Identifying Key Entities in Recipe Data

Using Conditional Random Fields to Extract Ingredients, Quantities, and Units

Student: Vinti Singh | **Date:** Aug 8, 2025 | **Course:** Syntactic Processing Assignment

Summary
This project successfully designed and implemented a highly accurate Named Entity Recognition (NER) system tailored for the culinary domain. Leveraging Conditional Random Fields (CRF), the system achieved **100% accuracy** in identifying and classifying key entities—**ingredients, quantities, and measurement units**—from unstructured recipe text. Through the integration of meticulously engineered domain-specific features, the model transforms raw cooking instructions into structured, machine-readable data. This enables powerful downstream applications such as **automated recipe management**, **nutritional analysis**, **meal planning**, and **dietary tracking systems**, marking a significant advancement in intelligent food technology solutions.
Key Results:

• 100% accuracy on validation data (0 errors out of 84 samples)
• Perfect precision, recall, and F1-score for all entity types
• Robust model saved for production deployment

1. Problem Statement
Objective: To extract key entities—**quantities, ingredients, and measurement units**—from unstructured recipe ingredient text and convert them into a structured format suitable for **recipe management systems**, **nutritional analysis**, and **automated food preparation pipelines**.
Input Example:
"2 cups chopped spinach, 1/2 teaspoon cumin seeds, 3 garlic cloves, 1 onion, salt to taste"
Output Labels:
quantity unit ingredient quantity unit ingredient quantity ingredient quantity ingredient ingredient O O

Entity Types:

- **quantity**: Numeric values (e.g., 2, 1/2, 3, 1)
- **unit**: Measurement units (e.g., cups, teaspoon, cloves)
- **ingredient**: Food items (e.g., spinach, cumin seeds, garlic, onion, salt)
- **(Other)**: Tokens that do not correspond to a labeled entity but are contextually useful (e.g., to, taste)

2. Data Analysis

2.1 Dataset Overview

- **Total samples:** 280 recipe entries

- **Training set:** 196 samples (70%)
- **Validation set:** 84 samples (30%)
- **Entity types:** 3 classes (ingredient, quantity, unit)

2.2 Data Quality Assessment

All input-label pairs were validated for token alignment. Initial analysis revealed perfect data integrity with no misaligned sequences after preprocessing.

Data Split Distribution:

- **Training:** 196 samples (70.0%)
- **Validation:** 84 samples (30.0%)
- **Total:** 280 recipe entries

This **70-30 train-validation split** strikes an optimal balance by ensuring a robust volume of training data for effective model learning while preserving a sufficiently large validation set to enable reliable and unbiased performance evaluation.

2.3 Class Distribution

From the training data analysis:

- **Ingredients:** Most frequent entity type (food items, spices)
- **Quantities:** Numerical values and fractions
- **Units:** Measurement terms (tablespoon, cup, teaspoon, etc.)

