Contents

Business Objective

The goal of this assignment is to train a Named Entity Recognition (NER) model using Conditional Random Fields (CRF) to extract key entities from recipe data. The model will classify words into predefined categories such as ingredients, quantities and units, enabling the creation of a structured database of recipes and ingredients that can be used to power advanced features in recipe management systems, dietary tracking apps, or e-commerce platforms.

Steps

1. Data preparation
2. Text processing
3. Model Training/ Feature engineering
4. Model Evaluation

Data Preparation

- The given data is in JSON format, representing a structured recipe ingredient list with Named Entity Recognition (NER) labels.

{
"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"
}

- Conversion from JSON to tabular format (e.g., DataFrame)
- Each token in a string is annotated as one of the following:
    - Quantity
    - Unit
    - Ingredient

Technical Details

- Techniques such as regular expressions and token normalization were used to clean and standardize the text.
- Additional preprocessing included part-of-speech (POS) tagging and token splitting for better structure.
- CRF for structured prediction in sequential data
- Flat classification metrics for evaluation
- The final model was saved using joblib, enabling easy deployment and reuse.
- Data visualization using tools such as Matplotlib and Seaborn.

Model Training

- A CRF model was implemented using the sklearn_crfsuite library.
- CRFs are ideal for sequence labelling because they account for contextual dependencies between adjacent labels.
- Custom features were extracted for each token to improve model accuracy.
- These include:
    - Lowercase transformation of the word
    - Common word suffixes
    - Identification of numeric and fractional tokens
    - Position-based context (previous and next words)
    - Capitalization status and punctuation flags

Model Evaluation

- The performance of the trained model was evaluated using
    - Precision
    - Recall
    - F1-score
    - Confusion matrix and classification report for deeper error analysis.

Scores

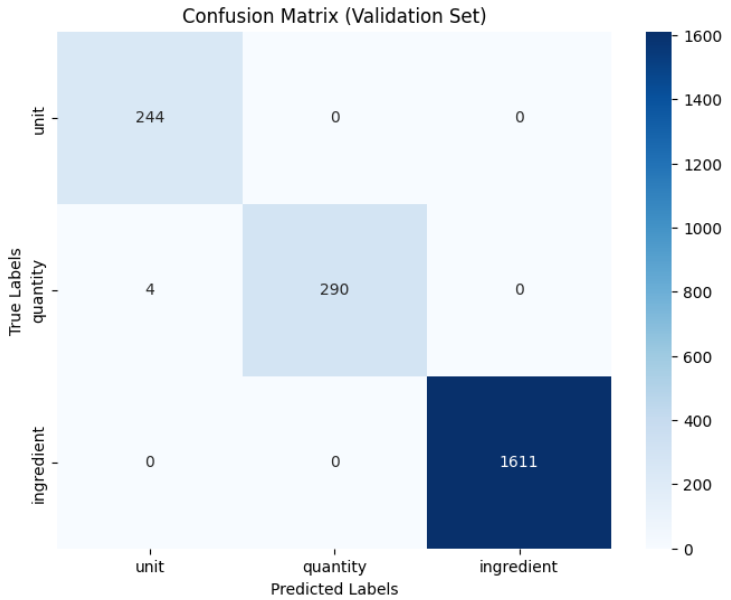
- Overall Model score

	precision	recall	f1-score	support
ingredient	1.00	1.00	1.00	1611
quantity	1.00	0.99	0.99	294
unit	0.98	1.00	0.99	244
accuracy			1.00	2149
macro avg	0.99	1.00	1.00	2149
weighted avg	1.00	1.00	1.00	2149

- Individual class label accuracy.

