Identifying Key Entities in Recipe Data - Report

Business Objective

The source dataset comprises culinary recipes with a focus on ingredient extraction and analysis. Each recipe features a structured ingredient list with labelled components, identifying ingredients, quantities and units. This diverse collection supports tasks such as understanding recipes and discovering culinary knowledge, enabling the development of models for information extraction in the culinary domain.

Input Data

- The input data comprises a Json file with two columns input & pos with 285 rows
- Column input comprises all the ingredients used for the Recipe & the Column pos comprises the corresponding POS tag.

```python
# display first five rows of the dataframe - df
inputDF.head(5)
```

Python

	input	pos
0	6 Karela Bitter Gourd Pavakkai Salt 1 Onion 3 tablespoon Gram flour besan 2 teaspoons Turmeric powder Haldi Red Chilli Cumin seeds Jeera Coriander Powder Dhania Amchur Dry Mango Sunflower Oil	quantity ingredient ingredient ingredient ingredient ingredient quantity ingredient quantity unit ingredient ingredient ingredient quantity unit ingredient ingredient ingredient ingredient ingredient ingredient ingredient ingredient ingredient ingredient ingredient ingredient ingredient ingredient ingredient ingredient
1	2-1/2 cups rice cooked 3 tomatoes teaspoons BC Belle Bhat powder 1 teaspoon chickpea lentils 1/2 cumin seeds white urad dal mustard green chilli dry red 2 cashew or peanuts 1-1/2 tablespoon oil asafoetida	quantity unit ingredient ingredient quantity ingredient unit ingredient ingredient ingredient ingredient quantity unit ingredient ingredient quantity ingredient ingredient ingredient ingredient ingredient ingredient ingredient ingredient ingredient ingredient quantity ingredient ingredient ingredient quantity unit ingredient ingredient
2	1-1/2 cups Rice Vermicelli Noodles Thin 1 Onion sliced 1/2 cup Carrots Gajjar chopped 1/3 Green peas Matar 2 Chillies 1/4 teaspoon Asafoetida hing Mustard seeds White Urad Dal Split Ghee sprig Curry leaves Salt Lemon juice	quantity unit ingredient ingredient ingredient ingredient quantity ingredient ingredient quantity unit ingredient ingredient ingredient quantity ingredient ingredient ingredient quantity ingredient quantity unit ingredient ingredient ingredient ingredient ingredient ingredient ingredient ingredient ingredient ingredient unit ingredient ingredient ingredient ingredient ingredient
3	500 grams Chicken 2 Onion chopped 1 Tomato 4 Green Chillies slit inch Ginger finely 6 cloves Garlic 1/2 teaspoon Turmeric powder Haldi Garam masala tablespoon Sesame Gingelly Oil 1/4 Methi Seeds Fenugreek Coriander Dhania Dry Red Fennel seeds Saunf cups Sorrel Leaves Gongura picked and	quantity unit ingredient quantity ingredient ingredient quantity ingredient quantity ingredient ingredient ingredient unit ingredient ingredient quantity unit ingredient quantity unit ingredient ingredient ingredient ingredient ingredient unit ingredient ingredient ingredient quantity ingredient ingredient ingredient ingredient ingredient ingredient ingredient ingredient ingredient ingredient unit ingredient ingredient ingredient ingredient ingredient
4	1 tablespoon chana dal white urad 2 red chillies coriander seeds 3 inches ginger onion tomato Teaspoon mustard asafoetida sprig curry	quantity unit ingredient ingredient ingredient ingredient quantity ingredient ingredient ingredient ingredient quantity unit ingredient ingredient ingredient unit ingredient ingredient unit ingredient

Generate + Code + Markdown

```python
# print the dimensions of dataframe - df
inputDF.shape
```

