Algorithmic Methods of Data Mining and Laboratory

Sapienza Università di Roma

Search Engine

S. Ballesteros*, F. Crema†, G. Nespoli‡

December 11, 2016

1 Introduction

Here we present a search engine writen in Python that is capable of indexing the information of the more than 11 000 recipes available at the BBC food section webpage and showing the most relevant results when searching a query through its graphical interface. In the following sections we describe the differents parts our work, as well as its authors: Information download, Preprocessing, Search engine and Graphical Interface.

2 Information download

The purpose of this part is to download as many recipes as possible from de BBC webpage, and to parse the relevant information to a TSV file. The proccess is as follows:

2.1 Finding the recipes links

The python module of this part is stored in *search/download/downloadData.py*. All the recipes links begin by *http://www.bbc.co.uk/food/recipes/*, in order to find those links, we investigated two sections of the BBC webpage, the first one is the ingredient section,

*sergio.ballesteros@estudiante.uam.es
†email
‡email

and the second one is their own search tool.

2.1.1 Ingredients section

Since each ingredient webpage link to several link recipes, we found the links to the ingredients webpages. In order to do so, our Python programm analyzes the following set of webpages *http://www.bbc.co.uk/food/ingredients/by/letter/[a-z]/* and then parses all the links that has the form *http://www.bbc.co.uk/food/[ingredient]*, which lead to the [ingredient] page.

After getting all the ingredient links, our programm explores each ingredient page to find the recipes links, which have the format *http://www.bbc.co.uk/food/recipes/[recipe]*. Both the links to the discovered recipes and the already explored ingredients are saved on the fly to two files on the disk. In this way, if the process is interrupted, the program can be ran again and by default it will load those files to resume the process in the point where it stopped.

```
Getting the ingredients of http://www.bbc.co.uk/food/ingredients/by/letter/a
Getting the ingredients of http://www.bbc.co.uk/food/ingredients/by/letter/b
Getting the ingredients of http://www.bbc.co.uk/food/ingredients/by/letter/c
Getting the ingredients of http://www.bbc.co.uk/food/ingredients/by/letter/d
Getting the ingredients of http://www.bbc.co.uk/food/ingredients/by/letter/e
Getting the ingredients of http://www.bbc.co.uk/food/ingredients/by/letter/f
Getting the ingredients of http://www.bbc.co.uk/food/ingredients/by/letter/g
Getting the ingredients of http://www.bbc.co.uk/food/ingredients/by/letter/h
Getting the ingredients of http://www.bbc.co.uk/food/ingredients/by/letter/i
Getting the ingredients of http://www.bbc.co.uk/food/ingredients/by/letter/j
Getting the ingredients of http://www.bbc.co.uk/food/ingredients/by/letter/k
Getting the ingredients of http://www.bbc.co.uk/food/ingredients/by/letter/l
Getting the ingredients of http://www.bbc.co.uk/food/ingredients/by/letter/m
Getting the ingredients of http://www.bbc.co.uk/food/ingredients/by/letter/n
Getting the ingredients of http://www.bbc.co.uk/food/ingredients/by/letter/o
Getting the ingredients of http://www.bbc.co.uk/food/ingredients/by/letter/p
Getting the ingredients of http://www.bbc.co.uk/food/ingredients/by/letter/q
Getting the ingredients of http://www.bbc.co.uk/food/ingredients/by/letter/r
Getting the ingredients of http://www.bbc.co.uk/food/ingredients/by/letter/s
Getting the ingredients of http://www.bbc.co.uk/food/ingredients/by/letter/t
Getting the ingredients of http://www.bbc.co.uk/food/ingredients/by/letter/u
Getting the ingredients of http://www.bbc.co.uk/food/ingredients/by/letter/v
Getting the ingredients of http://www.bbc.co.uk/food/ingredients/by/letter/w
Getting the ingredients of http://www.bbc.co.uk/food/ingredients/by/letter/y
Getting the ingredients of http://www.bbc.co.uk/food/ingredients/by/letter/z
### 1 / 1093 explored ingredients ### Total recipes 0 #### Fetching --->  http://www.bbc.co.uk/food/acidulated_water
### 2 / 1093 explored ingredients ### Total recipes 6 #### Fetching --->  http://www.bbc.co.uk/food/ackee
### 3 / 1093 explored ingredients ### Total recipes 10 #### Fetching --->  http://www.bbc.co.uk/food/acorn_squash
### 4 / 1093 explored ingredients ### Total recipes 12 #### Fetching --->  http://www.bbc.co.uk/food/aduki_beans
### 5 / 1093 explored ingredients ### Total recipes 13 #### Fetching --->  http://www.bbc.co.uk/food/egg_liqueur
### 6 / 1093 explored ingredients ### Total recipes 13 #### Fetching --->  http://www.bbc.co.uk/food/agar-agar
### 7 / 1093 explored ingredients ### Total recipes 20 #### Fetching --->  http://www.bbc.co.uk/food/ale
```

Figure 2.1: Our program output: in the top the program is finding the links to the ingredients webpages, and in the bottom is finding the link to the recipes inside each ingredient webpage.

