Data Mining the Content of Food Articles Using Web Crawling

Ramil Gupta*

Department of Computer Science and Engineering, Baba Farid College of Engineering and Technology, Bathinda, India

Abstract – With the growing internet, searching web is an important part. To retrieve the web pages automatically, web crawler is used. Web crawler feeds on a seed URL and visits all the subsequent URLs to gather information. The processed information is stored in JSON documents. To further find the relationships between web pages, association rule mining is used. The frequent items are found using Apriori algorithm. Association rules are formed using these frequent items. In this paper we proposed a crawler that crawls the recipe site. Then from the structured data of JSON file, association rules are predicted.

Keywords: Crawler, JSON, Association Rules, Apriori Algorithm

---------------------------------x---------------------------------

INTRODUCTION

With the aim of providing most comprehensive information, structuring of data is an important perspective. For structuring the data, data mining is an effective process (Estlick, et al., 2001). The data is scattered all across the web. In this paper, we have considered structuring the data for various recipes. The data about recipes from one of the famous recipe website has been identified by help of crawling. Further data mining has been applied to the data collected for extracting useful information. Association Rules are used to find interlink between the processes (Ahmed, et. al., 2006. Bonchi & Lucchese, 2006. Chi, et. al., 2006). The frequent item sets can be determined by use of Apriori algorithm (Elmasri & Shamkant, 2009)

Overview of Association rule mining

The association rules are presented in the form A -> B, which implies wherever there is an occurrence of A there is occurrence of B also. A is the item found in data and B is the item found in combination with A. The goal is to extract the important correlations among the items existing in the database. The important correlations are those which satisfy the criteria of a minimum confidence and support. Support is measured by calculating the probability of the occurrence of a particular item in a set of items. Confidence is the true occurrence of condition “if A occurs then B will also occur”. Association rule mining is accomplished with the help of certain rules or algorithms. One of the best algorithms for mining association rules is Apriori algorithm.

Overview of Apriori Algorithm

One of the most popular algorithms for association rule mining is Apriori algorithm. The searching technique used by it is breadth first search. A level wise search is done by the algorithm in which it uses n item sets to explore n-1 item sets. It iterates in several passes over the database (Charanjeer, 2013). In the first pass it searches for a large set of items, which is further used for discovering datasets in further passes. The functioning of the algorithm is based on the minimum support. All the frequent items above that minimum level of support are considered. The other constraint that can be added is of minimum defined confidence level.

Overview of Web crawler

A web crawler is a program that visits the web pages in a way humans do, with the objective of validating, analyzing and visualizing the web pages (Chakrabarti, et. Al., 1998) (Bharat and Henzinger, 1998). There are two steps in focused crawling process:

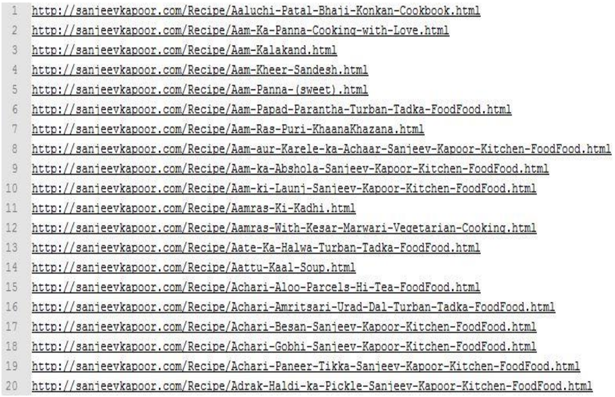
Identification of seed URL (i.e. starting URL). This is the primary and important step because without a starting URL the crawler cannot start. All the pages identified by the seed URL are retrieved. Then these pages are checked for further presence of URLs. The secondary step is to choose a technique for crawling. The URLs found in the pages of seed URL are queued for processing. It places the URLs in the queue based on their relevance. The queue is sorted on the basis of the relevance of URLs (Yuvarani, et. al., 2006).

Proposed Method

Our implementation works in following steps:

1. Choosing a seed URL
2. Fetch the list of URLs of recipes present.
3. Retrieval of desired information from html document and storing it in JSON document.
4. Applying Apriori to JSON data.

