(http://www.flickr.com/people/chrisw/)

(http://www.comp.nus.edu.sg)

Information Retrieval

NUS SoC, **2013/2014**, Semester II Video Conferencing Room (COM1 02 VCRm) / Fridays 11:00-13:00

Last updated: Thursday, March 20, 2014 09:59:45 AM SGT Brought over from 2013.

Homework #4 » Patent Retrieval Mini Project

patsnap (http://www.patsnap.com)

In our final Homework 4, we will hold an information retrieval contest with real-world documents and queries: the problem of patent retrieval. As described in lecture, patent retrieval is a case where recall is particularly important, as it is important to not miss any relevant documents (a requirement common to search engines working in the area of law).

Jump to the competition framework (http://wing.comp.nus.edu.sg/~wing.nus/cs3245/hw4/), the current leaderboard (http://wing.comp.nus.edu.sg/~wing.nus/cs3245/hw4/leaderboard.html), or 2013 leaderboard (http://wing.comp.nus.edu.sg/~wing.nus/cs3245/hw4/13/2013_leaderboard.html).

Commonalities with Homeworks #2 and #3

The indexing and query commands will use an (almost) identical input format to Homeworks #2 and #3, so that you need not modify any of your code to deal with command line processing. To recap:

Indexing: `$ python index.py -i directory-of-documents -d dictionary-file -p postings-file`

Searching:

```
$ python search.py -d dictionary-file -p postings-file -q query-file -o output-file-of-results
```

The difference from Homeworks #2 and #3 is that the `query-file` specifies a single query, and not a list of queries.

However, significantly different from the previous homework, we will be using a patent corpus, provided by PatSnap (http://www.patsnap.com), a company with origins partially from NUS (Disclaimer: I have no interest or affiliation with PatSnap, although one alumnus from my group is working there.)

> **Problem Statement**: Given 1) a patent corpus (to be posted to IVLE) as the candidate document collection to retrieve from, and 2) a set of free text information needs, return the list of the IDs of the relevant documents for each need, in sorted order or relevance. Your search engine should return the entire set of relevant documents (don't threshold to the top K relevant documents; as described, recall is important in patent search).
> Your system should return the results for the query `query-file` on a single line. Separate the IDs of different documents by a single space ' '. Return an empty line if no patents are relevant.

For this assignment, no holds are barred. You may use any type of preprocessing, post-processing, indexing and querying process you wish. You may wish to incorporate or use other python libraries or external resources; however, for python libraries, you'll have to include them with your submission properly -- I will not install new libraries to grade your submissions.

PatSnap, the company we are working with for this contest, is particularly interested in good IR systems for this problem and thus is cooperating with us for this homework assignments. They have provided the corpus (the patents are in the public domain, as is most government information) and relevance judgments for a small number of queries. Teams that do well may be approached by PatSnap to see whether you'd like to work further on your project to help them for pay. Note: Your README will be read by both Min and the PatSnap team, but your code will not be given to their team to use; if they are interested in what you have done, you may opt to license your work to them.

More detail on the inputs: Information Needs and Patents

The patents and the information needs have a particular structure in this task. Let's start with the information needs.

Information Need: We call the inputs *information needs*, as they describe the relevant documents at a semantic level, and not (necessarily) at the shallow, language level that the queries given to the search engine will have to execute. The needs will be given in a format similar to TREC queries. They will have a `title` field, which is a short noun phrase or sentence describing the information need. A `description` field will give more detail on what the relevant documents may or may not contain (will always start with "relevant documents will describe". Here is an example information need (also provided in the workbin):

```
<?xml version="1.0" ?>
<title>
  Washers that clean laundry with bubbles
</title>
<description>
  Relevant documents will describe washing technologies that clean or
  induce using bubbles, foam, by means of vacuuming, swirling, inducing
  flow or other mechanisms.
</description>
```

In PatSnap's own system, searchers need to transform these needs into actual search queries. For the above need, a patent engineer transformed it into the following Boolean query:

```
((bubble AND fine) OR microbubble)
```

This requires some human knowledge from the patent engineer to do, as "fine" and "microbubble" don't appear any where in the description or title of the query. This is shown for illustrative purposes, please don't interpret this as an actual step you'll need to do for your assignment. Note that this transformation 1) was done to deal with the Boolean nature of their search engine, and 2) may not reflect the best method to transform the need into a query.