Python

```
    # print the information of the dataframe
    inputDF.info()
```

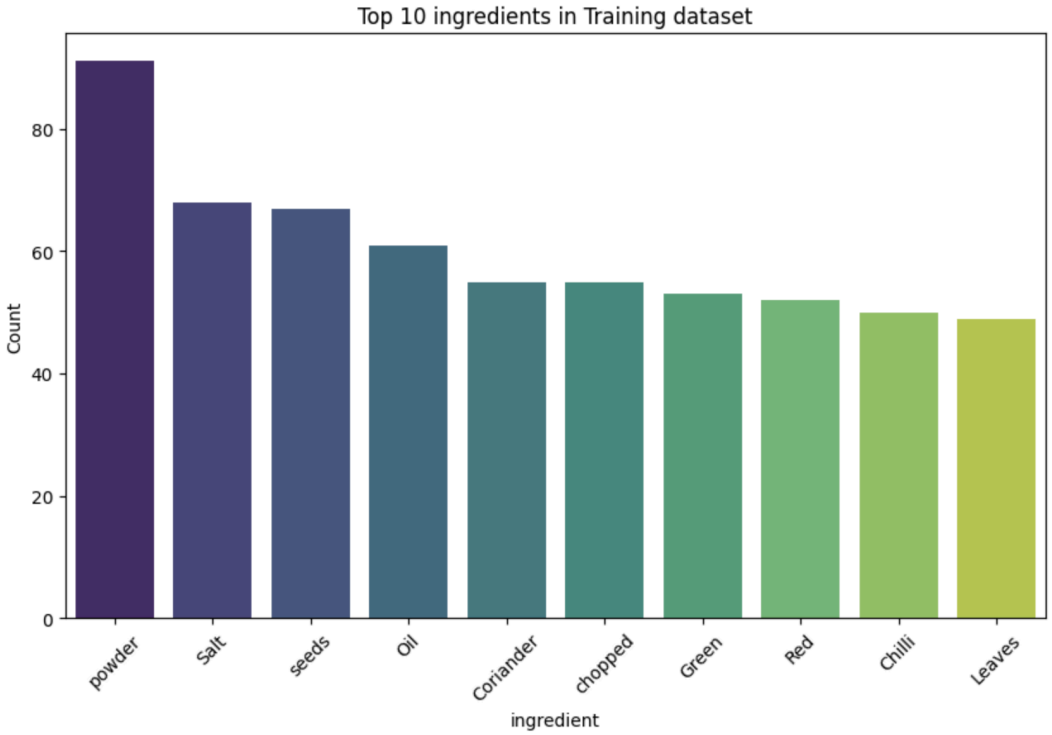
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 285 entries, 0 to 284
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   input   285 non-null    object
 1   pos     285 non-null    object
dtypes: object(2)
memory usage: 4.6+ KB
```

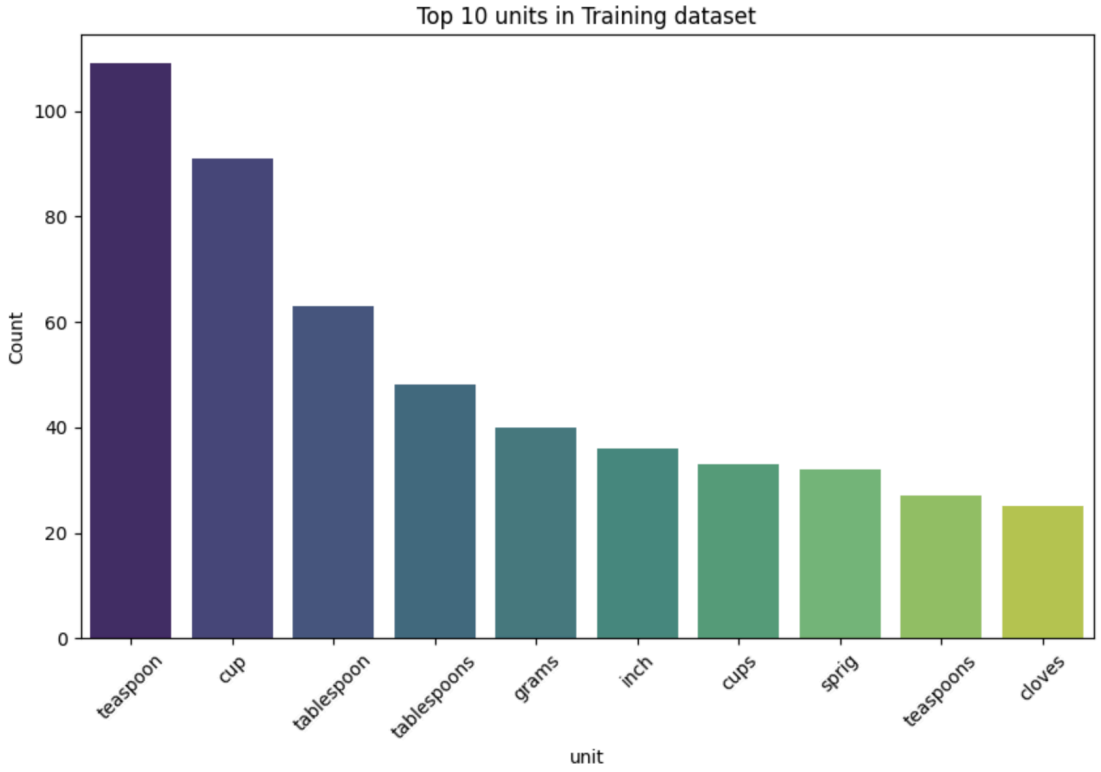
- There were 81 records for which the pos_tokens length was not equal to input_text length. So, they were dropped.

Data Analysis

