A Named Entity Based Approach to Model Recipes

1st Nirav Diwan
Dept. of Computer Science
IIIT-Delhi
New Delhi, India
nirav17072@iiitd.ac.in

1st Devansh Batra
Dept. of Information Technology
NSUT
New Delhi, India
devanshb.it.17@nsit.net.in

Ganesh Bagler
Center for Computational Biology
IIIT-Delhi
New Delhi, India
bagler@iiitd.ac.in

Abstract—Traditional cooking recipes follow a structure which can be modelled very well if the rules and semantics of the different sections of the recipe text are analyzed and represented accurately. We propose a structure that can accurately represent the recipe as well as a pipeline to infer the best representation of the recipe in this uniform structure. The Ingredients section in a recipe typically lists down the ingredients required and corresponding attributes such as quantity, temperature, and processing state. This can be modelled by defining these attributes and their values. The physical entities which make up a recipe can be broadly classified into utensils, ingredients and their combinations that are related by cooking techniques. The instruction section lists down a series of events in which a cooking technique or process is applied upon these utensils and ingredients. We model these relationships in the form of tuples. Thus, using a combination of these methods we model cooking recipe in the dataset RecipeDB [1] to show the efficacy of our method. This mined information model can have several applications which include translating recipes between languages, determining similarity between recipes, generation of novel recipes and estimation of the nutritional profile of recipes. For the purpose of recognition of ingredient attributes, we train the Named Entity Relationship (NER) models and analyze the inferences with the help of K-Means clustering. Our model presented with an F1 score of 0.95 across all datasets. We use a similar NER tagging model for labelling cooking techniques (F1 score = 0.88) and utensils (F1 score = 0.90) within the instructions section. Finally, we determine the temporal sequence of relationships between ingredients, utensils and cooking techniques for modeling the instruction steps.

Index Terms—Named Entity Recognition, Recipe Structure, Knowledge Mining, Clustering, POS Tagging

I. INTRODUCTION

Instructional language, comprising of step-by-step instructions to be performed in order to complete a task, is a very commonly used structure. Much work has been done to model the instructional prose [2]. Cooking recipes have traditionally been considered as a part of such instructional language paradigm, and specialised techniques have been developed to model their representation [3]. However, these techniques require labeled data. Other techniques use complex physical simulators in order to achieve lower level knowledge mining. Lately, unsupervised approaches specific to recipe datasets have caught traction [4]; however even these approaches model recipes as purely instructional in nature.

We challenge this assumption because of the lack of importance given to the ingredients section of the recipe. For advanced analysis of computable semantic insights from a
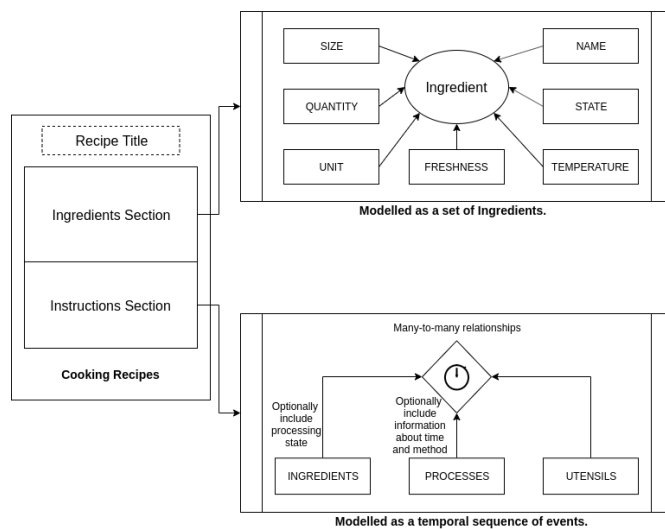

Fig. 1: *Diagrammatic representation of the proposed Recipe Data Structure*

recipe, the ingredients section is of utmost importance and should not be modelled like the rest of the instructional text to prevent data loss. Ingredient information itself can have use cases such as food pairing, flavor prediction, nutritional estimation, cost estimation and cuisine prediction. Therefore, there is a need to accurately represent the ingredient section in a computable format which can be further re-used for other applications similar to instructions. There are also a lot of differences between the natural language that the instruction sections follow and the ingredients section of the recipe which are devoid of known grammatical structures and rules. Thus, it can be understood why the modeling of both these sections must be done separately and using different techniques in order to get better inferences from the data, unlike the previous studies in this domain.