Only about 5 000 recipies links can be found exploring directly the ingredients. On the other hand, on the BBC webpage, it is anounced that there are more than 11 000 recipes available. To our knowledge, the only way to find all of them is to use the search engine of the own webpage. In order to do so, we sent petitions to the webpage of

the format *http://www.bbc.co.uk/food/recipes/search?keywords=[ingredient]&page=[page number]*, where [page number] ranges from 1 to a maximum number that we parsed also from the page 1. The following image shows where to get this links, although our program created the set of all links in the mentioned format using as a base the links that the webpagesent us.

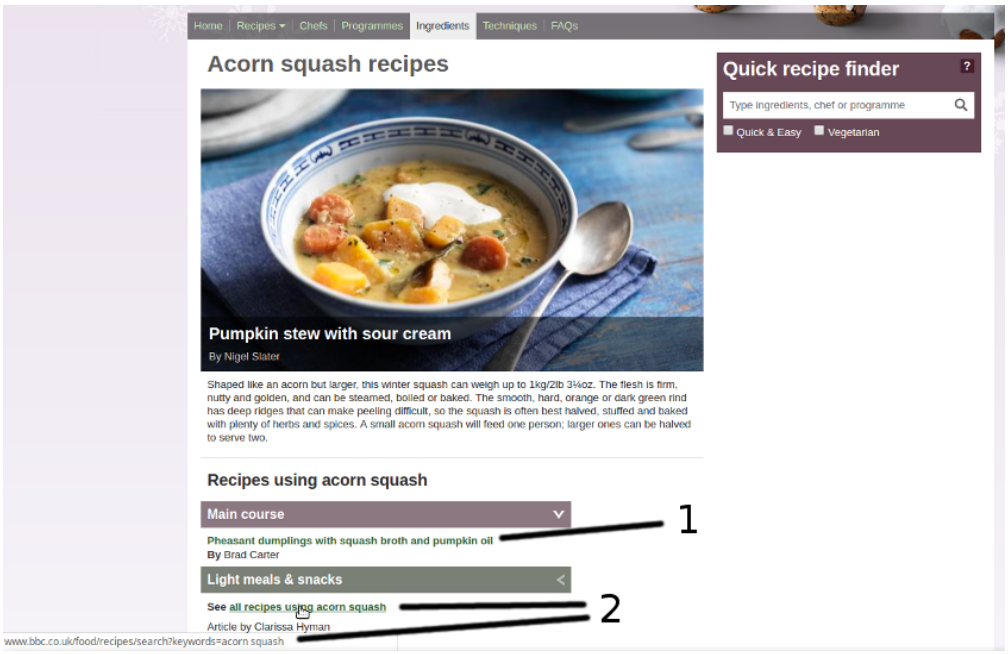

Figure 2.2: Sample ingredient webpage with: 1. Direct link to a recipe, 2. link to the search engine of the BBC webpage with a sample query.

