt
9.txt
7.txt
```

확장자(.txt)를 제외한 파일명을 문서번호로 사용하며 위 예와 같이 파일명은 정수형 숫자이다.

검색 실행파일에게 주어질 입력은 다음과 같다.

```
$ search query
```

Query를 포함하는 모든 문서번호를 다음과 같이 한 라인에 하나씩 출력한다.

```
1
4
9
10
```

실행 파일은 각각 index, search로 생성하여야 한다.

4. 기타

1) 영문 문서를 색인, 검색한다고 가정한다. (한글 등 multi-byte 제외)
2) 3-그램을 사용한다. (Q=3)
3) 쿼리는 항상 3글자 이상이라고 가정하고 3글자 미만의 글자는 색인, 검색하지 않아도 된다. 단, 3글자 미만의 글자도 색인, 검색이 되도록 구현하면 20점의 추가 점수를 받을 수 있다. 위 추가 사항을 구현했을 경우 보고서 첫 페이지에 눈에 띄도록 표시해주기 바란다.
4) 구현언어는 C++만 허용한다.

제출사항

1. 제출 방법 및 내용

   1) 프로젝트 구현 파일 제출 (DEV C++을 사용하세요!!!)

      - 파일은 ee 홈페이지에 과제 제출을 이용하여 제출한다.
      - 소스 파일과 컴파일을 할 수 있는 파일과 실행파일을 압축하여 제출한다.
      - 각 제출 파일 이름은 다음의 형식을 따른다.

        pro_3_학번.zip

        예) pro_3_2007-1234?.zip

      - 각 소스 파일 상단에 **E-mail주소와 작성자 학번 이름을** 적는다.

   2) 보고서 제출

      간단한 설명과 discussion을 하드카피로 수업시간 전에 제출한다.

2. 제출 일시

   파일 제출 마감일: **2008년 5월 19일과 5일 20일사이 새벽 0:00**

   구현 파일 제출 시각은 홈페이지 과제 제출 시각에 준한다.

   보고서는 2008년 5월 20일 수업시간에 제출

딜레이는 하루에 10%씩 감점

Copy 발견 시 모든 숙제 0점 처리