Identifying Key Entities in Recipe Data

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.

Problem Statement: The goal of this assignment is to identify and classify key entities in cooking recipe data using Named Entity Recognition (NER). Specifically, we aim to extract and correctly label entities such as 'ingredient', 'quantity', and 'unit' from textual recipe instructions. This task is vital for structuring and digitizing recipe data to make it searchable and usable in digital applications.

Assumptions Made:
During the development of the Named Entity Recognition (NER) pipeline for identifying key entities in recipe data, several assumptions were made to simplify the modeling process and focus the scope of the problem. These assumptions influenced data preprocessing, feature extraction, and model design:

- **Entity Classes Are Mutually Exclusive:** Each token is assumed to belong to only one of the three entity classes: 'ingredient', 'quantity', or 'unit'. No overlapping or nested entities are considered.
- **Labels Are Assigned at the Token Level:** The model processes and assigns labels at the token level. Multi-word entities are expected to be captured by a sequence of token-level predictions.
- **Context Is Limited to Neighboring Tokens:** Feature engineering includes context from a fixed number of preceding and following tokens, without considering the entire sentence or paragraph.
- **Class Weights Are Used to Address Imbalance:** To address imbalance in the dataset (e.g., more 'ingredient' tokens), class weights are computed and used in the training process.
- **No External Knowledge Base Is Used:** The model relies solely on patterns learned from the training data, without referencing external databases or lexicons for ingredients or units.
- **Evaluation Ignores Sentence-Level Entity Boundaries:** Evaluation is based on token-level accuracy and confusion matrices. Span-level or partial entity recognition is not accounted for.
- **Noise and Misspellings Are Minimal:** The code assumes a clean dataset without significant typos, OCR errors, or noisy data entries.

Methodology:

The approach involves preprocessing recipe texts, tokenizing them, and then applying NER to classify each token. Used traditional sequence labeling techniques along with feature engineering to build our models. The key stages in our methodology include:

- Importing necessary libraries
    - import sklearn_crfsuite # sklearn-crfsuite is a Py wrapper for CRFsuite (CRF implementation for sequence modeling)
    - import spacy # Library for advanced NLP tasks
    - from sklearn.model_selection import train_test_split
    - from sklearn_crfsuite import metrics # For evaluating CRF models

- Data Ingestion and Preparation
    - Read Recipe Data from Dataframe and prepare the data for analysis
    - df.shape and df.info() DataFrame

```
[8] # print the dimensions of dataframe - df
    df.shape

    (285, 2)

[9] # print the information of the dataframe
    df.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
```

