Identifying key Entities in Recipe data

Aditya Karnik

Objective

The business objective is to leverage the increasing popularity of online cooking platforms and meal-planning apps by enhancing the user experience. This can be achieved by implementing a custom-named entity recognition (NER) model to automatically tag ingredients, quantities and recipe names. This automation will streamline the process of organising recipes, improve searchability and enable users to easily find recipes based on available ingredients, portion sizes or specific dietary requirements. This will ultimately reduce the labour-intensive and inefficient manual tagging process, providing a more accessible and efficient way for businesses in the food and recipe industry to manage their recipe databases.

Given Data

Filename: ingredient_and_quantity.json

File type: Json

Structure: The data contains recipes in the input column and corresponding custom tags in the pos column. Samples are given below,

[

{

"input": "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",

"pos": "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"

},

{

"input": "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",

"pos": "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"

}

]

CRF model parameters:

CRF Model Hyperparameters Explanation

Parameter	Description
algorithm='lbfgs'	Optimisation algorithm used for training. `lbfgs` (Limited-memory Broyden–Fletcher–Goldfarb–Shanno) is a quasi-Newton optimisation method.
c1=0.5	L1 regularisation term to control sparsity in feature weights. Helps in feature selection.
c2=1.0	L2 regularisation term to prevent overfitting by penalising large weights.
max_iterations=100	Maximum number of iterations for model training. Higher values allow more convergence but increase computation time.
all_possible_transitions=True	Ensures that all possible state transitions are considered in training, making the model more robust.

Coding platform: Google colab in python notebook.

Important libraries -

1. Sklearn
2. Counter from collection
3. Pandas
4. Numpy

5. math
6. Joblib
7. spacy
8. Seaborn
9. matplotlib.pyplot

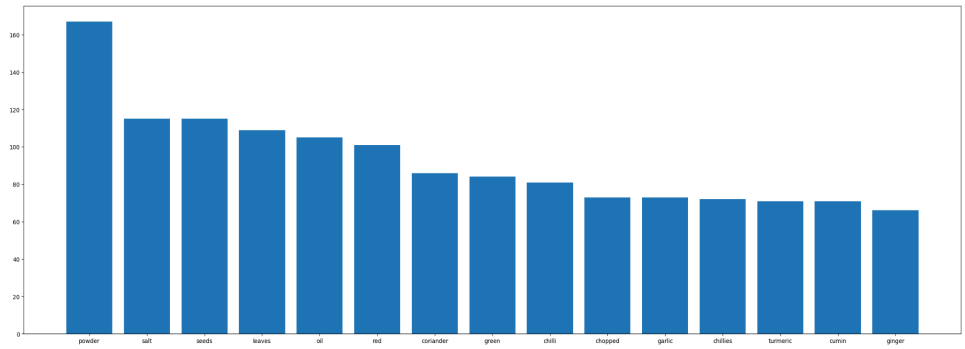
Steps:

1) Read the data from the file
2) Cleanup of the data
3) Standardisation
4) Conversion of sentences and corresponding tags in lists
5) Train test split in 0.7 and 03 respectively
6) EDA of the data with univariate and bivariate analysis
    a) Create tokens out of data
    b) Convert tokens in labels
    c) Check the frequency of the recipe words and corresponding tags along with the most frequently used words.
7) Load spacy 'en_core_web_sm' model for pos tagging
8) Define features for words
9) Use words' features on sentences and then on the dataframes
10) Extract labels from the target variable(y)
11) Use tf-idf to create the weighted dictionary
12) Extract features with class weights using X and y and weighted dictionary
13) Use these features along with CRF hyperparameters to build the CRF model
14) Use it for the training data
15) Predict the training data
16) Compare predicted Vs actual values of trained data
17) Predict the validation data
18) Compare predicted vs actual values of validation data
19) Check the accuracy and confusion matrix in both the scenarios

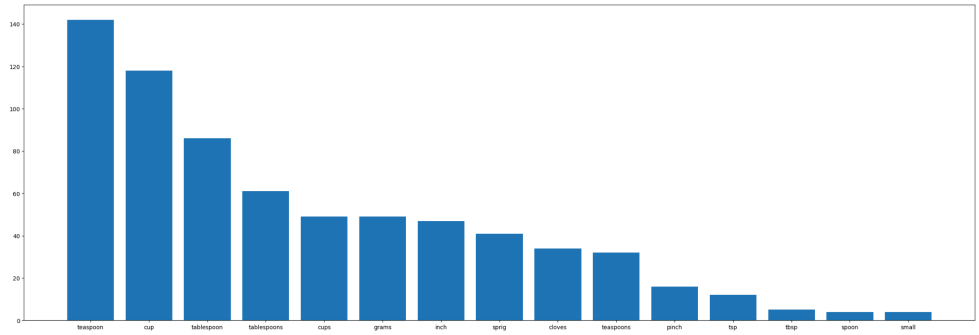
RESULTS

1. EDA analysis:

Top ingredients



Top units



Sample feature extraction and labeling along with weights:

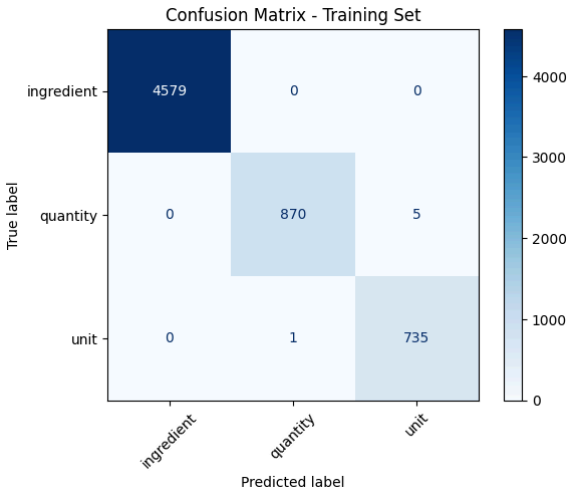
```
iteration 16

{'bias': 1.0, 'token': 'dhania', 'lemma': '', 'pos_tag': '', 'tag':
'', 'dep': '', 'shape': 'xxxx', 'is_stop': False, 'is_digit': False,
'has_digit': False, 'has_alpha': True, 'hyphenated': False,
'slash_present': False, 'is_title': False, 'is_upper': False,
'is_punct': False, 'is_quantity': False, 'is_unit': False,
'is_numeric': False, 'is_fraction': False, 'is_decimal': False,
'preceding_word': 'powder', 'following_word': 'red', 'prev_token':
'powder', 'prev_is_quantity': False, 'prev_is_digit': False,
'next_token': 'red', 'next_is_unit': False, 'next_is_ingredient':
True}

Ingredient

----------

iteration 17
```

```
{'bias': 1.0, 'token': 'red', 'lemma': '', 'pos_tag': '', 'tag': '',
'dep': '', 'shape': 'xxx', 'is_stop': False, 'is_digit': False,
'has_digit': False, 'has_alpha': True, 'hyphenated': False,
'slash_present': False, 'is_title': False, 'is_upper': False,
'is_punct': False, 'is_quantity': False, 'is_unit': False,
'is_numeric': False, 'is_fraction': False, 'is_decimal': False,
'preceding_word': 'dhania', 'following_word': 'turmeric',
'prev_token': 'dhania', 'prev_is_quantity': False, 'prev_is_digit':
False, 'next_token': 'turmeric', 'next_is_unit': False,
'next_is_ingredient': True}

ingredient

----------
```

Evaluation of training data using CRF model:

	precision	recall	f1-score	support
ingredient	1.00	1.00	1.00	4579
quantity	1.00	0.99	1.00	875
unit	0.99	1.00	1.00	736
accuracy			1.00	6190
macro avg	1.00	1.00	1.00	6190
weighted avg	1.00	1.00	1.00	6190

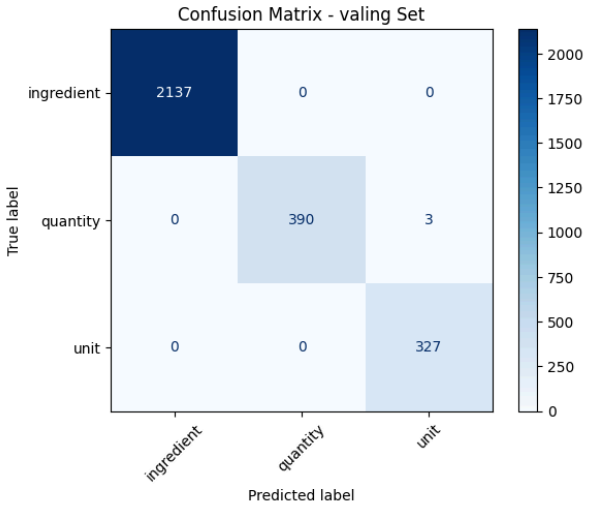
Confusion matrix of training data using CRF model:



Evaluation of the testing data using CRF model:

	precision	recall	f1-score	support
ingredient	1.00	1.00	1.00	2137
quantity	1.00	0.99	1.00	393
unit	0.99	1.00	1.00	327
accuracy			1.00	2857
macro avg	1.00	1.00	1.00	2857
weighted avg	1.00	1.00	1.00	2857

Confusion matrix:



Acknowledgement:

- IIITB and Upgrad tutorials on Exploratory Data Analysis (EDA) on the learning platform