2.1.2 Built in search engine

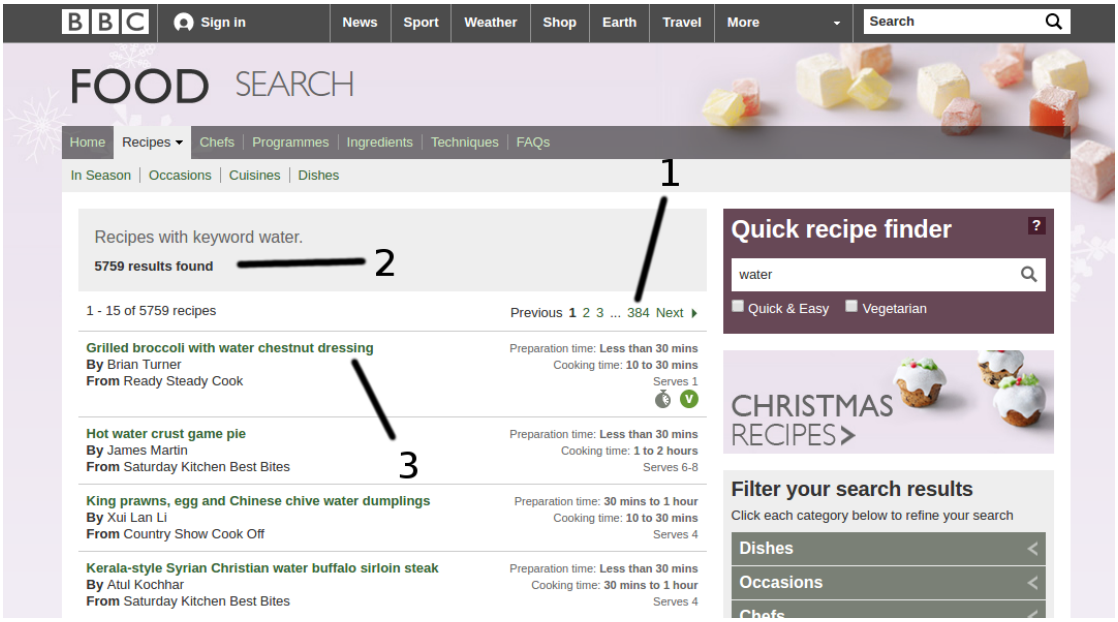

Figure 2.3: Example of the query *water* under the webpage search functionality with: 1. Maximum number of pages, 2. Total number of recipes under this query, 3. Example of one recipe link

Then the program explored each search webpage until found all the recipes, which by the time when we ran it were 11 224 recipes.
Also an other feature of our program is that it retries to get the webpage up to 3 times, with 5 seconds pause in between petitions. If still the information can not be retrieved, the link is saved to be explored the next time that the program is ran. We also made sure that the links were not explored twice and the links to the recipes were unique using sets to store them in memory before writing them to disk.

2.2 Analyzing the recipes

The python module of this part is stored in *search/download/analyzeRecipes.py*. In this part the information of each recipe is saved on the fly to a TSV file in disk that can be found in data/data.tsv. The extracted information of each recipe is the recipe name, author name, programme name, preparation time, cooking time, number of serves, URL of the picture, method of cooking, ingredients, vegetarian or not, caloric content (in kcal), grams of protein, carb, sugar, fat, saturated fat, fiber, salt. When the values where missing, the missing value was replaced by *NaN*. In order to do so we made use of the built in functionality of Python for managing errors *try*.

```
/home/sergio/.anaconda3/bin/python /home/sergio/PycharmProjects/AMD-HW2/scripts/download.py
### 1 of 11232 #### Fetching -->  http://www.bbc.co.uk/food/recipes/10minutepizza_87314
### 2 of 11232 #### Fetching -->  http://www.bbc.co.uk/food/recipes/15_minute_pasta_33407
### 3 of 11232 #### Fetching -->  http://www.bbc.co.uk/food/recipes/3d_biscuits_29555
### 4 of 11232 #### Fetching -->  http://www.bbc.co.uk/food/recipes/3wayswithlemoncurd_67266
### 5 of 11232 #### Fetching -->  http://www.bbc.co.uk/food/recipes/_81487
### 6 of 11232 #### Fetching -->  http://www.bbc.co.uk/food/recipes/_banoffee_mille_feuille_37951
### 7 of 11232 #### Fetching -->  http://www.bbc.co.uk/food/recipes/_chicken_chasseur_with_19163
### 8 of 11232 #### Fetching -->  http://www.bbc.co.uk/food/recipes/_cold-smoked_salmon_with_16940
### 9 of 11232 #### Fetching -->  http://www.bbc.co.uk/food/recipes/_fillet_of_beef_82682
### 10 of 11232 #### Fetching -->  http://www.bbc.co.uk/food/recipes/_schichttorte_49934
### 11 of 11232 #### Fetching -->  http://www.bbc.co.uk/food/recipes/_venison_massaman_curry_82057
### 12 of 11232 #### Fetching -->  http://www.bbc.co.uk/food/recipes/a_medley_of_shellfish_75055
### 13 of 11232 #### Fetching -->  http://www.bbc.co.uk/food/recipes/a_simple_layered_salad_21760
```

Figure 2.4: Example of extracting the information of the recipes.

2.3 How to run the program to download the recipes

The script to execute is located in *scripts/download.py*. The working directory should be set to the root folder. If the script is ran by default, it will find more recipes and analize them, updating the file *data.tsv*. On the other hand, if we want to find all the recipes from scratch, we have to ran the script setting the variables that we will find inside to *reset* = *True*. Right after that, we should change the variables *reset* again to False to make use of the feature of continue exploring the links and the recipes from the interrupted point without erasing again the stored files.

3 Preprocessing

4 Search engine

5 Graphical interface