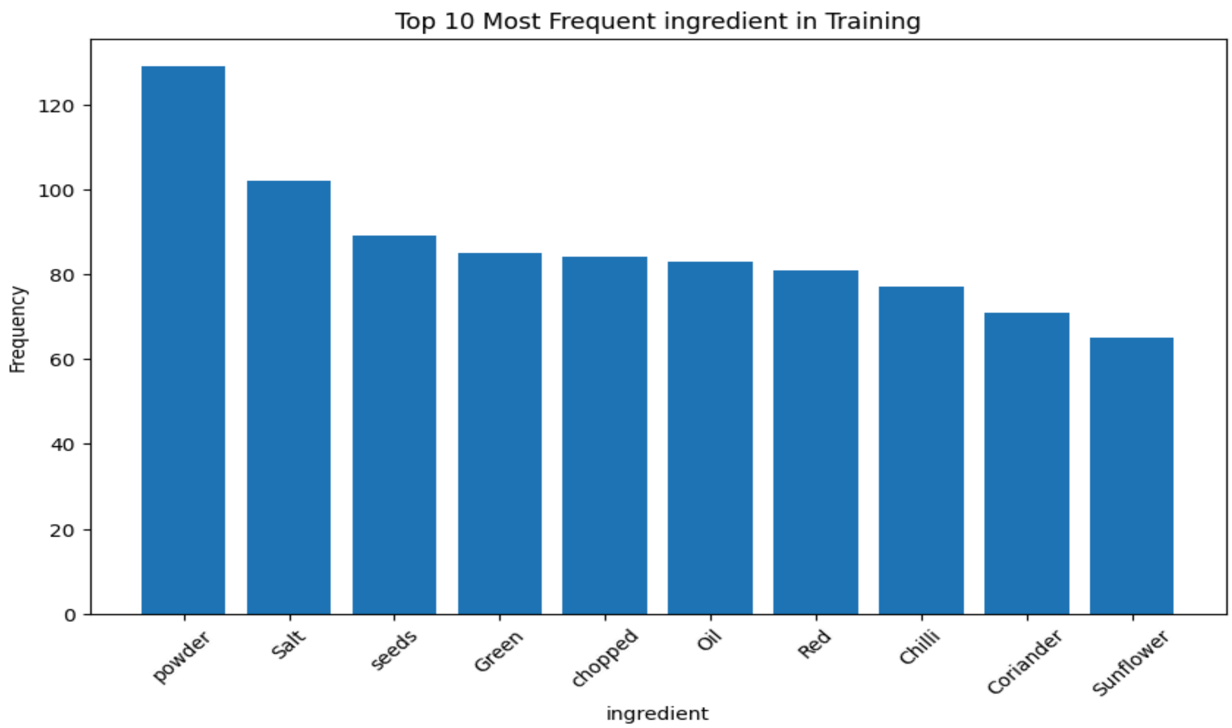
- Check for rows with unequal length of tokens in input and pos and remove them

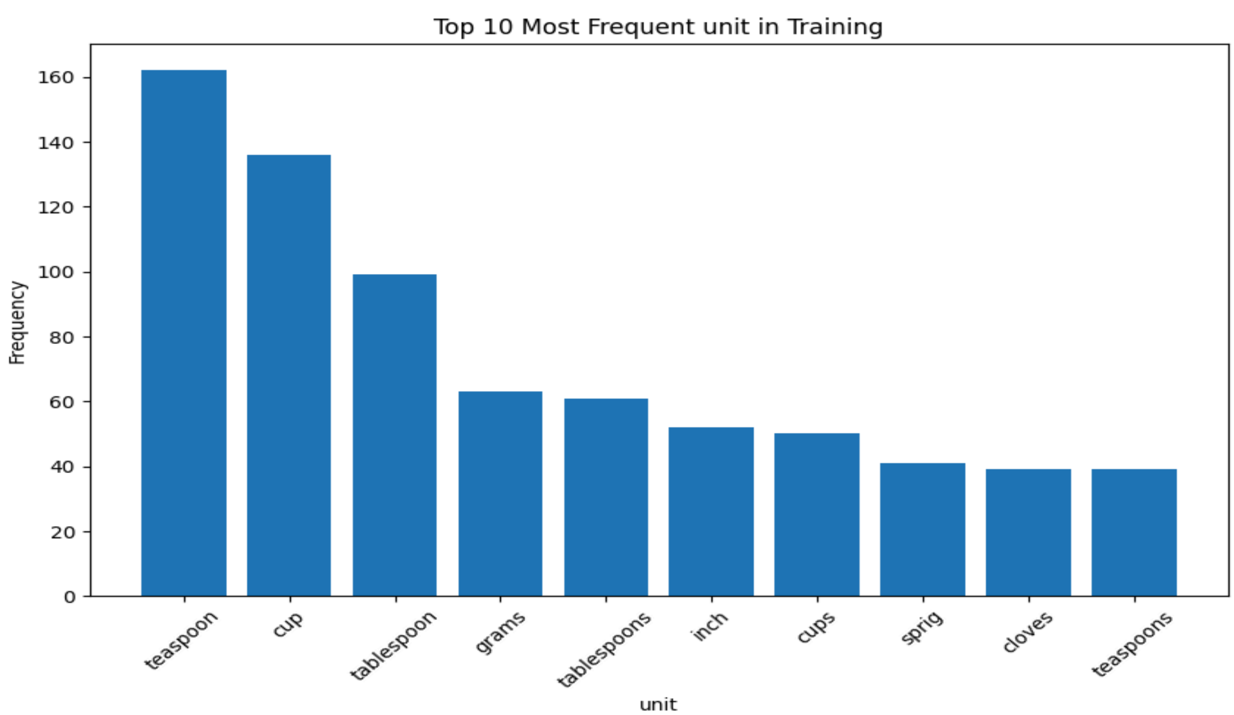
```
[13] # check for the equality of input_length and pos_length in the dataframe
     df[df["input_length"] != df["pos_length"]]
```

