

RECIPE SUGGESTION TOOL

A Writing Project

Presented to

The Faculty of the Department of Computer Science

San José State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Computer Science

by

Sakuntala Gangaraju

May 2010

ABSTRACT

RECIPE SUGGESTION TOOL

by Sakuntala Gangaraju

There is currently a great need for a tool to search cooking recipes based on ingredients. Current search engines do not provide this feature. Most of the recipe search results in current websites are not efficiently clustered based on relevance or categories resulting in a user getting lost in the huge search results presented. They also do not provide links to view images of the ingredients of a recipe.

My project aims to combine the features like search based on ingredients, suggestions for similar recipes, and images for the ingredients under one search engine and provide an intuitive interface for the same. I explored different clustering algorithms to find an efficient algorithm that can be used to cluster recipe data matching user's queries. As part of this project, I also built custom search engine wrappers around existing search engines to help users search images of ingredients.

Table of Contents

1. Introduction	4
2. Deliverable 1: Report on Dr. Pollett’s Search Engine	5
3. Deliverable 2: Implementation of k-means clustering.....	7
4. Deliverable 3: Implementation of Clustering using Kruskal’s algorithm	9
5. Deliverable 4: Google custom search engine Wrapper	11
6. Conclusion	13
7. References	14

1. Introduction

Search engines have made access to information easy. One only needs to get connected to the internet to get the information one needs. When searching for cooking recipes, sometimes user may prefer to search based on ingredients. It will be more helpful if the search also suggests similar recipes. The user sometimes may not know what an ingredient is just from its name. So images of the ingredients displayed beside the written name of the ingredient will be helpful to the user.

In this project, I am working on a recipe suggestion tool, which suggests similar recipes. Users can also search based on ingredients. I am also providing the facility for the user to view the images of the ingredients. I am also planning to cluster data based on the relevance which is not provided by most of the recipe websites.

In CS297 project, I categorized my work into four deliverables: (1) Research on Search Engine, (2) Implement K-means Clustering algorithm, (3) Implement clustering through minimum spanning tree and (4) Implement custom search engine wrapper for image based search.

Deliverable 1 includes a report on Dr. Pollett's Search Engine. Deliverable 2 includes implementation of the partition based clustering algorithm, K-means. Deliverable 3 includes implementation of clustering based on the minimum spanning tree. Kruskal's algorithm is used to construct the minimum spanning tree from the input data. Deliverable 4 includes implementation of Google custom search engine wrapper for image based search.

The report is mainly divided into seven sections: Introduction, Deliverable 1 to Deliverable 4, Conclusion and References. The Introduction mainly addresses the purpose, the plan followed and the scope of this report. It also describes the project motivation and gives an overview of the project. Deliverable 1 to Deliverable 4 explain the objective of each deliverable, the work done and concludes how it is helpful in my project. The Conclusion summarizes all the work done in the project and the work to be done in Fall 2010.

2. Deliverable 1: Report on Dr. Pollett's Search Engine

The goal of this project is to suggest recipes to the user. For this, a database of recipes needs to be maintained. A web search engine can be used to retrieve the required information from the web. The study of a search engine would be helpful in understanding how it works and how the data can be retrieved and maintained in the database.

I studied Dr. Pollett's search engine, the crawler and the indexer. The crawler retrieves the pages from the web. They are then indexed in a database. Search engines also measure the relevance of the result set it displays. It ranks the results to provide the best results first.

To run the search engine, Apache, PHP and Mysql need to be installed. The database to be maintained needs to be built first. Seed sites are to be provided to start the crawling. Crawling is done to populate the database with websites. Page rank algorithm is used to calculate the page ranks.

The crawler begins by creating the required tables in the database. A list of websites to be crawled is maintained. This is initially populated with the seed sites to start a new crawl. The crawler then grabs a crawl page from the list to be crawled and checks for the robots.txt file. The Robots.txt file prevents the crawler from accessing all or parts of a web site. If no robots.txt file exists, it proceeds to process for summary data. It does this by deleting the selected web page from the list of web pages to be crawled. It then extracts the title, description and links from the web page. The crawler then updates the list to be crawled with links extracted. The summary of the page is then stored in the database. The process continues until either a certain number of URLs or a certain depth is reached. An adjacency matrix is maintained, which represents all pages that link to a given page.