Fig. 1 presents various elements of our model used to represent both the instructions section and ingredients section of cooking recipes. A close look on the ingredients section presents it as a list of the ingredients and their corresponding attributes such as quantity, temperature and processing state. Our aim is to represent the ingredients section of recipes as a structure listing the various attributes. We explain the reasoning behind the attributes selected and our method of

TABLE I: *Results of Annotations on the Ingredients Section by the Named Entity Recognition Model*

Ingredient Phrase	Name	State	Quantity	Unit	Temperature	Dry/Fresh	Size
1 sheet frozen puff pastry (thawed)	puff pastry	thawed frozen	1	sheet	frozen		
6 ounces blue cheese,at room temperature	blue cheese		6	ounces			
1 tablespoon whole milk (or half-and-half)	milk		1	tablespoon			
2-3 medium tomatoes	tomato		2-3				medium
1/2 teaspoon pepper,freshly ground	pepper	ground	1/2	teaspoon			
1/2 teaspoon fresh thyme,minced	thyme	minced	1/2	teaspoon		fresh	
1 teaspoon extra virgin olive oil	extra virgin olive oil		1	teaspoon			

computation for the same as well as an analysis of how semantically meaningful these quantities are in the next section. Like the traditional approaches, we consider the instructions section to be comprised of a sequence of steps which provide a details of cooking events. We characterize the events as many-to-many relationships between three entities: utensils, processes and cooking techniques. Note that it is not necessary for all the three entities to co-occur. We provide more details on our approach for inferring these relationships in a subsequent section titled 'Knowledge Mining from Instruction Section'. This information can be used for further applications by interpreting Knowledge Graphs and Thought Graphs from such relationships, themes that have been well explored recently [5].

We acquired the dataset of 118,000 Recipes from RecipeDB [1]. This database comprises of recipes from primarily two websites AllRecipes.com and FOOD.com (formerly GeniusKitchen.com). We constructed our models, conducted subsequent analysis and testing based on this dataset. All data and code used in this paper are available online on GitHub. [1]

II. KNOWLEDGE MINING FROM INGREDIENTS SECTION

A. Challenges in identifying Ingredient and its Attributes

To extract ingredients and their corresponding attributes, we primarily investigate the ingredients section of a recipe. Ingredients occur in large variety across different recipes. The semantics in writing a recipe vary from region to region. In our database of recipes (118,000) from various regions across the world, we observe there exist primarily the following three challenges in extracting all ingredients and their associated attributes.

1) **Corpus of Attributes**: There is no known corpus which can identify all ingredients or all processing states in different cooking techniques. Recipes, by nature, are constantly evolving along with new ingredients and new cooking techniques. The model needs to be robust to identify unknown ingredients and unknown attributes.

2) **Identification of Attributes:** Attributes may be homographs. For example, "clove" (1 Clove) may refer to the ingredient itself or may refer to the unit of measurement (1 clove of cheese).

3) **Variation in Lexical Structure of Ingredient Phrases:** There exist a huge variation in the lexical structure of writing a ingredient phrase. The structure may be as simple as "34 cup sugar" OR "1 garlic clove, crushed"

to being as complex as "1 (8 ounce) package cream cheese, softened" or "1-2 fresh chili pepper very finely chopped."

B. Named Entity Recognition Model

Since the recipes evolve in terms of their ingredients and the corresponding attributes, understanding the structure of the ingredient phrases is critical.