	input	pos	input_tokens	pos_tokens	input_length	pos_length
17	2 cups curd 1 cup gourd cucumber green cor coriander 1/2 teaspoon cumin powder salt	quantity unit ingredient quantity unit ingredient ingredient ingredient ingredient quantity unit ingredient ingredient ingredient	[2, cups, curd, 1, cup, gourd, cucumber, green, cor, coriander, 1/2, teaspoon, cumin, powder, salt]	[quantity, unit, ingredient, quantity, unit, ingredient, ingredient, ingredient, ingredient, quantity, unit, ingredient, ingredient, ingredient]	15	14
27	1 Baguette sliced 1 1/2 tablespoon Butter 1/2 Garlic minced cup Spinach Leaves Palak Red Bell pepper Capsicum Tomato finely chopped Onion Black powder Italian seasoning teaspoon Fresh cream Cheddar cheese grated Salt Roasted tomato pasta sauce	quantity ingredient ingredient quantity unit ingredient quantity ingredient ingredient unit ingredient ingredient ingredient ingredient ingredient ingredient ingredient ingredient ingredient ingredient ingredient ingredient ingredient ingredient unit ingredient ingredient ingredient ingredient ingredient ingredient ingredient ingredient ingredient ingredient	[1, Baguette, sliced, 1, 1/2, tablespoon, Butter, 1/2, Garlic, minced, cup, Spinach, Leaves, Palak, Red, Bell, pepper, Capsicum, Tomato, finely, chopped, Onion, Black, powder, Italian, seasoning, teaspoon, Fresh, cream, Cheddar, cheese, grated, Salt, Roasted, tomato, pasta, sauce]	[quantity, ingredient, ingredient, quantity, unit, ingredient, quantity, ingredient, ingredient, unit, ingredient, ingredient, ingredient, ingredient, ingredient, ingredient, ingredient, ingredient, ingredient, ingredient, ingredient, ingredient, ingredient, ingredient, unit, ingredient, ingredient, ingredient, ingredient, ingredient, ingredient, ingredient, ingredient, ingredient, ingredient]	37	36
79	1/2 cup Poha Flattened rice 2 tablespoons Rice flour 2 1/2 liter Milk 1 Nolen Gur or brown sugar Cardamom Elaichi Pods/Seeds 8-10 Mixed nuts almonds/cashews tablespoon Raisins pinch Saffron strands and a little more for garnish Salt	quantity unit ingredient ingredient ingredient quantity unit ingredient ingredient quantity unit ingredient quantity ingredient ingredient ingredient ingredient ingredient ingredient ingredient ingredient quantity ingredient ingredient ingredient unit ingredient unit ingredient ingredient ingredient ingredient ingredient ingredient ingredient ingredient ingredient	[1/2, cup, Poha, Flattened, rice, 2, tablespoons, Rice, flour, 2, 1/2, liter, Milk, 1, Nolen, Gur, or, brown, sugar, Cardamom, Elaichi, Pods/Seeds, 8-10, Mixed, nuts, almonds/cashews, tablespoon, Raisins, pinch, Saffron, strands, and, a, little, more, for, garnish, Salt]	[quantity, unit, ingredient, ingredient, ingredient, quantity, unit, ingredient, ingredient, quantity, unit, ingredient, quantity, ingredient, ingredient, ingredient, ingredient, ingredient, ingredient, ingredient, ingredient, quantity, ingredient, ingredient, ingredient, unit, ingredient, unit, ingredient, ingredient, ingredient, ingredient, ingredient, ingredient, ingredient, ingredient, ingredient]	38	37
164	1/2 cup All Purpose Flour Maida Whole Wheat 1/4 Hung Curd Greek Yogurt 250 grams Chicken minced 1 Spinach Leaves Palak finely chopped Onion 4 cloves Garlic Tomatoes tablespoon Cumin powder Jeera Coriander Powder Dhania 1 1/2 teaspoon Paprika Black pepper 3 sprig Mint Pudina 10 Spring Bulb & Greens 100 Feta Cheese crumbled	quantity unit ingredient ingredient ingredient ingredient ingredient ingredient quantity ingredient ingredient ingredient ingredient quantity unit ingredient ingredient quantity ingredient ingredient ingredient ingredient ingredient ingredient quantity unit ingredient ingredient unit ingredient ingredient ingredient ingredient ingredient ingredient quantity unit ingredient ingredient ingredient quantity unit ingredient ingredient quantity ingredient ingredient ingredient ingredient quantity ingredient ingredient ingredient	[1/2, cup, All, Purpose, Flour, Maida, Whole, Wheat, 1/4, Hung, Curd, Greek, Yogurt, 250, grams, Chicken, minced, 1, Spinach, Leaves, Palak, finely, chopped, Onion, 4, cloves, Garlic, Tomatoes, tablespoon, Cumin, powder, Jeera, Coriander, Powder, Dhania, 1, 1/2, teaspoon, Paprika, Black, pepper, 3, sprig, Mint, Pudina, 10, Spring, Bulb, &, Greens, 100, Feta, Cheese, crumbled]	[quantity, unit, ingredient, ingredient, ingredient, ingredient, ingredient, ingredient, quantity, ingredient, ingredient, ingredient, ingredient, quantity, unit, ingredient, ingredient, quantity, ingredient, ingredient, ingredient, ingredient, ingredient, ingredient, quantity, unit, ingredient, ingredient, unit, ingredient, ingredient, ingredient, ingredient, ingredient, ingredient, quantity, unit, ingredient, ingredient, ingredient, quantity, unit, ingredient, ingredient, quantity, ingredient, ingredient, ingredient, ingredient, quantity, ingredient, ingredient, ingredient]	54	53
207	1 cup Cashew nuts Badam Almond 1 1/4 cups Sugar 1/2 Water teaspoon Cardamom Powder Ghee for greasing	quantity unit ingredient ingredient ingredient ingredient quantity unit ingredient quantity ingredient unit ingredient ingredient ingredient unit ingredient	[1, cup, Cashew, nuts, Badam, Almond, 1, 1/4, cups, Sugar, 1/2, Water, teaspoon, Cardamom, Powder, Ghee, for, greasing]	[quantity, unit, ingredient, ingredient, ingredient, ingredient, quantity, unit, ingredient, quantity, ingredient, unit, ingredient, ingredient, ingredient, unit, ingredient]	18	17

