Obtaining Structured Recipe From an Unstructured Corpus

Suryank Tiwari[1] **Rose Verma**[2] **Prateek Agarwal**[3]
Indraprashtra Institute of Information Technology, Delhi
1. suryank19019@iiitd.ac.
2. rose19052@iiitd.ac.
3. prateek19070@iiitd.ac.

Abstract

This project deals with an unstructured corpus of recipe data and derives a structure out of it. The structure thence obtained is used to construct a knowledge graph that can be queried to arrive at sub-graphs of nodes and edges. Inferences from the Knowledge Graph like community relations, centrality and connectedness are obtained and reviewed. We take up the task of named entity recognition to address this problem statement and then move on to construct a knowledge graph and derive inferences from it. On top of the knowledge graph created from recipe data, an information retrieval and displaying system is created. The work that follows can serve as a basis to cuisine classification, recommendation systems, knowledge graph based semantic derivation and so on.

1 Problem Definition

Numerous cooking websites present food recipes with varied cuisines and methodologies. Even though a plethora of recipe data exists, there is little to no structure to it. Any structures if present, are not really competent for deriving useful patterns from them. Therefore, our task is to derive some structure and meaningful relationships from this data and to be able to run an information retrieval system on it.

The motivation behind this project is to extract semantics and meaningful information from a given unstructured recipe corpus and to be able to query it in a manner that the semantics aren't lost.

2 Background

Information extraction and analysis has been a constant struggle pertaining to unstructured data. While no data is truly unstructured, extracting information from general instructions that are unformatted is a difficult task.

For arriving at intelligent structures like Knowledge Graphs and Recommendation Systems, a system that translates an unstructured corpus into a proper format is needed.

Search engines and querying systems generally fetch to a list of documents that best describe the queried terms. At the end of the query, we get unstructured documents as output which need to be processed further. Knowledge graph contain powerful semantic relationships and not just key word pairs. An information retrieval system on a Knowledge Graph would lead to smarter results that still hold their semantic properties.

3 Data sets

Epicurious data set:

https://www.kaggle.com/hugodarwood/epirecipes

This dataset contains approximately 20000 recipes out of which we identified top 100 recipes for the knowledge graph creation and top 30 recipes for derivation of inferences. These limitations are in place because of lack of processing power and resources.

4 Methodology

There are three major modules to this project. First one is to derive a structure from an unstructured corpus. Second one is the creation of a Knowledge Graph using the arrived structure and extracting inferences from it. The last module is the querying system to be created on the Knowledge Graph made.

4.1 Structuring the data

Phase one begins with the structuring process of the data. Extracting classified entities out of unstructured text via Named Entity Recognition is

the first hurdle. The results obtained from NER need to be clubbed with POS tagging on the same set of sentences to obtain the proposed output. The NER model needs to be trained on custom input to recognize utensils, ingredients etc. The structured output obtained is further used to arrive at tools from which patterns can be derived.
The model that is used for this phase is the NY Times CRF Ingredient Phrase Tagger. [4]
This model is implemented using the concept of named entity recognition. It takes ingredient data as input, trains the model and generates the model file. This model file then serves as a base to tag the ingredients in the test file to generate a CRF++ format data which is then converted to JSON data.

An example of the structuring process is depicted below:

Unstructured Recipe Sample:

Title: Mahi-Mahi in Tomato Olive Sauce
Ingredients:
2 teaspoons extra-virgin olive oil
1 cup chopped onion
1 cup dry white wine
1 teaspoon anchovy paste
4 6-ounce mahi-mahi fillets
1/2 cup large green olives, quartered, pitted
3 teaspoons chopped fresh oregano, divided
1 teaspoon (packed) finely grated orange peel

For each step in this unstructured corpus, a structured JSON element is created. The structure for each step is added with the recipe tag and an output JSON is created.

Following is a partial input which is converted to a sample output.

Partial Input: "2 teaspoons extra-virgin olive oil"

Proposed Output:

{
'title': 'Mahi-Mahi in Tomato Olive Sauce',
'ingredients': [{
'qty': '2',
'unit': 'tablespoon',
'comment': 'extra-virgin',
'name': 'olive oil',
'input': '2 tablespoons extra-virgin olive oil'
},
{
'qty': '1',
'unit': 'cup',
'comment': 'chopped',
'name': 'onion',
'input': '1 cup chopped onion'
}]
}

4.2 Structuring the data

From this structured data, the process of Knowledge Graph creation can begin. Knowledge Graph is created from a set of triples where every triple of the format ['source', 'target', 'edge']. Here, 'source' and 'target' are nodes which are connected by the edge 'edge', and will be found as such in the Knowledge Graph formed.

A collection of triples is created for each (recipe, structured cooking step) pair. For each pair, the 'source' node is the 'title', the 'target' node is the ingredient 'name' and the edge is the field 'comment' from the structured data obtained.