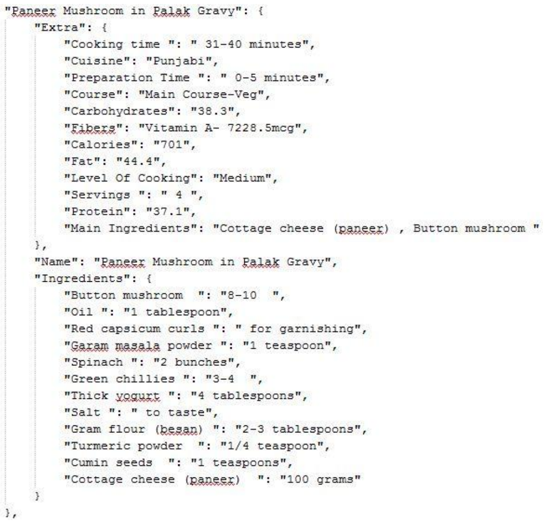
In our implementation we have chosen “sanjeevkapoor.com” as our seed URL. Then the pages of this URL are visited and a list of URLs of recipes is retrieved by hitting upon by Python and stored in a text file. For performing this activity various open source crawlers can be used. Here we have used Scrapy [5]. The list obtained is then normalized in order to remove duplicate URLs. The list is then sorted using set function and result is stored in a text file for easy retrieval.



```
http://sanjeevkapoor.com/Recipe/Aaluchi-Patal-Bhaji-Konkan-Cookbook.html
http://sanjeevkapoor.com/Recipe/Aam-Ka-Panna-Cooking-with-Love.html
http://sanjeevkapoor.com/Recipe/Aam-Kalakand.html
http://sanjeevkapoor.com/Recipe/Aam-Kheer-Sandesh.html
http://sanjeevkapoor.com/Recipe/Aam-Panna-(sweet).html
http://sanjeevkapoor.com/Recipe/Aam-Papad-Parantha-Turban-Tadka-FoodFood.html
http://sanjeevkapoor.com/Recipe/Aam-Ras-Puri-KhaanaKhazana.html
http://sanjeevkapoor.com/Recipe/Aam-aur-Karele-ka-Achaar-Sanjeev-Kapoor-Kitchen-FoodFood.html
http://sanjeevkapoor.com/Recipe/Aam-ka-Abshola-Sanjeev-Kapoor-Kitchen-FoodFood.html
http://sanjeevkapoor.com/Recipe/Aam-ki-Launj-Sanjeev-Kapoor-Kitchen-FoodFood.html
http://sanjeevkapoor.com/Recipe/Aamras-Ki-Kadhi.html
http://sanjeevkapoor.com/Recipe/Aamras-With-Kesar-Marwari-Vegetarian-Cooking.html
http://sanjeevkapoor.com/Recipe/Aate-Ka-Halwa-Turban-Tadka-FoodFood.html
http://sanjeevkapoor.com/Recipe/Aattu-Kaal-Soup.html
http://sanjeevkapoor.com/Recipe/Achari-Aloo-Parcels-Hi-Tea-FoodFood.html
http://sanjeevkapoor.com/Recipe/Achari-Amritsari-Urad-Dal-Turban-Tadka-FoodFood.html
http://sanjeevkapoor.com/Recipe/Achari-Besan-Sanjeev-Kapoor-Kitchen-FoodFood.html
http://sanjeevkapoor.com/Recipe/Achari-Gobhi-Sanjeev-Kapoor-Kitchen-FoodFood.html
http://sanjeevkapoor.com/Recipe/Achari-Paneer-Tikka-Sanjeev-Kapoor-Kitchen-FoodFood.html
http://sanjeevkapoor.com/Recipe/Adrak-Haldi-ka-Pickle-Sanjeev-Kapoor-Kitchen-FoodFood.html
```