Named Entity Recognition (NER) models are generally trained to automate the task of annotating words with tags. Hence, we propose training a NER model with the help of a manually annotated subset of recipes. The named entities we propose are– Name, Unit, Quantity, Processing State, Size, Dryness/Freshness state, and Temperature. These entities have been proposed after manually going through descriptions in the Standard Legacy Database by USDA, one of the recommended reference databases for nutritional information. We describe each of the tags in Table II.

We train the Stanford NER Tagger [6] and utilize it to annotate the ingredient sections of our database of 118,000 recipes. For more details of the training process and conducted refer II.F. Consider the recipe Tomato and Blue Cheese Tart [2]. Table I shows the results of applying the trained NER model for seven ingredient phrases.

C. Pre-Processing

Before passing the ingredient phrase to the NER model, we pre-process the ingredient phrases by eliminating stop words, lemmatizing all words using WordNet Lemmatizer, and converting the text to lower case. This pre-processing has advantages like correctly classifying ingredient terms which differ in plurality, capitalization, presence of hyphens as identical entities. For instance, both "tomatoes" and "Tomato" are output as "tomato", thus making the generated data much more useful for different purposes. As part of our implementation of the framework, we used the NLTK library for the purpose of pre-processing.

D. Parts of Speech (POS) Tagging to represent Vectors

As a preliminary experiment, we noticed that using a small set of annotated examples for training was not successful in accurately identifying the entities in the entirety of our diverse dataset. We attribute this to the presence of a number of lexical structures of the ingredient phrases which vary widely

across cuisines and original data sources, mentioned earlier as a challenge.

To efficiently tackle this problem, we devise a novel technique for finding the Parts of Speech (POS) Tags for these phrases. These Parts of Speech Tags assign tags to each token, such as noun, verb, adjective, etc. The mainstream sophisticated taggers available provide additional fine-grained POS tags such as 'noun-plural'.

In our experiment, we use the Stanford POS Twitter model [7] in order to find the POS tags for all ingredient phrases in the RecipeDB database. The intuition behind using this tagger is that ingredient phrases are not grammatically complete sentences, but resemble tweets in their lexical structures. We represented the output for every unique ingredient phrase as vectors (1×36) on the basis of the frequency of each POS tag present. The corresponding vectors for ingredient phrases with similar lexical structures will be close to each other in terms of euclidean distance. This intuition is backed by dimensional interrelationships from models like Word2Vec which is formed on a similar basis.

E. Analysis of vectors by clustering

We further create clusters within this vector space. The constituent ingredient phrases within a cluster are likely to have similar lexical structures. We can now pick a small subset of sentences such that it has a sufficient number of representatives from each of the clusters. These representative phrases form a good training set for the NER tagger, provided that the number of chosen phrases are sufficient and hyperparameters of the clustering algorithm have been optimised.

For the AllRecipes (16,000 Recipes) corpus (1.5 million of 11.5 million ingredient phrases), we grouped ingredient phrases into clusters. From each cluster, 1% unique ingredient phrases were picked which were used to form a training set of 1,500 ingredient phrases. Similarly, 0.33% of unique ingredient phrases (specifically excluding the ingredient phrases in the training set) were then picked which were used then used to form a testing set of 500 ingredient phrases. The same methodology was applied for the Food.com (102,000 recipes) corpus which had 10 million of the 11.5 million ingredient phrases. However, since the number of total ingredient phrases were much larger than the AllRecipes corpus (about 10 times as large) we proceeded to select 0.5% from each cluster to form the training set of 5000 ingredient phrases and to select 0.165% from each cluster to form the testing set of 1,500 ingredient phrases. Combining the above two corpora, we obtained a training set of 6,612 ingredient phrases and a testing set of 2188 ingredient phrases.