Once crawling is done, the page rank algorithm is implemented to calculate the page ranks of the pages. The adjacency matrix is converted into a stochastic matrix to preserve probabilities. An initial vector of all 1's is chosen. The vector is multiplied with the stochastic matrix until it converges. Some adjustments to the matrix are to be made to handle dangling nodes and strongly connected components. Dangling nodes are the nodes with no outgoing

links. These are handled by assuming that they are connected to every other node with some probability. Strongly connected components are parts of the web graph, which do not have any outgoing links to the rest of the graphs. A random surfer adjustment is made to handle strongly connected components.

For my project, the database should store information about the recipes only. So, a focused crawling or a topical crawling should be implemented. A focused crawler attempts to download the pages relevant to a pre-defined topic.

3. Deliverable 2: Implementation of K-means clustering

Most of the recipe search results shown by current websites are not efficiently clustered based on relevance or categories, resulting in a user getting lost in the huge search results presented. The main objective of Deliverable 2 and Deliverable 3 is to learn clustering of data.

Clustering, also known as Cluster analysis involves grouping a set of observations into clusters so that similar observations fall into the same cluster. It is a common technique used for statistical data analysis. The similarity of two observations is calculated based on distance measure. Common distance functions include Euclidean distance and Hamming distance. Clustering algorithms can be Hierarchical clustering or Partitional clustering.

The objective of this deliverable is to implement K-means clustering. It is one of the partitional clustering algorithms. The number of clusters to produce in the input data is to be specified prior to the execution.

The K-means algorithm assigns each of the input data to the cluster whose center is the nearest. The center of a cluster is the average of all points belonging to the cluster.

The algorithm steps include:

1. Choose the number of clusters, k .
2. Randomly generate k clusters and determine the cluster centers, or directly generate k random points as cluster centers.
3. Assign each point to the nearest cluster center.
4. Recompute the new cluster centers.
5. Repeat the two previous steps until assignment hasn't changed (convergence criterion).

The implementation was done in PHP. The number of clusters to be formed (K) and the data to be clustered is the input. The input data was represented as a point in Euclidean space. K random points were selected as the centroids initially. The Euclidean distance to each of the centroids from the data points was calculated. The data point was assigned to the cluster to which the distance to the center is the least. The centroids were recalculated from the points

assigned to the cluster. The points were reassigned to the clusters. This process was repeated until the new assignment was same as the previous assignment. The output displayed the input data, centroids of each cluster and the clusters. Figure 1 below shows a sample output for K=3.

```
Data :  
Recipe 1 : (1,0)  
Recipe 2 : (2,0)  
Recipe 3 : (10,0)  
Recipe 4 : (11,0)  
Recipe 5 : (15,1)  
Recipe 6 : (20,5)  
  
Centroids :  
Centroid 1 : (10.5,0)  
Centroid 2 : (1.5,0)  
Centroid 3 : (17.5,3)  
  
Clusters :  
Cluster 1: {Recipe 3(10,0), Recipe 4(11,0)}  
Cluster 2: {Recipe 1(1,0), Recipe 2(2,0)}  
Cluster 3: {Recipe 5(15,1), Recipe 6(20,5)}
```

Figure 1: Sample output for K=3

In this implementation of K-means, the first set of centroids was assigned randomly. This random assignment results in different outputs for the same input.

4. Deliverable 3: Implementation of Clustering using Kruskal's algorithm

The main objective of this deliverable was to implement Hierarchical clustering.

Hierarchical clustering algorithms find the clusters using the previously established clusters. The algorithms can be agglomerative or divisive. The agglomerative start with each node as one cluster and merges the clusters successively where as the divisive starts at the root and splits the clusters recursively.

The data is represented as the undirected weighted graph. The implementation is done in two parts.

- I. Kruskal's algorithm is implemented to find the minimum spanning tree of the graph. It finds the subset of edges which includes all the vertices and the total weight of the edges are minimized. The algorithm includes steps:
 1. Create a forest F (a set of trees), where each vertex in the graph is a separate tree.
 2. Create a set S containing all the edges in the graph.
 3. while S is nonempty and F is not yet spanning
 - i. remove an edge with minimum weight from S
 - ii. if that edge connects two different trees, then add it to the forest, combining two trees into a single tree
 - iii. Otherwise discard that edge.
- II. Clustering is done by deleting the most expensive edge from the minimum spanning tree.