Label	Class Weight	Accuracy
ingredient	0.2227	1.00
Quantity	2.4197	0.9951
unit	2.924	0.9944

Confusion Matrix

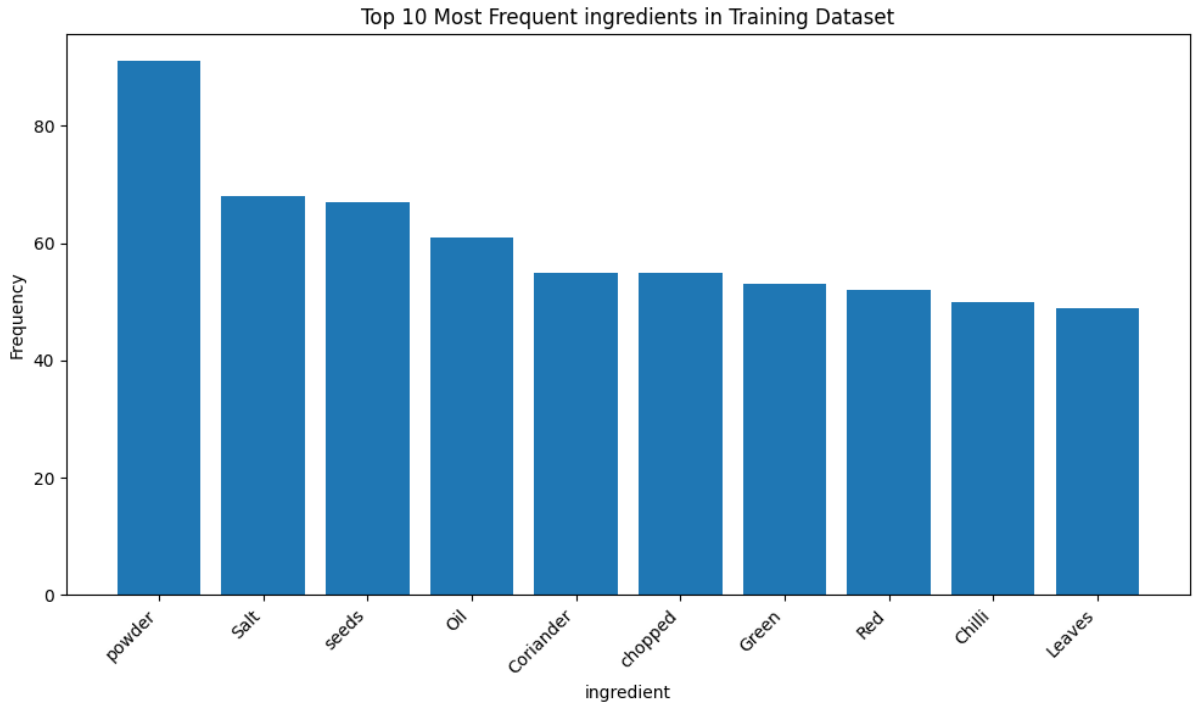


Errors

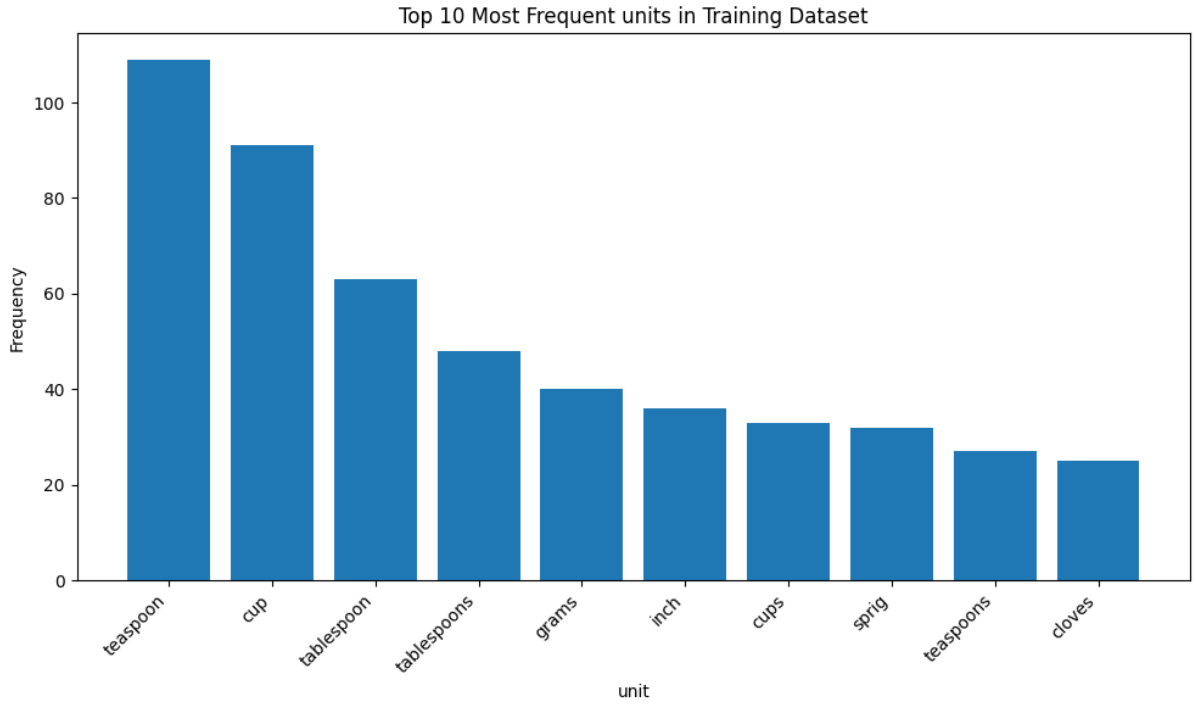
	token	previous_token	next_token	true_label	predicted_label	context
0	little	leaves	Salt	quantity	unit	curry leaves little Salt as
1	per	as	taste	quantity	unit	Salt as per taste 1/2
2	taste	per	1/2	quantity	unit	as per taste 1/2 cup
3	a	Leaves	bunch	quantity	unit	Dhania Leaves a bunch finely

EDA – Insights

- Top most frequent ingredients



- Top most frequent units



- The most common ingredients are powder and salt.
- "Chopped" and "green" are the most frequent preparation styles.
- The frequent mention of "seeds" indicates a reliance on whole spices (e.g., cumin, mustard).
- Recipes predominantly use teaspoons and cups (almost 50% of unit mentions), followed by tablespoons, less frequent use units like inches and sprigs.

Conclusion

1. Data Preparation
    - The initial dataset contained inconsistencies where the lengths of the 'input' and 'pos' fields did not match. These inconsistencies were identified and addressed by dropping the affected rows, ensuring data integrity for model training.
2. Overall Accuracy
    - The model demonstrates a high overall accuracy on the validation set (99.81%), indicating its ability to generalize well to unseen data.
3. Label-Wise Performance
    - While the overall accuracy is high, the label-wise performance reveals nuances.
    - Certain labels, ingredient, unit, achieve near-perfect accuracy (100%), suggesting the model effectively identifies it. However, other labels quantity shows a slightly lower accuracy (98.64%), indicating the model might struggle with more complex names or those with ambiguous contexts
4. Error Analysis Insights
    - The analysis of misclassified samples reveals that the model occasionally misclassifies unit vs quantities. Further investigation of those specific tokens and their contexts is needed.
    - Data quality: If the validation set includes inconsistencies or errors in labelling, they would impact the model's performance.
5. Class Weights Influence
    - The observed class weights, especially the adjusted weight (2.7425) for 'unit', likely played a role in the model's performance.
    - The lower weight (0.2263) for 'ingredient' indicates an effort to prevent overfitting to the more frequent 'ingredient' class, potentially helping improve the recognition of other labels.
6. Further Improvement
    - Adding more contextual features: Enhance feature engineering to capture more contextual information.
    - Incorporating more training data: Include more instances of ambiguous or rare ingredients and units in the training data.
    - Adjusting class weights: Fine-tune class weights further to optimize the balance between classes.