Fig 1: List of URLs retrieved

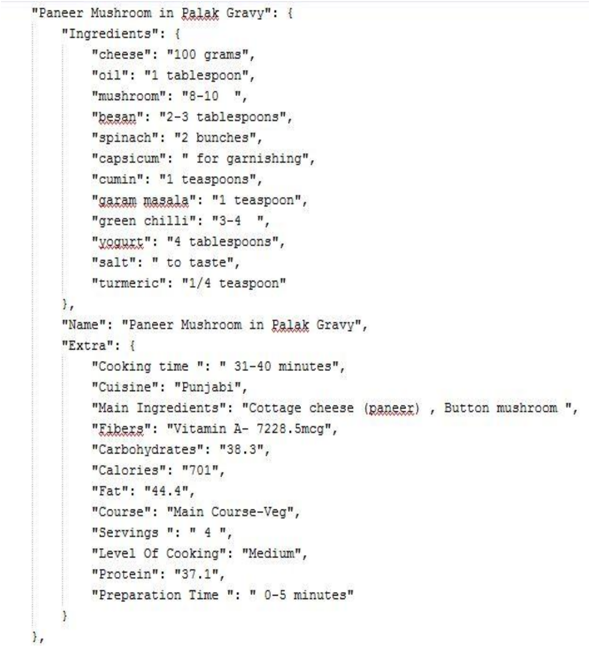
After the retrieval of the list of URLs, the crawler has been configured. Now the data needs to be extracted from different web pages and structure them as required.



```
"Paneer Mushroom in Palak Gravy": {
    "Extra": {
        "Cooking time ": " 31-40 minutes",
        "Cuisine": "Punjabi",
        "Preparation Time ": " 0-5 minutes",
        "Course": "Main Course-Veg",
        "Carbohydrates": "38.3",
        "Fibers": "Vitamin A- 7228.5mcg",
        "Calories": "701",
        "Fat": "44.4",
        "Level Of Cooking": "Medium",
        "Servings ": " 4 ",
        "Protein": "37.1",
        "Main Ingredients": "Cottage cheese (paneer) , Button mushroom "
    },
    "Name": "Paneer Mushroom in Palak Gravy",
    "Ingredients": {
        "Button mushroom  ": "8-10  ",
        "Oil ": "1 tablespoon",
        "Red capsicum curls ": " for garnishing",
        "Garam masala powder ": "1 teaspoon",
        "Spinach ": "2 bunches",
        "Green chillies ": "3-4  ",
        "Thick yogurt ": "4 tablespoons",
        "Salt ": " to taste",
        "Gram flour (besan) ": "2-3 tablespoons",
        "Turmeric powder  ": "1/4 teaspoon",
        "Cumin seeds  ": "1 teaspoons",
        "Cottage cheese (paneer)  ": "100 grams"
    }
},
```

Fig 2: Retrieved data

When the pages are hit using Scarpy the result is html tags. These tags may contain irrelevant data also as shown in figure 2. To find the relevant information the page has to be navigated. Data cleaning has been done with the help of beautiful soup which helps in finding regular expressions using Python. After finding the relevant information, it is stored in a structured format i.e.in JSON document.



```
"Paneer Mushroom in Palak Gravy": {
    "Ingredients": {
        "cheese": "100 grams",
        "oil": "1 tablespoon",
        "mushroom": "8-10  ",
        "besan": "2-3 tablespoons",
        "spinach": "2 bunches",
        "capsicum": " for garnishing",
        "cumin": "1 teaspoons",
        "garam masala": "1 teaspoon",
        "green chilli": "3-4  ",
        "yogurt": "4 tablespoons",
        "salt": " to taste",
        "turmeric": "1/4 teaspoon"
    },
    "Name": "Paneer Mushroom in Palak Gravy",
    "Extra": {
        "Cooking time ": " 31-40 minutes",
        "Cuisine": "Punjabi",
        "Main Ingredients": "Cottage cheese (paneer) , Button mushroom ",
        "Fibers": "Vitamin A- 7228.5mcg",
        "Carbohydrates": "38.3",
        "Calories": "701",
        "Fat": "44.4",
        "Course": "Main Course-Veg",
        "Servings ": " 4 ",
        "Level Of Cooking": "Medium",
        "Protein": "37.1",
        "Preparation Time ": " 0-5 minutes"
    }
},
```