K-Means Clustering Algorithm was utilized to cluster the ingredient vectors. The clusters are formed on the basis of the frequency of the tags through a Bag-of-Words Approach. The decision of selecting the number of clusters formed was based on two factors: inertia of the clusters formed (Elbow Criterion Method [8]) and interpretation of the clusters. For example, the phrases "3 teaspoons olive oil" and "2 tablespoons all-purpose flour", ideally, should belong to the same cluster since

they correspond to form a similar sentence structure as they both contain one cardinal number, two nouns and one plural-noun. Based on these factors, **23 distinct clusters (Fig 2)** were identified. The ingredient phrases were then manually tagged as described in Table II.

TABLE II: Named Entity Recognition Tags

Tag	Significance	Example
NAME	Name of Ingredient	salt, pepper
STATE	Processing State of Ingredient.	ground, thawed
UNIT	Measuring unit(s)	gram, cup
QUANTITY	Quantity associated with the unit(s).	1, $1\frac{1}{2}$, 2-4
SIZE	Portion sizes mentioned	small, large
TEMP	Temperature applied prior to cooking	hot, frozen
DRY/FRESH	Fresh otherwise as mentioned.	dry, fresh

TABLE III: Training and Testing Dataset Sizes For NER on Ingredients Section

Datasets	AllRecipes	FOOD.com	BOTH
Training Set Size	1470	5142	6612
Testing Set Size	483	1705	2188

F. Evaluation

We trained the Stanford NER Model [6] based on our three datasets divided into training and testing sets as described in Table III. We tested the three models on the subsequent testing sets across each datasets as highlighted in Table IV. The models were validated by 5-fold cross validation. The NER model trained on AllRecipes Dataset gives an F1 Score of 0.9682 on the AllRecipes.com Test Set. Similarly, the NER Model trained of FOOD.com Dataset gives an F1 Score of 0.9519 on the FOOD.com Test Set. However, the individual models do not consistently give the same F1 scores across all datasets. We attribute this to the variation of the recipes between the two datasets. The model trained on the composite dataset of AllRecipes and FOOD.com delivers the best performance with an F1 scores of over 0.95 across the three datasets. The model is significant as it allows to derive ingredient and its corresponding attributes given an ingredient phrase from the ingredients section.

From our database of 118,000 recipes, Through our model we extracted **20,280** unique ingredient names. It is important to note that there are a number of aliases of same ingredients included in this final count. For example, okhra and ladyfinger are counted as two different ingredient names although they represent the same ingredient.

TABLE IV: Evaluation of NER Model for Ingredients Section

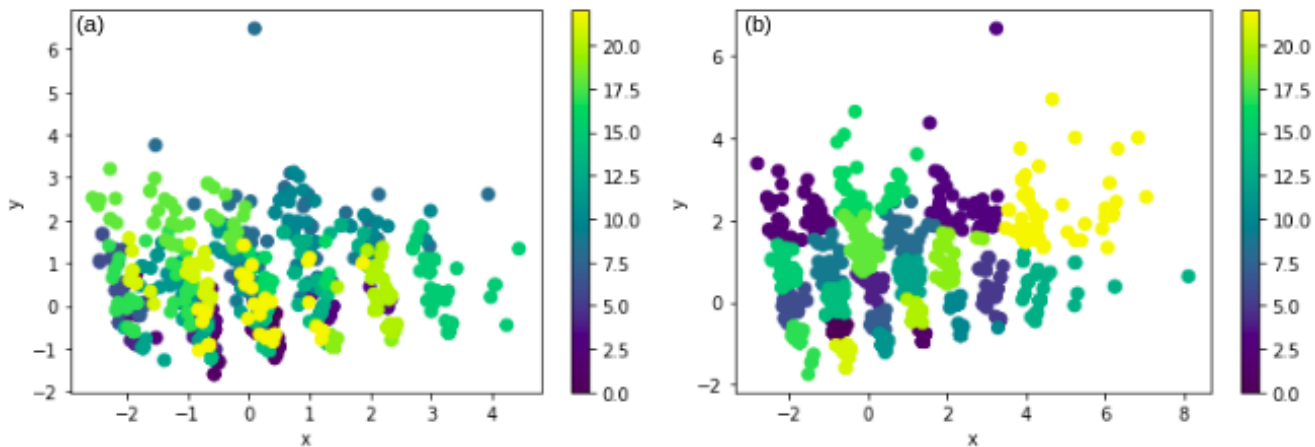| Testing Set	Training Set Model		
	AllRecipes	FOOD.com	BOTH
AllRecipes	**0.9682**	0.9317	**0.9709**
FOOD.com	0.8672	**0.9519**	0.9498
BOTH	0.8972	0.9472	**0.9611**