Patents:

Patents are structured documents. For the purposes of our assignment, we are going to use an XML representation of a patent. Below is a document, ID EP2067524A1, which is relevant to the above query:

```
<?xml version="1.0" ?>
<doc>
<doc>
  <str name="Patent Number">EP2067524A1</str>
  <str name="Application Number">EP2007828700</str>
  <str name="Kind Code">A1</str>
  <str name="Title">SWIRLING FLOW PRODUCING APPARATUS, METHOD OF PRODUCING SWIRLING FLOW, VAPOR PHASE GENERATING APPARATUS, MICROBUBBLE GENERATING APPARATUS, FLUID MIXER AND FLUID INJECTION NOZZLE</str>
  <str name="Abstract">
    There are provided a fluid injection nozzle, a fluid mixer, a microbubble generating apparatus, a vapor phase generating apparatus, a method of producing swirling flow, and a swirling flow producing apparatus that can be applied to any kind of fluid and can efficiently generate a swirling flow at high speed.
    The swirling flow producing apparatus includes a housing and a cylindrical member. The housing includes a cylindrical portion of which at least one end is opened, and a fluid introducing passage that is opened on an inner peripheral surface of the cylindrical portion. The cylindrical member is provided in the cylindrical portion of the housing. The cylindrical member includes a cylindrical portion of which at least one end in a direction corresponding to an opening direction of the cylindrical portion is opened, and holes formed in a peripheral wall of the cylindrical portion. A fluid introduced from the fluid introducing passage flows into the cylindrical portion of the cylindrical member through the holes so as to generate a swirling flow, and flows out from the housing and the cylindrical member.
  </str>
  <str name="Document Types">EP | EPA | DOCDB</str>
  <str name="Application Date">2007-09-28</str>
  <str name="Application Year">2007</str>
  <str name="Application(Year/Month)">2007-09</str>
  <str name="Publication Date">2009-06-10</str>
  <str name="Publication Year">2009</str>
  <str name="Publication(Year/Month)">2009-06</str>
  <str name="All IPC">B05B1/34 | B01F5/00</str>
```

```
20.     <str name="IPC Primary">B05B1/34</str>
21.     <str name="IPC Section">B</str>
22.     <str name="IPC Class">B05</str>
23.     <str name="IPC Subclass">B05B</str>
24.     <str name="IPC Group">B05B1</str>
25.     <str name="Family Members">KR1020090028835A | WO2008038763A1 | CN10150585
      9A | US20090201761A1 | EP2067524A1</str>
26.     <str name="Family Member Count">5</str>
27.     <str name="Family Members Cited By Count">1</str>
28.     <str name="Other References">See references of WO  2008038763A1</str>
29.     <str name="Other References Count">1</str>
30.     <str name="Cited By Count">0</str>
31.     <str name="Priority Country">JP</str>
32.     <str name="Priority Number">2006264652</str>
33.     <str name="Priority Date">2006-09-28</str>
34.     <str name="Assignee(s)">NAKATA COATING CO., LTD.</str>
35.     <str name="1st Assignee">NAKATA COATING CO., LTD.</str>
36.     <str name="Number of Assignees">1</str>
37.     <str name="1st Assignee Address">82, Higashikawashima-cho, Hodogaya-ku, Y
      okohama-shi, Kanagawa 240-0041, JP</str>
38.     <str name="Assignee(s) Address">82, Higashikawashima-cho, Hodogaya-ku, Yo
      kohama-shi, Kanagawa 240-0041, JP</str>
39.     <str name="Inventor(s)">MATSUNO, TAKEMI | NAKATA, AKIO</str>
40.     <str name="1st Inventor">MATSUNO, TAKEMI</str>
41.     <str name="Number of Inventors">2</str>
42.     <str name="1st Inventor Address">NAKATA, COATING, CO., LTD., 82, Higashik
      awashima-cho, Hodogaya-ku, Yokohama-shi, Kanagawa, 240-0041, JP</str>
43.     <str name="Inventor(s) Address">NAKATA, COATING, CO., LTD., 82, Higashika
      washima-cho, Hodogaya-ku, Yokohama-shi, Kanagawa, 240-0041, JP | NAKATA, CO
      ATING, CO., LTD., 82, Higashikawashima-cho, Hodogaya-ku, Yokohama-shi, Kana
      gawa, 240-0041, JP</str>
44.     <str name="Agent/Attorney">HOFFMANN, ECKART</str>
45.     <str name="cited by within 3 years">0</str>
46.     <str name="cited by within 5 years">0</str>
47.   </doc>
```