Fig 3 : Processed JSON data

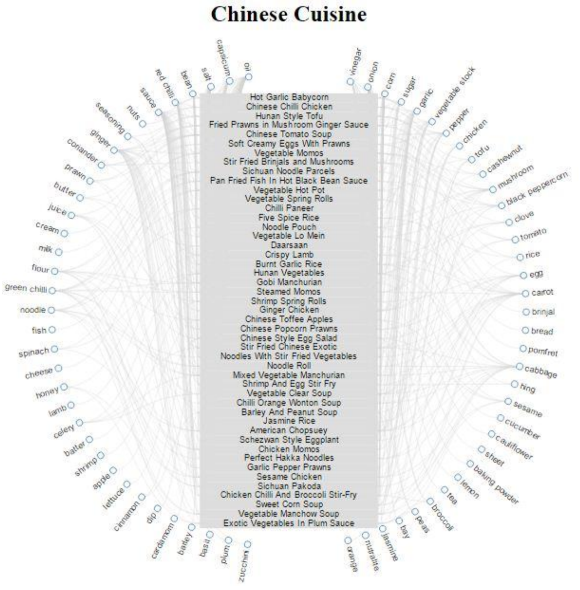


Fig 4: Graph depicting ingredients and list of recipies of Chinese cuisine.

From this processed Json data we have gathered the results in form of a graph, which depicts the list of ingredients used in a particular cuisine. We have selected Cuisine as the criteria for making the graphs. One of the example has been depicted above in figure 4. On taking the mouse cursor on a particular recipie name, all the ingredients used in that recipie are highlighted. This makes it easy to identify the prerequisites for making a particular recipe.

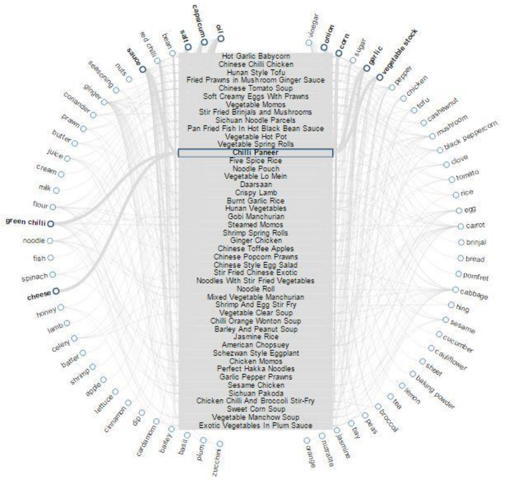
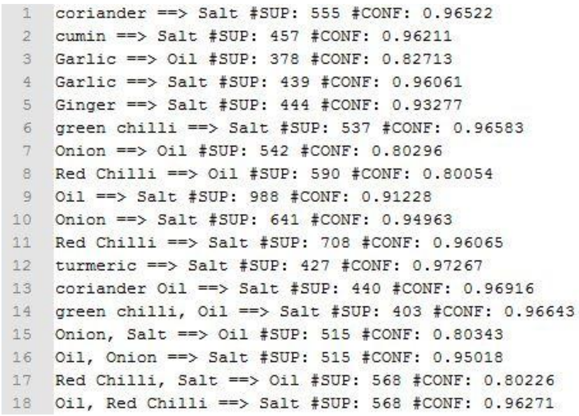


Fig 5 : Highlighted ingredients on selecting the recipe

Now on this processed data, Apriori algorithm is applied. For finding the associations between web pages on the basis of keywords, association rule mining is the best technique (Agrawal and Srikant, 1994). Apriori algorithm has been chosen to handle the web pages Apriori Algorithm for Web crawling (Estlick, et. al., 2001).

We have considered minimum support to be 20% and minimum confidence to be 0.80. If any item doesn't match the support criteria, the Apriori algorithm doesn't consider it for further evaluation, rather rejects it.

Support is the ratio of frequency of item to the total number of items. Confidence deals with conditional probability. It calculates the occurrence of B whenever A has ocurred (i.e. it calculates if –else possibilities). By applying the association rules we found that by marking Minimum support of 20% and minimum confidence of 80%, there are total of 18 association rules as shown in figure 4.



```
1   coriander ==> Salt #SUP: 555 #CONF: 0.96522
2   cumin ==> Salt #SUP: 457 #CONF: 0.96211
3   Garlic ==> Oil #SUP: 378 #CONF: 0.82713
4   Garlic ==> Salt #SUP: 439 #CONF: 0.96061
5   Ginger ==> Salt #SUP: 444 #CONF: 0.93277
6   green chilli ==> Salt #SUP: 537 #CONF: 0.96583
7   Onion ==> Oil #SUP: 542 #CONF: 0.80296
8   Red Chilli ==> Oil #SUP: 590 #CONF: 0.80054
9   Oil ==> Salt #SUP: 988 #CONF: 0.91228
10  Onion ==> Salt #SUP: 641 #CONF: 0.94963
11  Red Chilli ==> Salt #SUP: 708 #CONF: 0.96065
12  turmeric ==> Salt #SUP: 427 #CONF: 0.97267
13  coriander Oil ==> Salt #SUP: 440 #CONF: 0.96916
14  green chilli, Oil ==> Salt #SUP: 403 #CONF: 0.96643
15  Onion, Salt ==> Oil #SUP: 515 #CONF: 0.80343
16  Oil, Onion ==> Salt #SUP: 515 #CONF: 0.95018
17  Red Chilli, Salt ==> Oil #SUP: 568 #CONF: 0.80226
18  Oil, Red Chilli ==> Salt #SUP: 568 #CONF: 0.96271
```