Fig. 2: *(a) Visualization of the Clusters formed by applying k - means Clustering on the vectors formed by determining the frequency of POS Tags in the ingredient phrase. **After clustering the vectors**, the visualization has been generated by applying Principal Component Analysis on the vectors for converting 36 - dimensional vectors into 2 dimensions. The clusters are separable at higher dimensions (b) Visualization of the clusters formed by applying k - means Clustering on the vectors formed by determining the frequency of POS Tags [11] in the ingredient phrase. **Before clustering the vectors**, we apply Principal Component Analysis on the vectors for converting 36 - dimensional vectors into 2 dimensions. We then proceed to cluster the vectors. The visualizations display 50 unique ingredient phrases from each of the 23 distinct clusters*

III. Knowledge Mining From Instructions Section

In order to help machine understanding of structured sequential instructions data, a lot of approaches have been applied historically [9]. However, unsupervised learning in this domain has only caught up recently [10]. Our goal is to similarly learn the event chain within a recipe in an unsupervised manner because of the complex and extremely time-consuming task of labelling training data.

We define recipe instructions as being a narrative chain a sequence of events which is constituted of multiple protagonists interacting together in predefined relationships. The protagonists in a recipe are the utensils and ingredients. We model the relationships using Cooking Techniques/Processes (the act of processing ingredients and utensils such as preheating an oven, or boiling a potato). Thus the task is to identify these temporal relations. For this purpose we follow a two step strategy.

A. Entity Recognition within Instruction Set

Firstly, we train another NER model which identifies the Processes, Utensils and Ingredients within the instruction set. The training process was very similar to the one followed for the ingredients section. The recipes with the longest instructions section from 40 different cuisines were extracted to further annotate 268 processes/techniques, ingredients, and 69 utensils. The Stanford NER tagger [6] was trained upon this corpus. The model thus obtained was then used for tagging the cooking recipes within RecipeDB.

We used this pre-trained NER model to label entities within the instruction sets of recipes in RecipeDB after pre-processing them in a manner similar to the one implemented for ingredients section. We then used the threshold frequencies (47 and

10 respectively), to create a dictionary of Cooking Techniques and Utensils which was used to filter the data from the NER model, removing most of the inconsistencies.

Fig. 4 shows an example of entity tags inferred using this approach. The performance of the model is explained by Table V.

TABLE V: Evaluation of NER model for Instructions Section

F1	Precision	Recall	F1 Score
Processes	0.92	0.85	0.88
Utensils	0.94	0.86	0.90

Performance Analysis of Named Entity Recognition Model on Instructions Section for the extraction of Processes and Utensils