For the above mentioned structured data, the following Knowledge Graph is obtained.

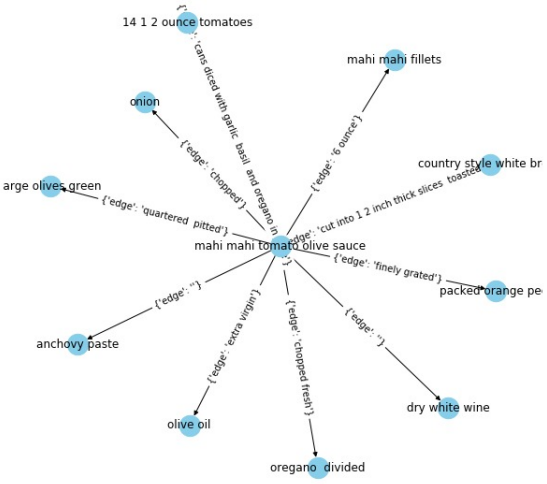

Figure 1: Knowledge Graph of single recipe

From the knowledge graph obtained inferences are dug out, such as:

- Connected components and communities
- The communities hint at a possible classification of the recipes that can be performed. These clusters have recipes of the same kind.

- Most Extensive recipe
- Most popular ingredient
- Rare and important ingredients

Inferences that are obtained are elaborated in further sections.

4.3 Querying the Knowledge Graph

Every triple that has been fed into the graph has been pre-processed with the following:

- Lowercase Conversion
- Punctuation Removal
- Lemmatization
- Stopword Removal
- Leading and trailing space removal

An inverted index is created for the query system in which every keyword encountered points to the nodes it belongs to.

Example:

'brown': {**'brown lentil'**}
'lentil': {**'lentil apple turkey wrap', 'french green lentil', 'brown lentil'**}

This inverted index is used to survey through candidate nodes and edges for each query term. Each query term also goes through the same pre-processing steps. For each candidate of a query term, the candidate can either be a node or an edge. If the candidate is a node, then the subgraph of all the neighbours of that node and the node itself are displayed. If the candidate is an edge, then all the edges with that term and the nodes the edges connect are shown as a subgraph.

Multiple term queries are allowed. If the query terms are a mixture of nodes and edges, all the subgraphs will be included in the answer, connected to each other if there are connections present due to query terms, otherwise as disconnected components.

5 Literature Review

Rahul Agarwal and Kevin Richard Miller [1] performed a similar task of converting unstructured recipe data into a machine readable format. They perfomed this task by making use of Named Entity Recognition (NER), Maximum Entropy Markov Model (MEMM) and Semantic Role Labelling (SRL).

Erica Greene [2] presented her work on the NYT Ingredient Tagger which performs CRF on unstructured ingredient data to a high degree of success. Nuno Silva, et al.[3] performed the task of structuring the recipes using the NYT Ingredient Tagger and were successful in obtaining the desired results. Therefore, this model is also chosen as a baseline model over which the task of building knowledge graphs is to be done.

FoodKG is a Semantics-Driven Knowledge Graph for Food Recommendation.[6] It is constructed over 1 million recipes and computes the optimal recipes that can be made given a set of ingredients while handling constraints such as allergies into account. It is also equipped with a cognitive agent that performs natural language question answering on the knowledge graph.

6 Results

The unstructured recipe ingredients for each recipe given as an input resulted into a structured ingredients. **The accuracy of the system is calculated on word basis as well as sentence basis**. Word level accuracy implies correct number of words identified out of total words whereas sentence level accuracy implies the total number of sentences in which all the words were correctly identified out of total sentences.

We were able to achieve 74% accuracy at sentence level whereas 90% accuracy at word level

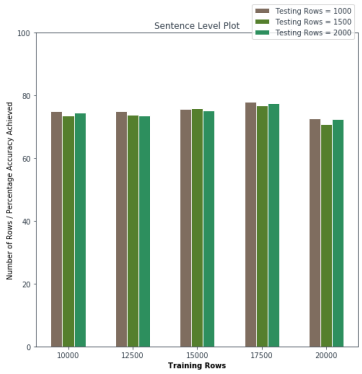

Figure 2: Sentence plot of structured data

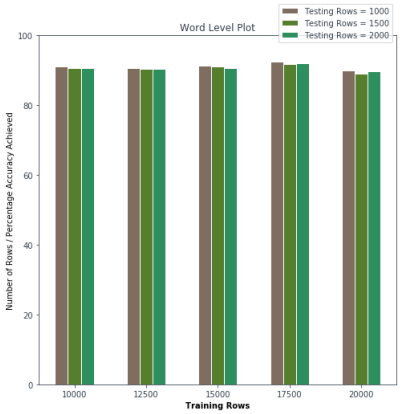

Figure 3: Word plot of structured data

Knowledge Graphs

Following is the knowledge graph on 100 recipe data:

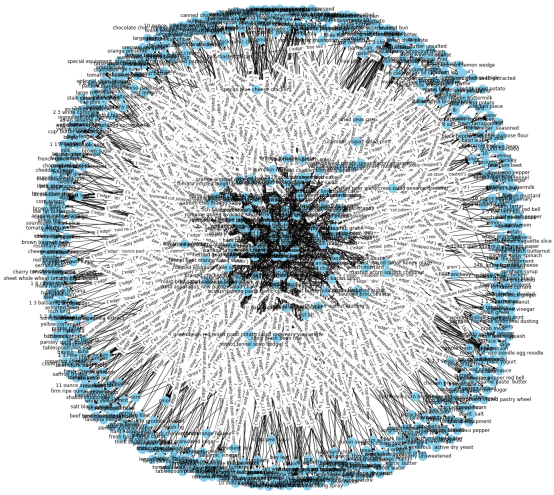

Figure 4: Knowledge Graph of 100 recipes

The graph obtained is dense and has a total 105 nodes and 113 edges. Zooming the graph gives us a better perspective.

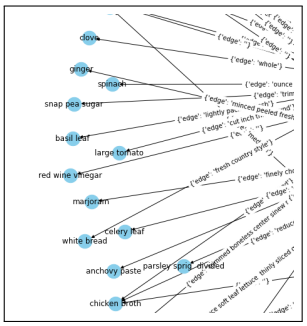

Figure 5: Zoomed Knowledge Graph: 100 recipes

Querying the graph leads to simpler subgraphs that can be understood clearly.

Simple Query: **'potato'**

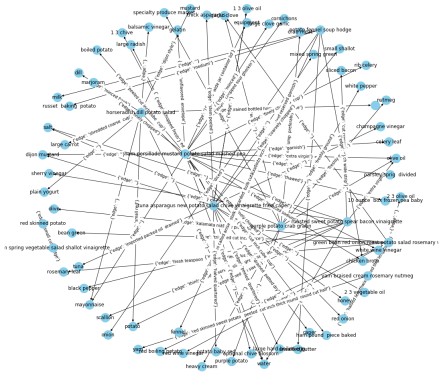

Figure 6: Sub-Knowledge Graph of Potato on 100 recipes

We can see there are multiple central nodes, and the graph as a lot of nodes still. This shows that potato is a widely used ingredient.

Simple Query: **'cardamom'**

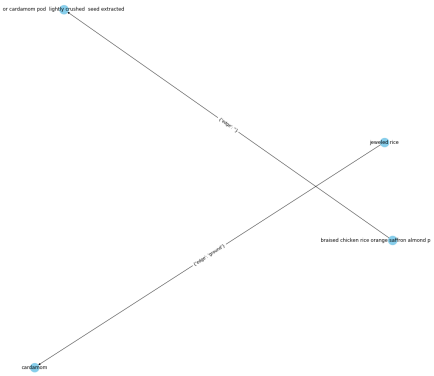

Figure 7: Sub-Knowledge Graph of Cardamom on 100 recipes

The KG for cardamom has only four nodes, and the graph returned is disconnected. This shows that cardamom is a rarely used ingredient. If any connection between the current set of nodes present could have been unveiled by the query, the resultant subgraph would have been connected.

The retrieval system also accounts edges while querying and this can be seen with this example.

Simple Query: **'canned'**

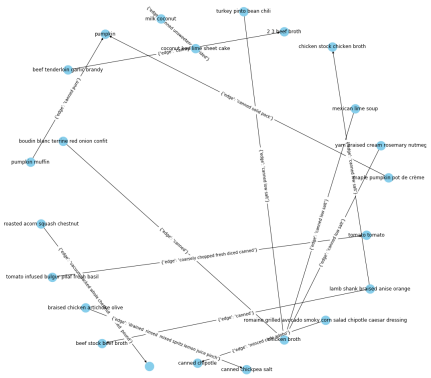

Figure 8: Sub-Knowledge Graph of Canned on 100 recipes

Complex queries containing terms from both edges and nodes present connected graphs if any connections have been unveiled through the query terms.