Fig 4: Association rules retrieved

CONCLUSION AND FUTURE

In this paper we have implemented a crawler for recipe website. It detects the entry URL first and the processes the pages to find the subsequent URLs. It uses the sorted order to perform crawling. Then using Apriori algorithm, the association rules have been formed. These association rules help to give us the occurrence of two ingredients together. We can easily correlate how two ingredients are linked. In future, the crawler could be extended to other sites like blog sites, forum sites and social networking sites.

REFERENCES

[1] Ramez Elmasri, Shamkant B. Navathe (2009). "Fundamentals of Database Systems", PEARSON, fifth edition, 2009, 978-81-317-1625-0.

[2] Charanjeer Kaur (2013). Association Rule Mining Using Apriori Algorithm:A Survey Internationl Journal of advanced Research in Computer Enginerring & Technology (IJARCET) Volume2,Issue 6,June 2013

[3] S. Chakrabarti, B. Dom, P. Raghavan, S. Rajagopalan, D. Gibson, and J. Kleinberg

(1998). Automatic resource compilation by analyzing hyperlink structure and associated text," in Proc. 7th World Wide Web Conference, Brisbane, Australia.

[4] K. Bharat and M. Henzinger (1998). "Improved algorithms for topic distillation in hyperlinked environments," in Proceedings 21st Int'l ACM SIGIR Conference., 1998.

[5] Scrapy Web tool, http://scrapy.org/.

[6] Agrawal, R. and Srikant, R. (1994). Fast algorithms for mining association rules. In Proc. of the 20th VLDB Conference, pp. 487-499.

[7] M. Estlick, M. Leeser, J. Szymanski, and J. Theiler (2001). Algorithmic Transformations in the Implementation of K-means Clustering on Recon_gurable Hardware. In Proceedings of the Ninth Annual IEEE Sym-posium on Field Programmable Custom Computing Machines 2001 (FCCM '01), 2001.

[8] C. Wolinski, M. Gokhale, and K. McCabe (2004). A Recongurable Computing Fabric. In Proceedings of the Engineering of Recon_gurable Systems and Algorithms ERSA '02, 2004.

[9] Q. Zhang, R. D. Chamberlain, R. Indeck, B. M. West, and J. White (2004). Massively Parallel Data Mining using Recon_gurable Hardware: Approximate String Matching. In Proceedings of the 18th Annual IEEE International Parallel and Distributed Processing Symposium (IPDPS '04), 2004.

[10] Ahmed S., Coenen F., Leng P.H. (2006). Tree-based partitioning of date for association rule mining. Knowl. Inf. Syst. 10(3): pp. 315–331.

[11] Bonchi F., Lucchese C. (2006). On condensed representations of constrained frequent patterns. Knowl. Inf. Syst. 9(2): pp. 180–201

[12] Chi Y., Wang H., Yu P.S., Muntz R.R. (2006). Catch the moment: maintaining closed frequent itemsets over a data stream sliding window. Knowl. Inf. Syst. 10(3): pp. 265–294.

[13] M. Yuvarani, N. Ch. S. N. Iyengar and A. Kannan (2006). "L S Crawler: A Framework for an Enhanced Focused Web Crawler based on Link Semantics" in Proceedings of the IEEEIWIC/ACM International Conference on Web Intelligence, 2006.

Corresponding Author

Ramil Gupta*

Department of Computer Science and Engineering, Baba Farid College of Engineering and Technology, Bathinda, India

E-Mail – ramilgupta.bfcet@gmail.com