You will notice that there are a lot of fields in the patent. However, not all fields things are relevant to assessing a patent's relevance to the query (and thus you may not want to index them), but are included for the sake of completeness.

in particular, the IPC (International Patent Classification) (http://www.wipo.int/classifications/ipc/en/) is a useful piece of data that you may want to use to assess the relevance of a document. It is a hierarchical classification of the patent into a ontology. However you may need to parse this information in some way to make it useful to your system.

Zones and Fields

As introduced in Week 8, **Zones** are free text areas usually within a document that holds some special significance. **Fields** are more akin to database columns (in a database, we would actually make them columns), in that they take on a specific value from some (possibly infinite) enumerated set of values.

Along with the standard notion of a document as a ordered set of words, handling either / both zones and fields is important for certain aspects of patent retrieval.

Query Expansion

You might notice that many of the terms used in the patents themselves do not overlap with the query times used. This is known as the *anomalous state of knowledge (ASK) problem* or *vocabulary mismatch*, in which the searcher may use terminology that doesn't fit the documents' expression of the same semantics. A simple way that you can deal with the problem is to utilize **query expansion**.

In this technique, we use a first round of retrieval on the query terms used by a searcher to find some sample documents. Assuming that these documents are relevant, we can extract sometimes found these documents or use the entire documents themselves as queries, used in a second round of retrieval. The idea is that the sample documents have terminology that match the document corpus, overcoming the problem of vocabulary mismatch.

What to turn in?

You are required to submit `index.py`, `search.py`, `dictionary.txt`, and `postings.txt`. Please do not include the patent corpus.

Submission Formatting

You are allowed to do this assignment individually or as a team of up to 4 students. There will be no difference in grading criteria if you do the assignment as a large team or individually. For the submission information below, simply replace any mention of a matric number with the matric numbers concatenated with a separating dash (e.g., A000000X-A000001Y-A000002Z). Please ensure you use the same identifier (matric numbers in the same order) in all places that require a matric number

For us to grade this assignment in a timely manner, we need you to adhere strictly to the following submission guidelines. They will help me grade the assignment in an appropriate manner. You will be penalized if you do not follow these instructions. Your matric number in all of the following statements should not have any spaces and any letters should be in CAPITALS. You are to turn in the following files:

- A plain text documentation file `README.txt`: this is a text only file that describes any information you want me to know about your submission. You should not include any identifiable information about your assignment (your name, phone number, etc.) except your matric number and email (we need the email to contact you about your grade, please use your A*******@nus.edu.sg address, not your email alias). This is to help you get an objective grade in your assignment, as we won't associate matric numbers with student names. **You should use the `README.txt` template given to you in Homework #1 as a start. In particular, you need to assert whether you followed class policy for the assignment or not.**
- All source code. We will be reading your code, so please do us a favor and format it nicely. Again, if you're using external libraries, make sure to include some nicely so they play well with our default ir_env environment (and acknowledge your external libraries as a source of help in your submission).

These files will need to be suitably zipped in a single file called `<matric number>.zip`. Please use a zip archive and not tar.gz, bzip, rar or cab files. Make sure when the archive unzips that all of the necessary files are found in a directory called <matric number>. Upload the resulting zip file to the IVLE workbin by the due date: **Updated** 11 April 2014, 11:59:59 pm SGT. There will absolutely be no extensions to the deadline of this assignment. Read the late policy if you're not sure about grade penalties for lateness (grading.html#late).