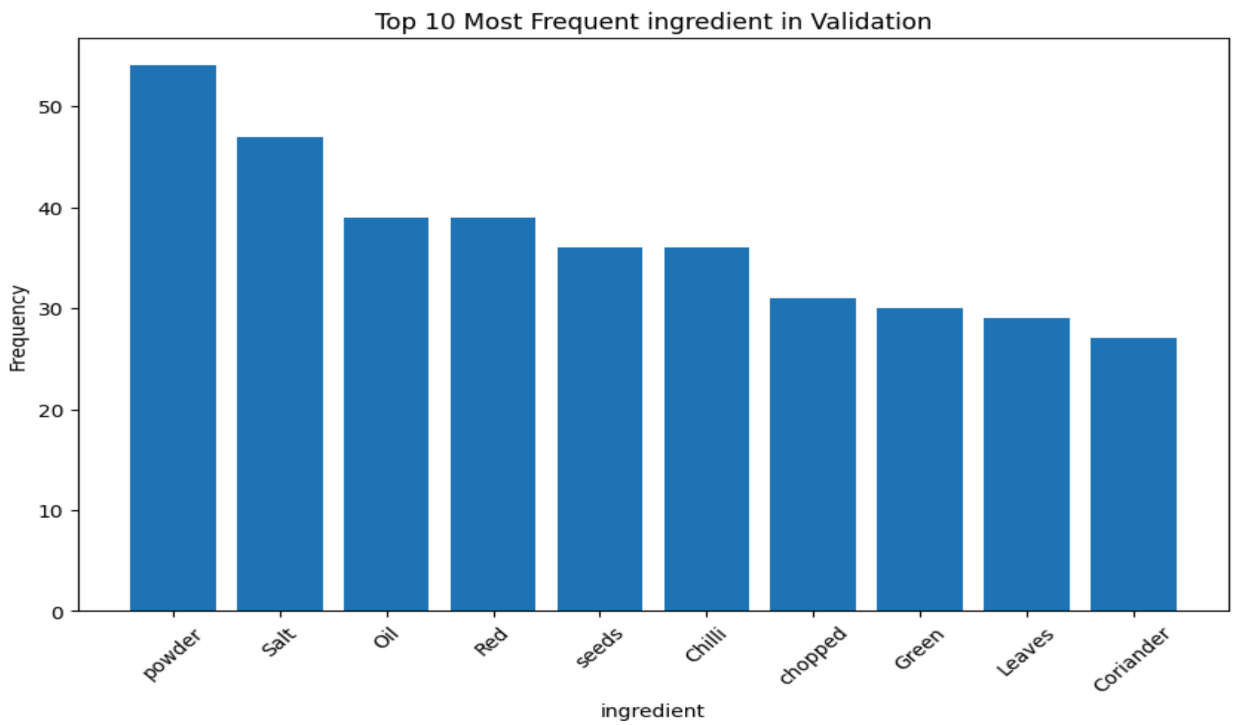
- We can observe that we have only 3 unique pos labels in the recipe: {'unit', 'quantity', 'ingredient'}