Implementation was done in PHP. The edges of the graph with weights and the number of clusters to be formed were given as the input. Implementation includes creating the minimum spanning tree using Kruskal's algorithm. Each vertex was considered as a separate tree. Each of the minimum edges was added to the minimum spanning tree if it did not create a cycle. The algorithm was implemented until all the vertices were covered. The minimum

spanning tree obtained by applying Kruskal's algorithm was used to create the required number of clusters. Clusters were formed by removing the most expensive edge from the minimum spanning tree. The number of edges to be removed was one less than the required number of clusters. The output displayed the input tree, the minimum spanning tree edges and the clusters of vertices. Figure 2 below shows a sample output for creating 3 clusters.

```
Data:
AB=>14
BC=>12
BD=>28
CD=>19
DE=>17
AF=>11
FC=>19
AE=>10
AD=>30
BE=>30

Minimum Edges:
AE=>10
AF=>11
BC=>12
AB=>14
DE=>17

Clusters:
Cluster 1={D}
Cluster 2={E,A,F}
Cluster 3={B,C}
```

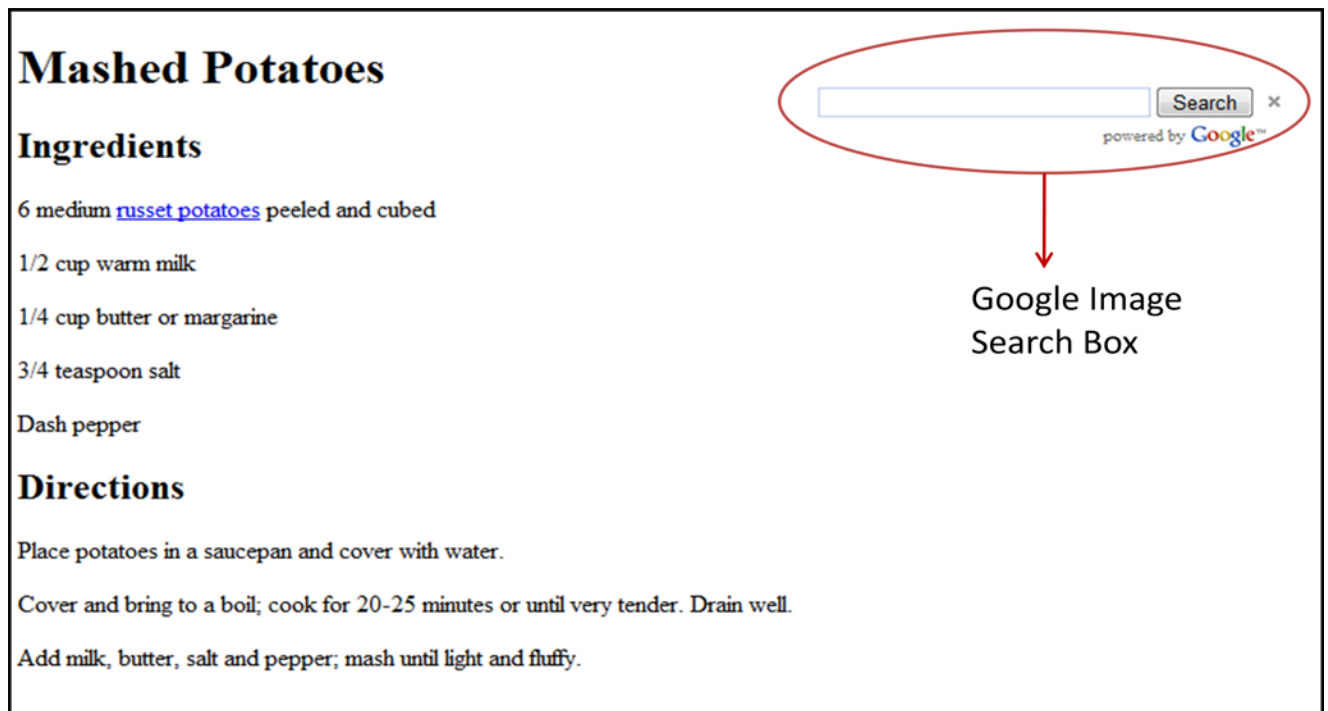
Figure 2: Sample output for creating 3 clusters using Kruskal's algorithm

In this implementation of clustering based on the minimum spanning tree, output is consistent unlike K-means. Clustering based on the minimum spanning tree helps to categorize recipe results and also presents users with a percentage of relevance to the original query making it easier for users to select recipes.

5. Deliverable 4: Google custom search engine Wrapper

Sometimes an image of the ingredient provides more information rather than its name. If the image is accessible within the same page it will be more helpful to the end-user. The user need not search for the images on multiple pages. Google AJAX Search APIs allow for Google image search on web pages with JavaScript. A dynamic search box can be embedded into the web pages and can display the results in the same web page.