```
First 10 records in Training dataset:
Input Token --> POS Token
1 --> quantity
cup --> unit
Gram --> ingredient
flour --> ingredient
besan --> ingredient
1/2 --> quantity
teaspoon --> unit
Ajwain --> ingredient
Carom --> ingredient
seeds --> ingredient
```

```
First 10 records in Validation dataset:
Input Token --> POS Token
1 --> quantity
cup --> unit
rice --> ingredient
12 --> quantity
small --> unit
onions --> ingredient
2 --> quantity
cloves --> ingredient
garlic --> ingredient
inch --> unit
```


Top 10 ingredients in Training dataset
Count
0
20
40
60
80
powder
Salt
seeds
Oil
Coriander
chopped
Green
Red
Chilli
Leaves
ingredient


Top 10 units in Training dataset
Count
0
20
40
60
80
100
teaspoon
cup
tablespoon
tablespoons
grams
inch
cups
sprig
teaspoons
cloves
unit

Observations

- Powder is the top most ingredient in the entire dataset
- Along with powder, Salt & Seeds are also mostly commonly used
- Teaspoon, cup & tablespoon are the most units

Feature Engineering

- We identified common Unit and Quantity keywords by analyzing frequent terms.
- We defined regular expressions to match different forms of Units
- We also converted words to features to analyse the impact

Core Features

- Token properties
- Character patterns

Quantity & Unit Detection

- Regular Expressions for Quantities
- Numeric pattern recognition for Units

Contextual Features

- Boundary Markers
- Previous & Next words/tokens

Model Training

CRF model is used to train the recipe data. Below are the parameters initialized:

```
    # initialise CRF model with the specified hyperparameters and use weight_dict
    crfModel = sklearn_crfsuite.CRF(
        algorithm='lbfgs',
        c1=0.5,
        c2=1.0,
        max_iterations=100,
        all_possible_transitions=True
        )
    # train the CRF model with the weighted training data
    crfModel.fit(XTrainWeightedFeatures, yTrainLabels)
```

```
CRF(algorithm='lbfgs', all_possible_transitions=True, c1=0.5, c2=1.0,
    max_iterations=100)
```

Model Evaluation

Training Data Results

	precision	recall	f1-score	support
ingredient	1.00	1.00	1.00	3411
quantity	1.00	1.00	1.00	658
unit	0.99	1.00	1.00	563
accuracy			1.00	4632
macro avg	1.00	1.00	1.00	4632
weighted avg	1.00	1.00	1.00	4632

```
Confusion Matrix:
Labels ['unit', 'ingredient', 'quantity']
[[ 562    0    1]
 [   0 3411    0]
 [   3    0  655]]
```

Validation Data Results

```
    # specify flat classification report
    yValTrueFlat = [label for sent in yValLabels for label in sent]
    yValPredFlat = [label for sent in yValPred for label in sent]
    print(classification_report(yValTrueFlat, yValPredFlat))
```

```
              precision    recall  f1-score   support

  ingredient       1.00      1.00      1.00      1611
    quantity       1.00      0.99      1.00       294
        unit       0.99      1.00      1.00       244

    accuracy                           1.00      2149
   macro avg       1.00      1.00      1.00      2149
weighted avg       1.00      1.00      1.00      2149
```

```
Confusion Matrix:
Labels ['unit', 'ingredient', 'quantity']
[[ 244    0    0]
 [   0 1611    0]
 [   2    0  292]]
```

Validation Accuracy is very good - **0.9906**
This indicates that the Model has been trained very well.

Error Analysis

```
Label: quantity, ErrorCount: 2, Class Weight: 7.0395
Error DataFrame:
    token true_label predicted_label prev_token next_token  class_weight
0  little   quantity            unit     leaves       Salt      7.039514
1   taste   quantity            unit        per        1/2      7.039514
```

Conclusion

- Model shows strong performance in recognising recipe entities.
- Model Accuracy is very high on both Training & Test data sets
- Recommend this model to be tested on other real world practical data.