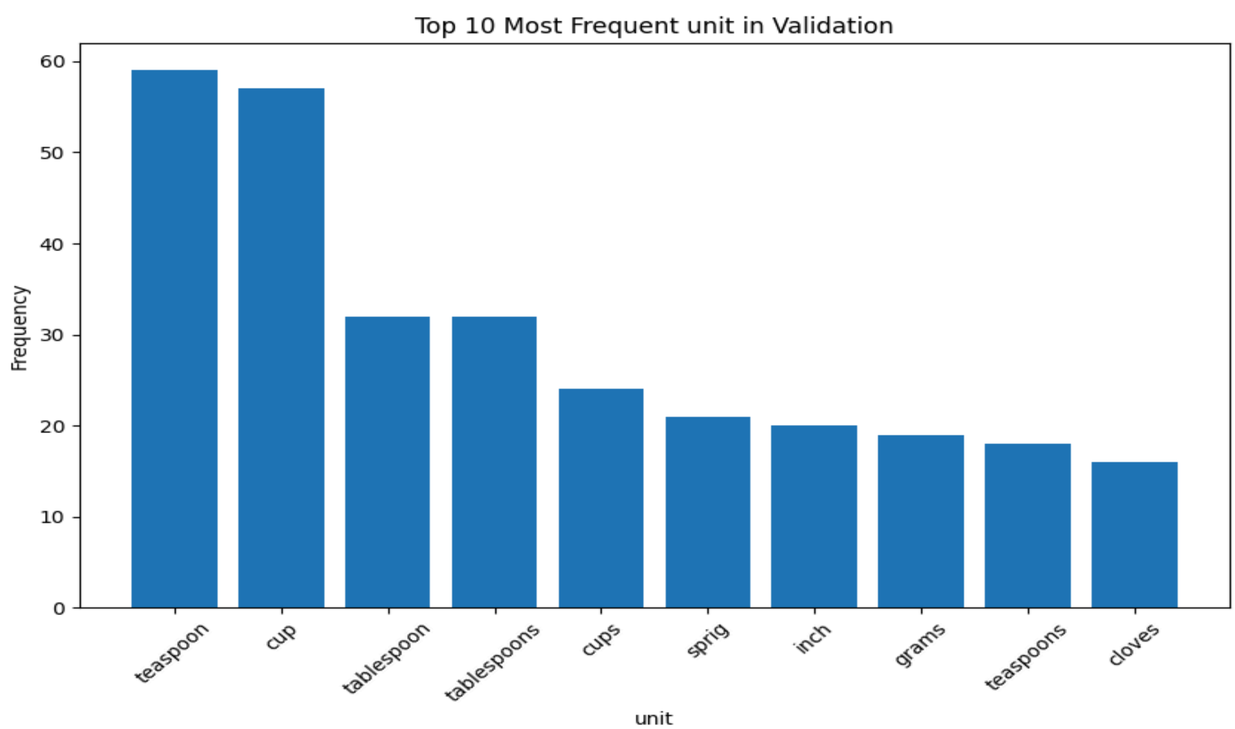
- Train-Validation Split
  - After splitting data into training (70%) and validation (30%) we have: Training samples: 196 and Validation samples: 84
- Exploratory Data Analysis on Training Data
  - Categorizing tokens into labels (unit, ingredient, quantity)
  - List top 10 frequent items in ingredient and unit lists for Training Data