The main objective of this deliverable was to put an Image Search box on a web page and search for images dynamically. Figure 3 below shows a web page consisting of a description of the recipe, “Mashed Potatoes”. The ingredients are listed, followed by directions. An image search box is provided on the right side of the page. The results of the search are displayed right below the box. One ingredient of the recipe, russet potatoes is hyperlinked.



The screenshot shows a recipe page for "Mashed Potatoes". The page is divided into sections: "Mashed Potatoes", "Ingredients", and "Directions". The ingredients list includes "6 medium russet potatoes peeled and cubed", "1/2 cup warm milk", "1/4 cup butter or margarine", "3/4 teaspoon salt", and "Dash pepper". The directions section contains three paragraphs: "Place potatoes in a saucepan and cover with water.", "Cover and bring to a boil; cook for 20-25 minutes or until very tender. Drain well.", and "Add milk, butter, salt and pepper; mash until light and fluffy." On the right side of the page, there is a Google Image Search box. The search box is a white input field with a "Search" button and a close "x" icon. Below the search box, it says "powered by Google™". A red oval highlights the search box, and a red arrow points from the oval to the text "Google Image Search Box" below it.

Figure 3: Recipe Description Page

When the user clicks on the hyperlinked ingredient, images of the ingredient are displayed below the search box with the ingredient as the search phrase. Figure 4 below shows the case when the user clicks on the russet potatoes. Images of potatoes are shown below the search box.

Mashed Potatoes

Ingredients


6 medium [russet potatoes](#) peeled and cubed
1/2 cup warm milk
1/4 cup butter or margarine
3/4 teaspoon salt
Dash pepper

Directions


Place potatoes in a saucepan and cover with water.
Cover and bring to a boil; cook for 20-25 minutes or until very tender. Drain well.
Add milk, butter, salt and pepper; mash until light and fluffy.

×
powered by Google™

Image



russet potatoes
700 x 466
www.faqs.org



russet potatoes
100 x 67
www.faqs.org

Figure 4: Image results after the ingredient is clicked

To use the search API, the URL for Google AJAX APIs loader (<http://www.google.com/jsapi>) should be included in the webpage. The main object used by Google AJAX Search API is an instance of SearchControl. ImageSearch is one of their child objects which provides the image search service. SearchControl's execute method takes in search terms as argument and displays the results.

In this deliverable, the hyperlinked ingredient, russet potatoes, was passed as the argument to the execute method, which displays the images of the russet potatoes.

Google AJAX Search APIs provide other search services like local search, web search, video search, etc. It also provides the facility to restrict the sites from which the data needs to be retrieved.

6. Conclusion

With information increasing day by day, the need for a domain based search engine is increasing. My project provides a recipe based search engine. I am planning to include new features like search based on ingredients, suggestion of similar recipes, and images for the ingredients. I am also planning to provide a clustering based on the relevance, which is not provided by most of the current recipe websites.

Studying Dr.Pollett's search engine helped me in understanding the crawling process. The crawler needs to do a focused crawling to download only the recipes web pages. This way, the recipes database can be built.

Clustering involves grouping of data such that similar data fall into similar clusters. The clustering algorithms are mainly two types – Partitional and Hierarchical. Implementation of K-means clustering and clustering based on the minimum spanning tree mainly helped me in understanding the clustering process. From my experiments, I understand that clustering based on the minimum spanning tree gives consistent results.

Google AJAX APIs help in providing a custom search on the same page. Image based search for the ingredients can be done using these APIs. The image results can be restricted based on the sites so that only relevant data is displayed.

In Fall 2010, I will extend this project by combining the features experimented. I will build the recipes database. I will use a focused crawler to download only web pages on recipes. Once the data is collected, I will do clustering of data based on the minimum spanning tree. The clustering can be used to provide features like suggestion of similar recipes, search based on ingredients, search filters, and suggestion of recipes based on user's previous searches. I will use the custom search engine wrappers to include image based and location based searches. This feature helps the user to look for images of the ingredients or look for grocery stores carrying the ingredients listed based on the location of the user. I will also work on the web based interface, which provides all these features in one place.

7. References

1. K-means clustering retrieved from http://en.wikipedia.org/wiki/K-means_clustering
2. Kruskal's algorithm for minimum spanning tree retrieved from http://en.wikipedia.org/wiki/Kruskal's_algorithm
3. Berkhin, P. A Survey of Clustering Data Mining Techniques.
4. Xu, R., Wunsch II D., Survey of Clustering Algorithms. *IEEE TRANSACTIONS ON NEURAL NETWORKS*, 16(3), MAY 2005.
5. Google Ajax APIs retrieved from <http://code.google.com/apis/ajax/>