Contents

Business Objective 2

Data Description........ 2

EDA – Insights........ 2

Model Evaluation 3

Scores 3

Confusion Matrix........ 4

Errors........ 4

Conclusion........ 4

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.

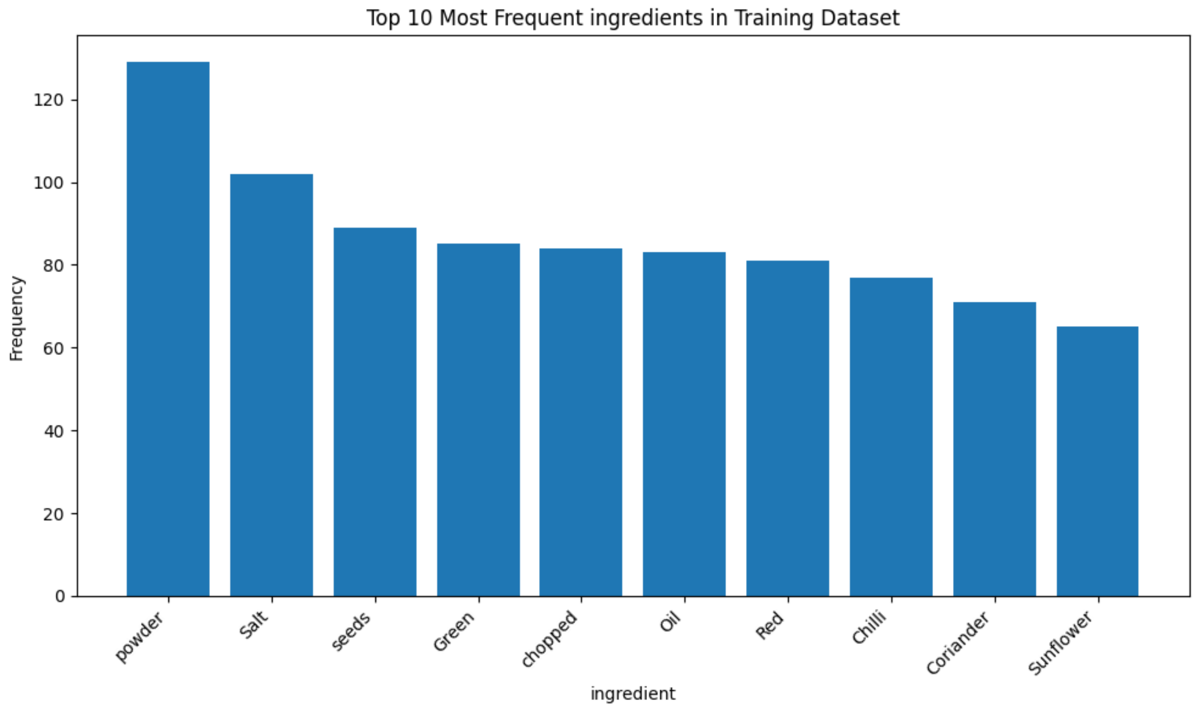
Data Description

The given data is in JSON format, representing a structured recipe ingredient list with Named Entity Recognition (NER) labels. Below is a breakdown of the data fields:

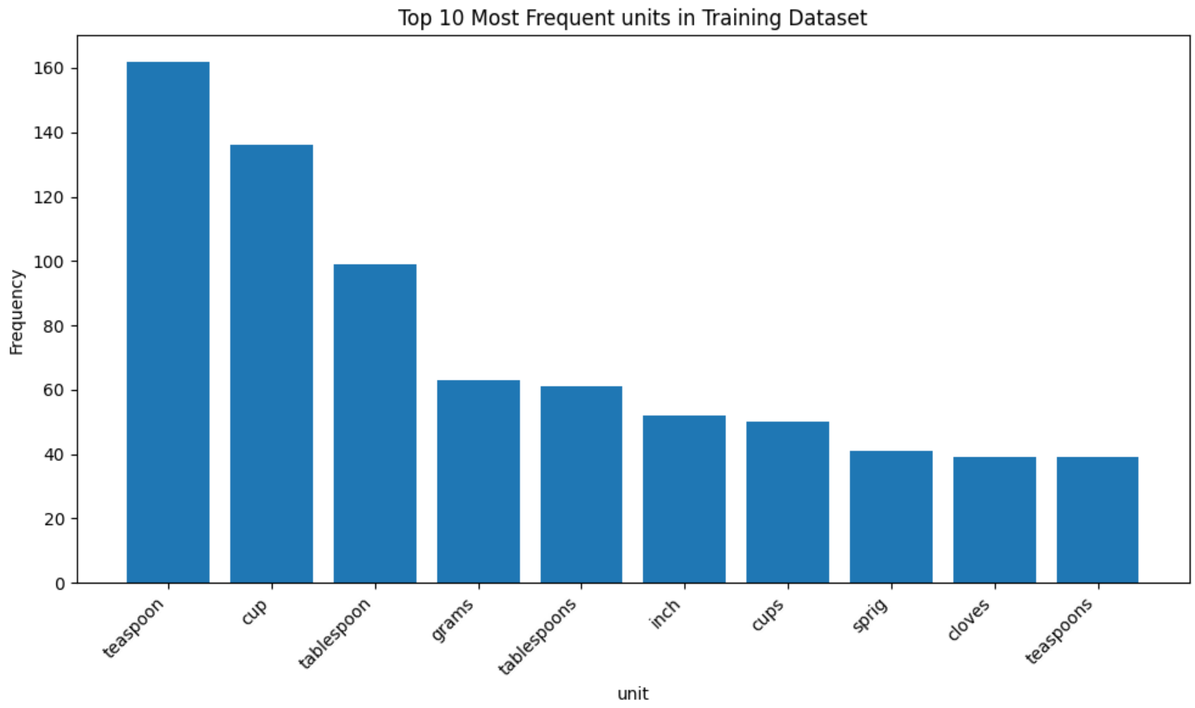
{
"input": "6 Karela Bitter Gourd Pavakkai Salt 1 Onion 3 tablespoon Gram flour besan 2 teaspoons Turmeric powder Haldi Red Chilli Cumin seeds Jeera Coriander Powder Dhania Amchur Dry Mango Sunflower Oil",
"pos": "quantity ingredient ingredient ingredient ingredient ingredient quantity ingredient quantity unit ingredient ingredient ingredient quantity unit ingredient ingredient ingredient ingredient ingredient ingredient ingredient ingredient ingredient ingredient ingredient ingredient ingredient ingredient ingredient ingredient"
}

