산업컴퓨팅개론 Assignment #4 2010-12086 황영석

**Web Crawler**

Subject : Web site를 crawling하여 단어를 수집한 후 이를 이용하여 informative한 tag cloud를 만드시오

**<Subject Concept>**

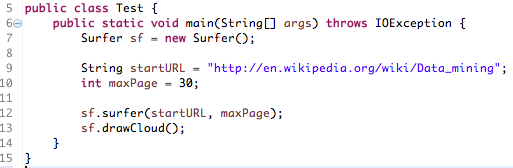
Wikipedia에는 수많은 위키 페이지들이 존재한다. 그리고 각 위키 페이지에는 수많은 링킹단어들이 존재한다. 링킹단어를 클릭하면 또다른 위키 페이지로 이동하게 된다.

페이지 A가 포함하고 있는 링킹단어를 통해 이동한 위키 페이지들의 집합에는(one-depth wiki-pages) A와 관련된 여러 링킹 단어들이 또 존재할 것이다. 그런데 이 집합에 공통적으로 나오는 링킹단어들은 A와 연관성이 있을 것이라고 생각할 수 있다. 예를 들어, 위키피디아의 Data\_mining 위키페이지엔 machine\_learning, data\_analysis 등의 링킹단어가 있는데, 이 위키페이지들에 들어갔을 때 존재하는 링킹단어들을 추려내면 Data\_mining과 관련된 단어들을 얻을 수 있을 것이다. 이런 믿음에서 위키피디아를 크롤링해서, 하나의 위키페이지와 연관되어 있는 단어들을 태그 클라우드로 만들기로 하였다.

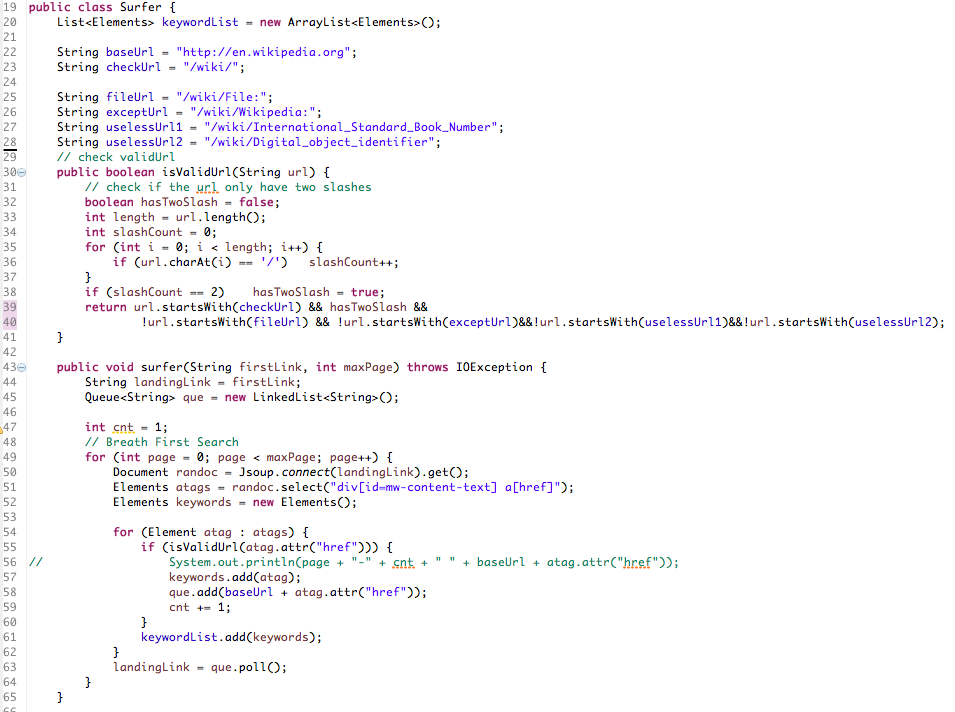
링킹단어들은 **div[id=mw-content-text]**의 **a** 태그를 가졌다. 그런데 a 태그는 하이퍼링크가 걸려있으면 모두 가졌으며, 이 중에서 유의미한 링킹단어들만 골라내는 작업이 필요했다. 예를 들어 확대되는 그림 파일, [edit] 링크 등의 잡다한 하이퍼링크들은 제외했어야 했다. 그 과정을 Surfer.java의 앞부분에 구현하였다.

**<CODE>**

[Test.java]



Test.java는 main함수가 있는 클래스이다. 시작 페이지를 받고, Surfer 클래스의 surfer 메쏘드와 drawCloud 메쏘드를 실행한다.



[Surfer.java]

Surfer 클래스는 크게 surfer메쏘드와 drawCloud메쏘드로 구성되어 있다

(22~41) 앞부분에서는 junk a-tag들을 걸러내는 작업을 하였다. 유의미한 a-tag의 href는 ‘/wiki/machine\_learning’ 식으로 구성되는데, 이에 맞지 않는 href를 갖고 있는 특징적인 링크들은 모두 제외하였다. 여러가지 heuristic한 방법이 사용되었다.

(49~64) surfer 메쏘드는 BFS(Breath First Search)로 첫 시작페이지의 maxPage수만큼의 one-depth pages에 들어가서 유의미한 a-tag를 가져오는 작업을 구현하였다. Queue를 이용하여 구현하였는데, 원래 idea는 depth를 좀 더 깊게 한다든지 시작 페이지의 모든 유의미한 링킹단어들에 대해서 Search를 하는 것이었다. 그러나 태그 클라우드는 사용해보니 띄울 수 있는 단어의 갯수가 제한적이고, maxPage = 30개 정도의 앞부분 링킹단어만 서치하여도 충분하여서 그냥 위와 같이 구현하였다.(위키페이지의 앞부분에는 대부분의 중요 단어들이 집중되어 있다.)

(67~이후) 에는 태그 클라우드를 GUI로 띄우는 것을 구현하였다. 별로 어렵지 않은 부분이므로 보고서에서는 생략합니다.