- Exploratory Data Analysis on Validation Data
    - Categorizing tokens into labels (unit, ingredient, quantity)
    - List top 10 frequent items in ingredient and unit lists for Validation Data





- Feature Extraction for CRF Model
  - Preform feature extraction to extract each token from recipe (word2features)
  - Applying weights to feature sets {'quantity': 2.4197 , 'unit': 2.9239 , 'ingredient': 0.4454}
  - Penalising ingredient label with 0.5 reducing weights_dict to {'quantity': 2.4197 , 'unit': 2.9239 , 'ingredient': 0.2227}
- Model Building and Training
  - Initializing CRF model with specified hyperparameters

```
# initialise CRF model with the specified hyperparameters and use weight_dict
crf_model = 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
crf_model.fit(X_train_weighted_features, y_train_labels)
```

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

- Prediction and Model Evaluation
  - Evaluate training data and print classification report

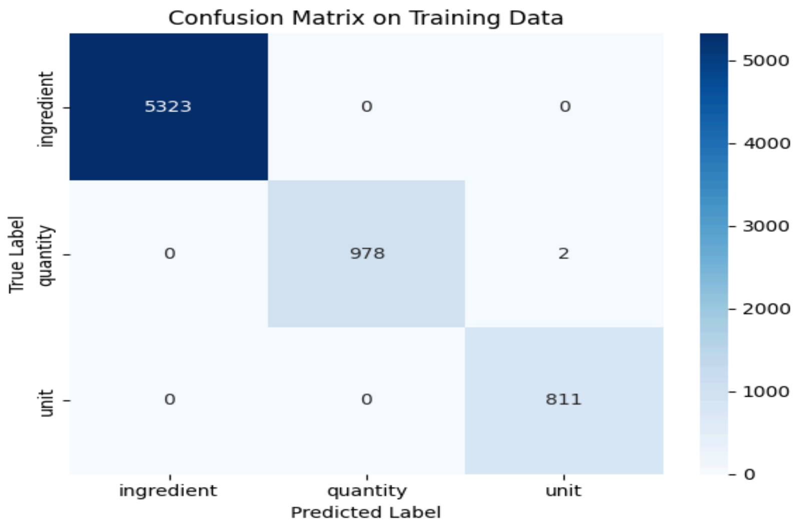
```
[61] # specify the flat classification report by using training data for evaluation
print("Flat classification report by using training data for evaluation")
print(metrics.flat_classification_report(y_train_labels, y_train_pred))
```

```
Flat classification report by using training data for evaluation
              precision    recall  f1-score   support

  ingredient       1.00      1.00      1.00      5323
    quantity       1.00      1.00      1.00       980
        unit       1.00      1.00      1.00       811

    accuracy                           1.00      7114
   macro avg       1.00      1.00      1.00      7114
weighted avg       1.00      1.00      1.00      7114
```