B. Relationship Extraction

In order to extract the events from instructions section of cooking recipes, we study the grammatical structure of these recipes to form tuples which can model the section well. Many approaches have already been proposed for dependency parsing and subject-verb extraction for relation extraction [6]. We have described below our approach in this regard. The tuples to be inferred include the entities (both Utensils and Ingredients) and the relationships between these entities. The relationships are defined by the Cooking Techniques being applied on the ingredients and utensils. The relationships are many-to-many for a particular Cooking Technique at an instance. This can be better understood with an example. An instruction may state that potatoes are boiled. It may also state that potatoes are fried with olive oil in a pan. In this case, the relation extends to both the ingredients - olive oil and potatoes as well as the utensil - pan.
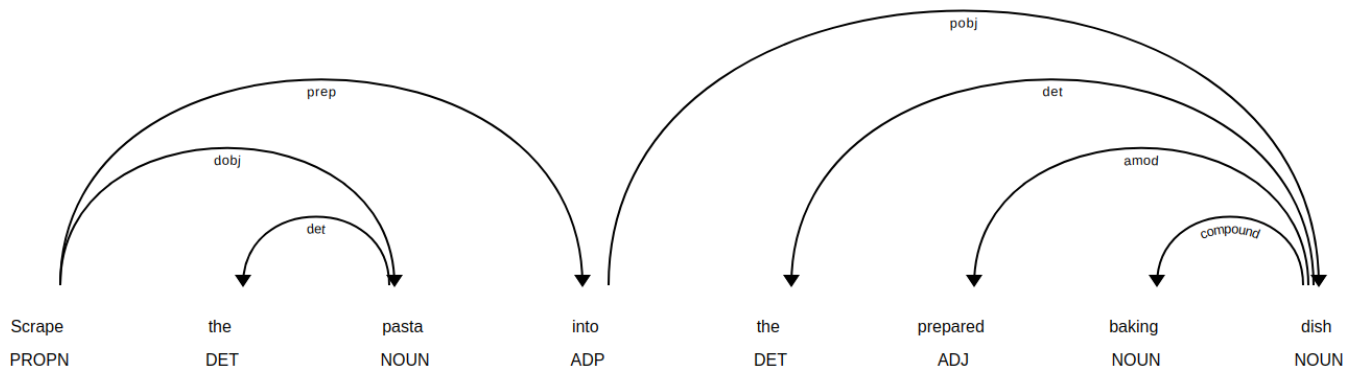
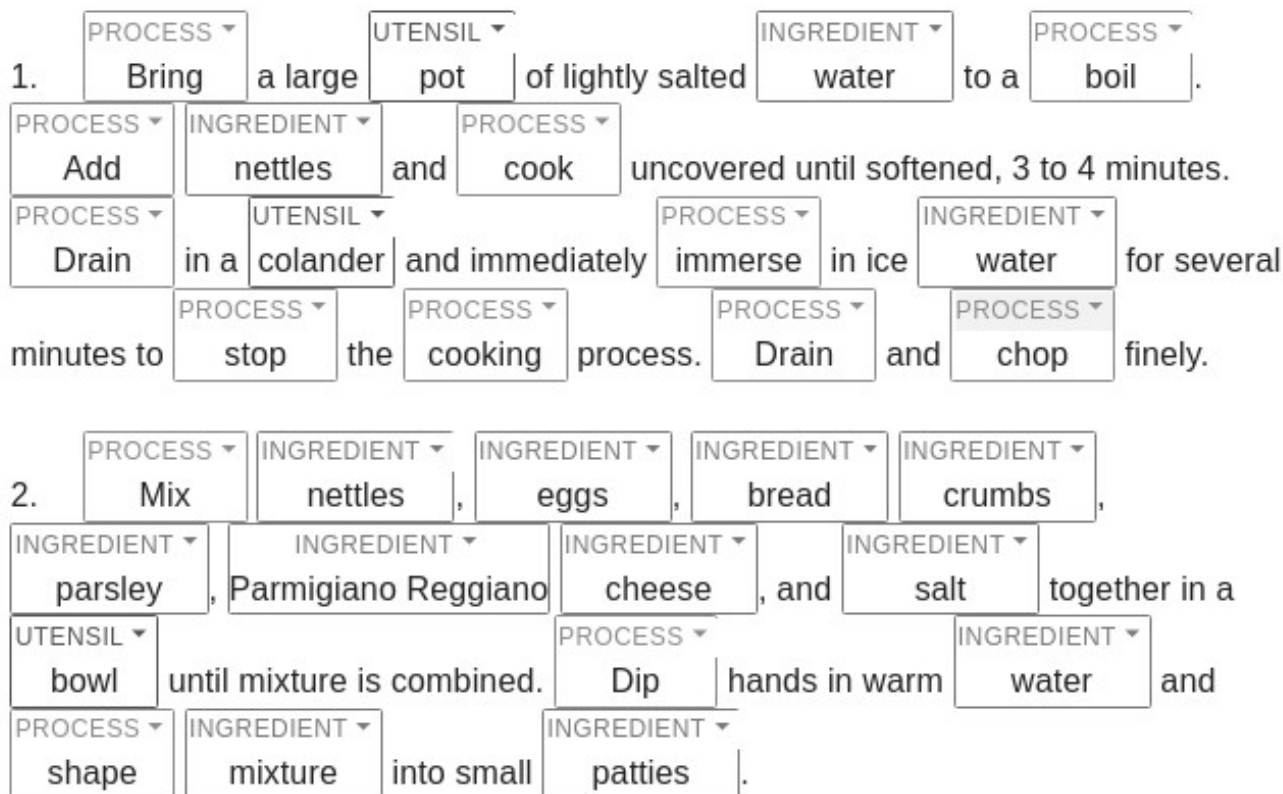Fig. 3: *A dependency parsed structure of a typical instruction using the Spacy library*