EDA – Insights

- Top most frequent ingredients



- Top most frequent units



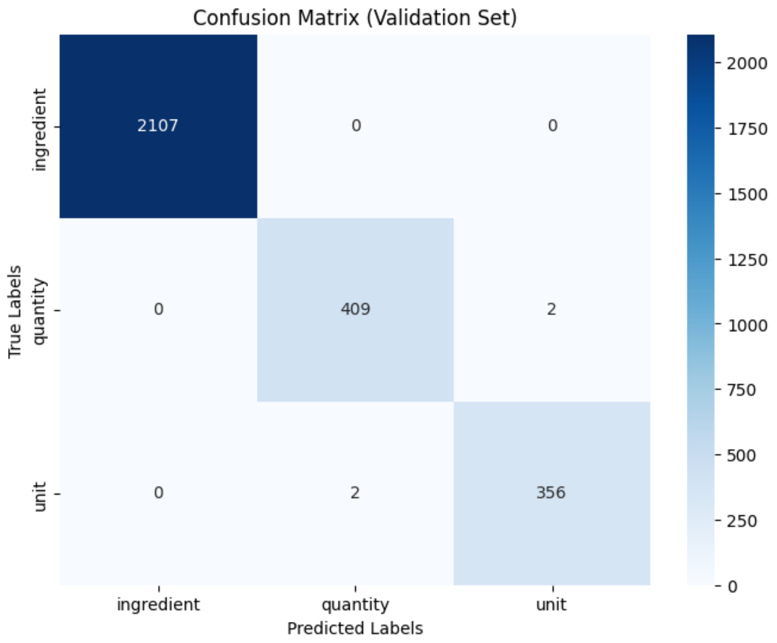
- The most common ingredients are powder and salt.
- "Chopped" and "green" are the most frequent preparation styles.
- The frequent mention of "seeds" indicates a reliance on whole spices (e.g., cumin, mustard).
- Recipes predominantly use teaspoons and cups (almost 50% of unit mentions), followed by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

Model Evaluation

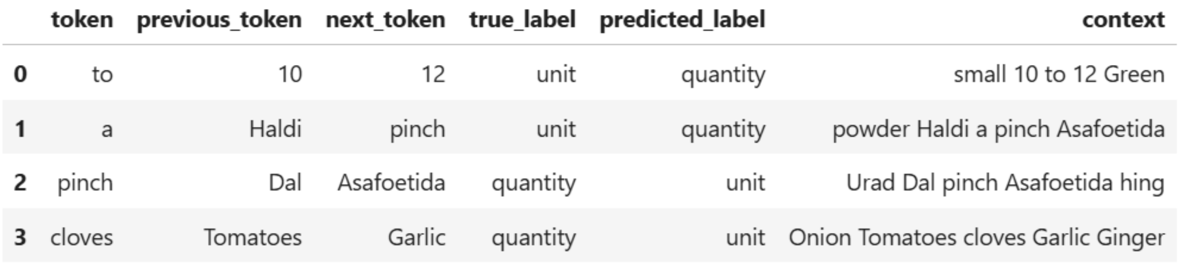
Scores

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	to	10	12	unit	quantity	small 10 to 12 Green
1	a	Haldi	pinch	unit	quantity	powder Haldi a pinch Asafoetida
2	pinch	Dal	Asafoetida	quantity	unit	Urad Dal pinch Asafoetida hing
3	cloves	Tomatoes	Garlic	quantity	unit	Onion Tomatoes cloves Garlic Ginger

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, 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, achieve near-perfect accuracy, suggesting the model effectively identifies it. However, other labels like units and quantity show a slightly lower accuracy, indicating the model might struggle with more complex names or those with ambiguous contexts.
4. Error Analysis Insights

- Examining the error is crucial for identifying patterns in misclassifications. The analysis of misclassified samples reveals that the model occasionally misclassifies unit vs quantities, and vice-versa. Further investigation of those specific tokens and their contexts is needed.
- Contextual ambiguities: Certain quantity might be similar to units (e.g., “pinch”, ‘cloves”).
- Rare ingredients or units: The model may struggle with rare or less-frequent ingredients or units.
- 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 for 'ingredient', likely played a role in the model's performance.
    - The lower weight for 'ingredient' (applied during training) 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.