- Display Confusion Matrix on Training Data



- Evaluate validation data and print classification report, from which we can draw that our model has performed incredibly well

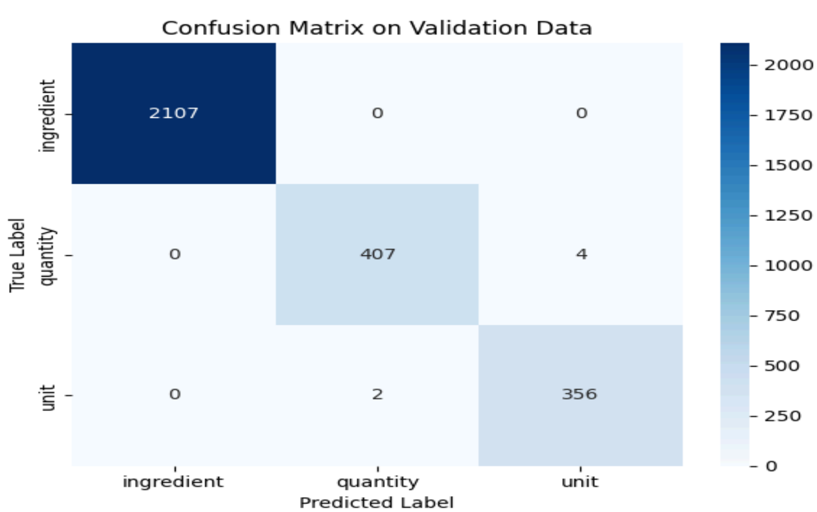
```
[66] # specify flat classification report
     print("Flat Classification Report on Validation Data:")
     print(metrics.flat_classification_report(y_val_labels, y_val_pred))
```

```
Flat Classification Report on Validation Data:
              precision    recall  f1-score   support

  ingredient       1.00      1.00      1.00      2107
    quantity       1.00      0.99      0.99       411
        unit       0.99      0.99      0.99       358

    accuracy                           1.00      2876
   macro avg       0.99      0.99      0.99      2876
weighted avg       1.00      1.00      1.00      2876
```

- Display Confusion Matrix on Validation Data



- Performing Error Analysis on Validation Data
    - Identify misclassified samples in validation dataset

```
Validation Error DataFrame:
    token true_label predicted_label prev_token  next_token  class_weight
0      is   quantity            unit        Pur           2      2.419728
1     for   quantity            unit        Oil    kneading      2.419728
2      to       unit        quantity         10          12      2.923962
3       a       unit        quantity      Haldi       pinch      2.923962
4   pinch   quantity            unit        Dal  Asafoetida      2.419728
5  cloves   quantity            unit   Tomatoes      Garlic      2.419728
```

    - Ingredient label predictions are perfect, however its low class_weight indicates it’s the most frequent label in the dataset, which might result in a biased model or overfitting for this class.
    - Our model only made 4 errors in predicting quantity labels and 2 errors for unit labels. Despite their class_weights (2.42 and 2.87), helped the model learn to prioritize them.

Conclusion:

- With all classes achieving over 99% accuracy, the model demonstrates excellent performance and generalization on validation set
- For further analysis consider
    - Cross-validation for model’s robustness
    - Monitoring performance when scaling to new recipes