Fig. 4: *The inference from the NER model for the instruction section of a recipe*

Fig 3 shows the dependency parsing tree obtained using the Spacy library [12]. Our aim is to find out the relationships using grammatical rules. It is easy to follow from intuition that the Cooking Techniques are generally tagged as verb in any instruction. Thus, for all the verbs classified as processes (having filtered after pre-processing and comparing with the terms in the dictionary mentioned before), we attempt to find the associated subjects and objects within the dependency tree. We also obtain prepositional objects. These are likely to be the processes and utensils associated with the relationship. In order to satisfy the many-to-many entity criteria, we also filter this list of relationships using the NER inferred Ingredients and Utensils to obtain the final relationships. Fig. 5 shows an example of relationships obtained using this approach.

IV. APPLICATIONS AND FUTURE SCOPE

The proposed data structure captures a significant part of the cooking recipe: Ingredients and their associated characteristics; Cooking processes utilised in the recipe; Temporal sequence of cooking processes; Relationship between cooking processes and ingredients; Utensils utilised in the recipe; and Temporal relationships between utensils, ingredients and processes.
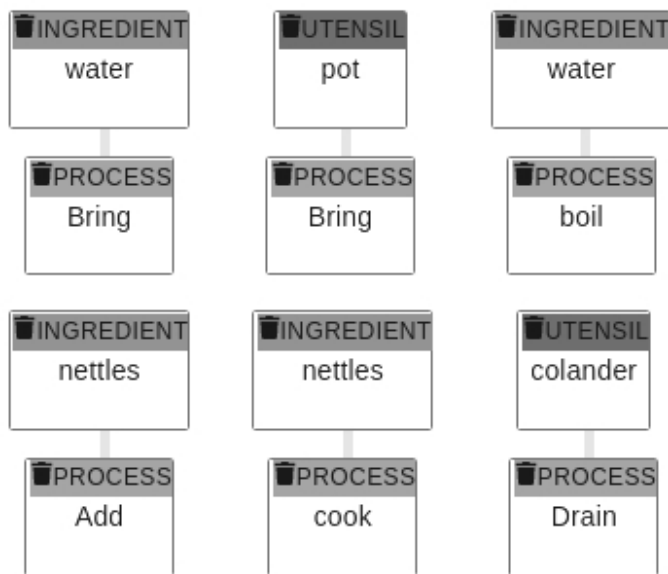
Fig. 5: *The inference from relationship tagging for part of the first instruction in Fig. 4. Notice that both Bring + Water and Bring + Pot relations are modeled by the same process - Bring. Multiple such one-one relations are combined together if they are found to be the dependency of the same process to form compound many-to-many relations*

Such a structured information is of key value for researchers. A similar model which does not take into account the information from Ingredients Section has been useful in tasks like Artificial Intelligence based Recipe Text Generation [4]. Our model could help to extract information out of the Ingredients Section synchronised with the Instructions Section. We have implemented our model for estimation of nutritional from cooking recipes [13], as well as for finding similar recipes in RecipeDB [1].