Grading Guidelines

The grading criteria for the assignment is below. You should note that there are no essay questions for this assignment.

- 30% Documentation. This component is graded with a higher weightage in this assignment than in previous ones. This component breaks down into the following two subcomponents:
  - 5% For following the submission instructions and formatting your documentation accordingly.
  - 5% For code level documentation.
  - 10% For the originality of your ideas. Submissions attempting to do something unusual or interesting will be given higher marks. Sometimes attempting an interesting algorithm may have negative consequences on the performance of your system. In such cases, you can comment out the code that does not work as well. You should still document this code, so I can give you an appropriate documentation score. However, I will then assess your code's performance based on what actually runs from your submission.
  - 10% For your high level documentation, in your README document. This component comprises of an overview of your system, your system architecture and how your system deals with each of the optional components (query expansion, utilizing external resources, field/zone treatment, run-time optimizations, and the allocation of work to each of the individual members of the project team. Note that this component absorbs the weightage of the essay questions pounding previous assignments, so you should be especially thorough in your documentation.
- 70% Performance of your code. This component breaks down into several subcomponents:
  - 20% I will compare it against a baseline TF×IDF ranked retrieval implementation, in which the entire document is treated without zones (i.e., all zone/field information is removed). If your system works at least as good as the standard baseline, you will receive all 20% for this component
  - 35% We will use a competition framework to assign credit to teams and to show the leaderboard. Submissions will be auto-run with a 24 hour delay (to prevent phishing for relevant documents). You can make submissions starting 1 week after the homework assignment is open (e.g., 25 March).
  - 5% will be due to the time efficiency of your system to answer queries (not testable by the competition framework, since you provide the answers to it). Your system should be able to answer a query within two minutes. This requirement is mostly so that I can grade assignments in a timely manner. I may also use another computer aside from Sunfire to this test efficiency of your entries.

Hints

- If there are certain online services or APIs (e.g., web services), you wish to invoke in running your system, you **can** use these through function calls to external services. You may also wish to pre-compile certain (patent-specific) resources for use by your system at run-time. For using online resources, it maybe helpful to use python's utilities to retrieve web pages (URLs), and save it as a temporary page for more analysis. You may use temporary file names of the form: `temp-*.*` in the current directory for query processing.
- Working in a group can be helpful but can be counter-productive too. I have observed that group work tends to make homework submissions grades slightly higher than single submissions but also more average (it's harder to have an outstanding grade with a group). If you think you work better in a larger group, please do. Make sure to clearly partition the work and demarcate the API between your modules.
- While you don't need to print scores out for the official runs that your system will be graded on, you

may find it useful to include such information in your debugging output when developing your solution.

- The US Patent Office Patent Search (http://www.uspto.gov/patents/process/search/) and other patent search engines may be helpful for you to explore. I highly suggest that you do some patent searches yourself so that you can familiarize yourself both of the IPC classification and how patent search generally works.
- Similar to homework assignments #2 and #3, we will only be giving you a few queries to work with and (incomplete) query relevance judgments. We can only give you a few query relevance judgments, as the patent relevance process also takes time to do for our human expert at PatSnap to assemble. However, we suggest you use your peers to pose some queries yourself and assess whether they are relevant. Documentation and participation marks will be given to student teams who do this.
- Bulletproof your input and output. Make sure directories (e.g., arguments to `-i`) are correctly interpreted (add trailing slash if needed). Check that your output is in the correct format (docIDs separated by single spaces, no quotations, no tabs).
- If you're fishing for ideas about how to do patent retrieval, you might find past iterations of the NTCIR patent retrieval task (http://scholar.google.com.sg/scholar?hl=en&q=patent+retrieval) interesting. This is a yearly contest, like TREC, that features patent retrieval. You are encouraged, but not obliged, to use ideas from this research community.
- Finally, please note that the documents are in XML format, as are the queries. You will want to choose a suitable Python library (some of which are included directly in the Python system libraries) to process the data. Please plan on spending a bit of time to familiarize yourself with XML processing.