Complex Query: **'diced spinach'**

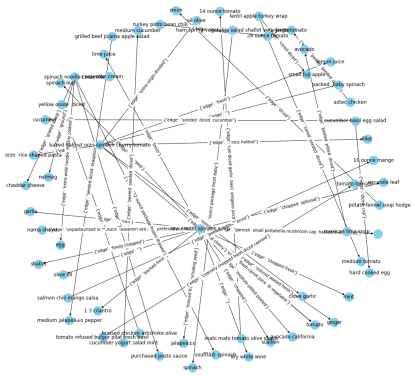

Figure 9: Sub-Knowledge Graph of Diced Spinach on 100 recipes

Inferences such as communities within a knowledge graph:

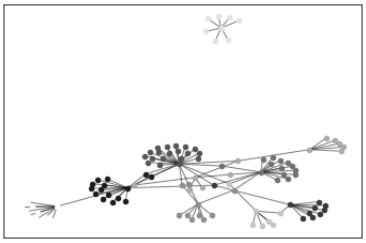

Figure 10: Communities within a knowledge graph

7 Inferences

The constructed Knowledge Graph is used to draw inferences to find out important information. Graphs are widely used for Analytics and reveal relevant patterns in the data.

The following properties of the Graphs were mainly used to draw inferences:

7.1 Centrality:

- **Degree Centrality:** Degree Centrality is a measure of popularity. It determines the nodes that are connected to the most number of nodes. This measure determines the nodes in the graph that can quickly spread information in a localised region.

  For the context of recipe data, an ingredient node with high degree centrality is an ingredient that is popular and is used in a lot of recipes or with a lot of other ingredients.

  For the dataset of 100 recipes, the most popular ingredients are: Salt, Olive oil, Sugar, Egg, Onion and Garlic Clove.

  For a recipe node with a high degree centrality, it indicates that this recipe uses a lot of ingredients or that it has a lot of common ingredients with other recipes.

  For our dataset, these were some of the most extensive, ingredient heavy recipes: boudin blanc terrine red onion confit, aztec chicken, braised brisket bourbon peach glaze, and coconut key lime sheet cake.

- **Betweenness Centrality:** Betweenness Centrality is a measure that provides a sense about which nodes are important not because they have a large number of connections but because they provide connectivity and cohesion to the network.

  With this measure, the most important ingredients in the recipe corpus are identified: Water, Black Pepper, and Lemon Juice. These nodes are the ones that are important ingredients of a recipe and they have high betweenness centrality but low degree centrality.

  The most important recipes in the corpus are also identified: ham persillade mustard potato salad mashed pea, braised chicken artichoke

olive, braised brisket bourbon peach glaze, spicy noodle soup.

These nodes serve as articulation points in the knowledge graph.

- **Eigen Vector Centrality:** Eigen Vector Centrality is useful for understanding which nodes can get information to a large number of nodes quickly. It can also be seen as related influence, or the nodes that play the role of power behind the scenes. Essentially, the nodes that are neighbours of the important nodes in the knowledge graph.

  Such ingredients, that have influence beyond popularity, are Salt, Sugar, Egg, Olive Oil, and Vanilla Extract. These ingredients can be seen as the most essential ingredients.

7.2 Connected Components of the Graph

The connected components of the graph hint at the different genres of recipes that can be present inside the knowledge graph.

The connected components hint at the recipes that are of the same kind. It is seen that a majority of recipes form the largest connected component of the graph and the rest contain one or two recipes. This hints that some recipes are very unique and different due to their ingredients or methods of preparation. These recipes are like the outliers in the sample space as their method of preparation and/or the ingredient set is very different than the most recipes.

Examples of such unique recipes: honey rye and fried chicken

7.3 Community detection

Community detection is an approach by which the knowledge graph is partitioned in the best possible way. It is seen that the recipes form groupings together. These groupings are called communities and hint at the segregation of recipes based on the ingredients or the method of preparation.

It was seen that in a knowledge graph constructed from a corpus of 100 recipes with 2 connected components had 9 communities within.

The communities hint at a possible classification of the recipes that can be performed. These clusters have recipes of the same kind.

8 Conclusion and Future Work

Structural recipes from an unstuctured recipe corpus were successfully created. This structure was used to create, infer and query knowledge graphs. The size of the graphs was limited to 500 nodes or so given the lack of proper resources for processing.

Future work may include containing several copies of architecturally different Knowledge graphs with different rules for triple mapping. A combination of different structures might be able to better represent complex relationships.

References

[1] Agarwal, Rahul and Kevin Richard Miller. 'Information Extraction from Recipes.' (2011).

[2] Erica Greene. 'NY times Ingredient Tagger'.

[3] Nuno Silva, David Ribeiro and Liliana Ferreira. 'Information Extraction from Unstructured Recipe Data.' (2019).

[4] https://github.com/nytimes/ingredient-phrase-tagger

[5] http://open.blogs.nytimes.com/2015/04/09/extracting-structured-data-from-recipes-using-conditional-random-fields/?_r=0

[6] Haussmann, Steven & Seneviratne, Oshani & Chen, Yu & Ne'eman, Yarden & Codella, James & Chen, Ching-Hua & Mcguinness, Deborah & Zaki, Mohammed. (2019). FoodKG: A Semantics-Driven Knowledge Graph for Food Recommendation. 10.1007/978-3-030-30796-7_10.