Algorithmic Methods of Data Mining

Homework 2

Due: 11/12/2016, 23:59.

Instructions

You must hand in the homeworks electronically and before the due date and time.

Groups: For this homework you have to work in groups of 3 people. By **27/11/2016** you have to send your group by email to `aris@dis.uniroma1.it`. Groups have to be of exactly 3 people. If you do not have some member(s), send an email and we will match you.

Handing in: You must hand in the homeworks by the due date and time by an email to `aris@dis.uniroma1.it` that will contain as attachment **(not links to some file-uploading server!)** a .zip or a .tar.gz file with your answers and subject
`[Algorithmic Methods for Data Mining] Homework 2`
After you submit, you will receive an acknowledgement email that your project has been received and at what date and time. If you have not received an acknowledgement email within 2 days after the deadline then contact the instructor.

The solutions for the theoretical exercises must contain your answers either typed up or hand written clearly and scanned.

For information about collaboration, and about being late check the web page.

For this question you have to implement a search engine for recipes. It has several parts that you need to implement. For the linguistic analysis you can use the NLTK Python library.

1. First you need to download the recipes. We will use the recipes at `http://www.bbc.co.uk/food/recipes`. You will need to find a way to download all the recipes from the site. You can use any method you want. It is important to put some time delay between requests; at least 1sec between two requests. For that you can use the `time.sleep` command of Python.

2. After you download them you have to preprocess them. For each recipe store all the information (title, who wrote it, preparation time, cooking itme, number of people it serves, dietary information, ingredients, and method). Note that some fields (e.g., dietary information) may be missing. Store the final output as a large single tab-separated file. After that, you can do whatever preprocessing you think is essential (e.g., stopword removal, normalization, stemming).

3. The next step is to build a search-engine index. First, you need to build an inverted index, and store it in a file. Build an index that allows to perform proximity queries using the cosine-similarity measure. Then build also a query-processing part, which, given some terms it will bring the most related recipes. You can use any query-processing way that you prefer, although the project will be evaluated better if you use the algorithm with pointers that we covered in class.

4. **Extra credit:** Use your imagination to think of features you would like such a service to have. For example, you may want to weigh different the ingredients based on the quantity. You may want to try to find methods that take care of ingredients that are written in different ways, refering though to the same thing. You may want to give different weights to the title–ingredients–method. You may want to provide a query that satisfies some people, such as,

vegetarians, lactose intolerant, and so on. Other ideas might be on the presentation of the results. Or you can find photos from google images with final food, and so on.

Hand in the code, along with some examples of queries and screenshots of the results. Try short queries, such as a few terms, as well as long queries, which can be other recipes, where the goal would be, for example, to find the types of food that are more similar to *parmigiana di melanzane*.