V. CONCLUSIONS

We present a novel method to extract ingredients and its corresponding attributes in a structured format via the utilization of clustering and Named Entity Recognition. We also present a labelled dataset of 8800 Ingredient phrases, divided into training and testing sets III, manually tagged into 7 categories as described in Table II. Our Best Model gives a F1 Score of greater than or equal to 0.95 across all datasets.

For data extraction from the instruction section, we applied the aforementioned pipeline (from Section III) on 40,000 randomly chosen recipes with 174,932 instruction steps from RecipeDB [1]. After filtering out the tuples to only include the relationships defined by processes predicted by the NER tagger, we observed that on average, an instruction yielded 6.164 relations with a standard deviation of 5.70 which shows the large variation in the number of entities a cooking technique may be applied on. The large standard deviation re-affirms the need to model relationships in a many-to-many fashion. Else, a

lot of information about an event may have been lost if one-to-many, one-to-one, or many-to-one relations were used for the modeling since they would not have been able to accommodate the large variation in number of ingredients and utensils at the same time.

In summary, we propose the Named Entity based approach to represent recipes. This approach has several applications which include translating recipes between languages, determining similarity between recipes, generation of novel recipes and estimation of nutritional profile of recipes. The proposed model was utilized in RecipeDB [1] to determine similarity between recipes and for nutritional estimation of recipes.

REFERENCES

[1] Batra, Devansh, Nirav Diwan, Utkarsh Upadhyay, Jushaan S. Kalra, Tript Sharma, et al. "RecipeDB: A Resource for Exploring Recipes." Available at SSRN 3482237 (2019), preprint.

[2] Lau, Tessa, Clemens Drews, and Jeffrey W. Nichols. "Interpreting written how-to instructions." Twenty-First International Joint Conference on Artificial Intelligence. 2009, doi: 10.1.1.149.9141

[3] Mori, Shinsuke, et al. "Flow Graph Corpus from Recipe Texts." LREC. 2014, pp. 23702377.

[4] Kiddon, Chlo, et al. "Mise en place: Unsupervised interpretation of instructional recipes." Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. 2015, pp. 982992

[5] Hao, Qi, Jeroen Keppens, and Odinaldo Rodrigues. "A Verb-based Algorithm for Multiple-Relation Extraction from Single Sentences." Proceedings of the International Conference on Information and Knowledge Engineering (IKE). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2017, pp. 115-121

[6] Finkel, Jenny Rose, Trond Grenager, and Christopher Manning. 2005. "Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling". Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005), pp. 363-370.

[7] Toutanova, Kristina and Christopher D. Manning. 2000. "Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagger"". In Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-2000), pp. 63-70.

[8] Kodinariya, Trupti M., and Prashant R. Makwana. "Review on determining number of Cluster in K-Means Clustering." International Journal 1.6, 2013, pp. 90-95.

[9] Schank, Roger C., and Robert P. Abelson. "Scripts, plans, goals, and understanding: An inquiry into human knowledge structures." Psychology Press, 2013, doi: 10.4324/9780203781036.

[10] Chambers, Nathanael, and Dan Jurafsky. "Unsupervised learning of narrative event chains." Proceedings of ACL-08: HLT. 2008, pp. 789-797

[11] Schmid, Helmut. "Part-of-speech tagging with neural networks." Proceedings of the 15th conference on Computational linguistics-Volume 1. Association for Computational Linguistics, 1994. doi: 10.3115/991886.991915

[12] Honnibal, Matthew, and Ines Montani. "Spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing.", unpublished (to appear).

[13] Kalra, Jushaan S., Devansh Batra, Nirav Diwan and Ganesh Bagler. "Nutritional Profile Estimation in Cooking Recipes", unpublished (submitted to DECOR workshop at International Conference in Data Engineering) (2020).