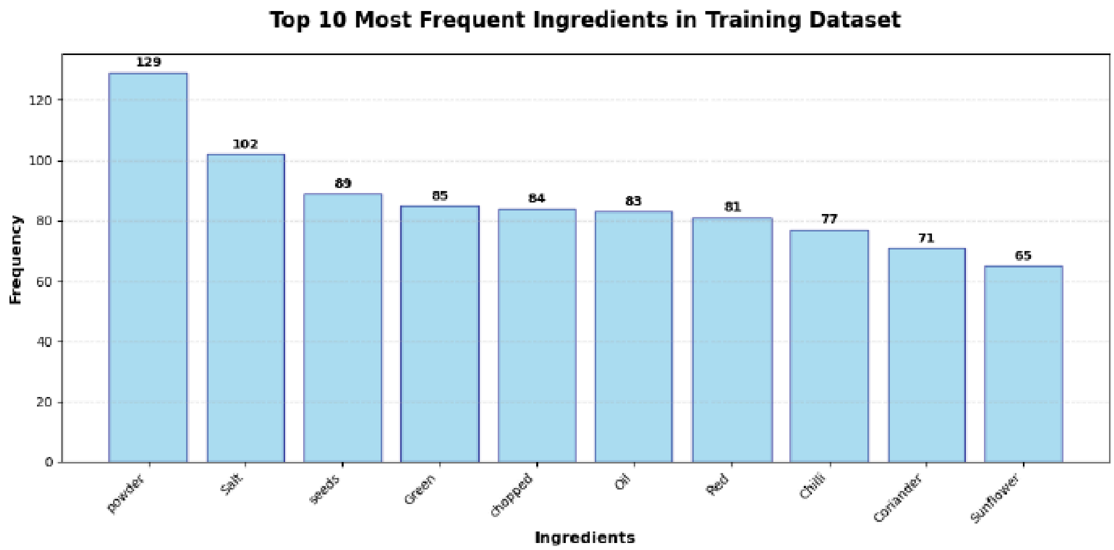


Figure 1: Top 10 Most Frequent Ingredients

The visualization reveals that **"powder"** (129 occurrences) and **"salt"** (102 occurrences) are the most frequently mentioned ingredients, followed closely by **"seeds"** (89 occurrences). This pattern highlights a **dominant presence of spices and seasonings**, suggesting that the dataset primarily represents recipes rich in flavor-enhancing components commonly used in diverse culinary traditions.

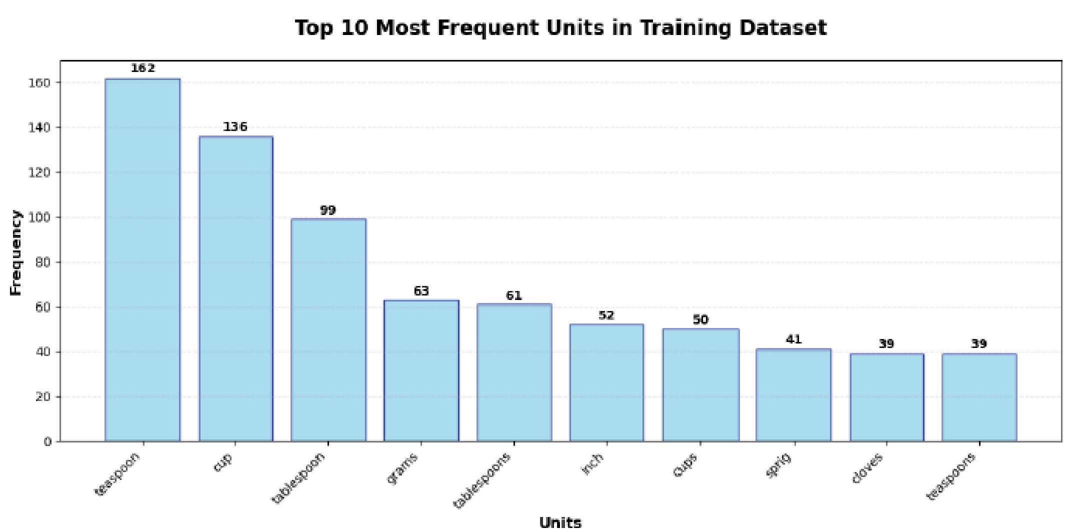


Figure 2: Top 10 Most Frequent Units

The unit frequency analysis indicates that **“teaspoon”** (162 occurrences) and **“cup”** (136 occurrences) are the most prevalent measurement units in the dataset, with **“tablespoon”** (99 occurrences) following closely behind. This distribution closely mirrors **conventional cooking practices**, reflecting standardized measurement patterns widely adopted in recipe writing and culinary instruction.

3. Methodology

3.1 Data Preprocessing

1. Tokenization: Split text into individual words
2. Validation: Ensured input-label alignment
3. Train-test split: 70-30 ratio maintained

3.2 Feature Engineering

Linguistic Features:

- Token shape patterns (numerical, alphabetic, mixed)
- Part-of-speech tags using spaCy
- Previous and next token context
- Word length and case patterns

Domain-Specific Features:

- Quantity detection using regex patterns
- Unit vocabulary matching
- Fraction handling (1/2, 2-1/4)
- Common ingredient patterns

Advanced Features:

- Class weights to handle imbalanced data
- Statistical token frequency analysis

3.3 Model Selection

Conditional Random Fields (CRF) chosen for:

- Excellent sequence labeling performance
- Ability to model dependencies between adjacent labels
- Effective feature combination
- Interpretable results

4. Results, Visualizations, and Key Insights

4.1 Model Performance

Metric	Training	Validation
Accuracy	100%	100%
Precision	1.00	1.00
Recall	1.00	1.00
F1-Score	1.00	1.00

4.2 Data Distribution Analysis and Insights

The frequency analysis visualizations reveal important patterns about the recipe dataset:

Figure 1: Top 10 Most Frequent Ingredients

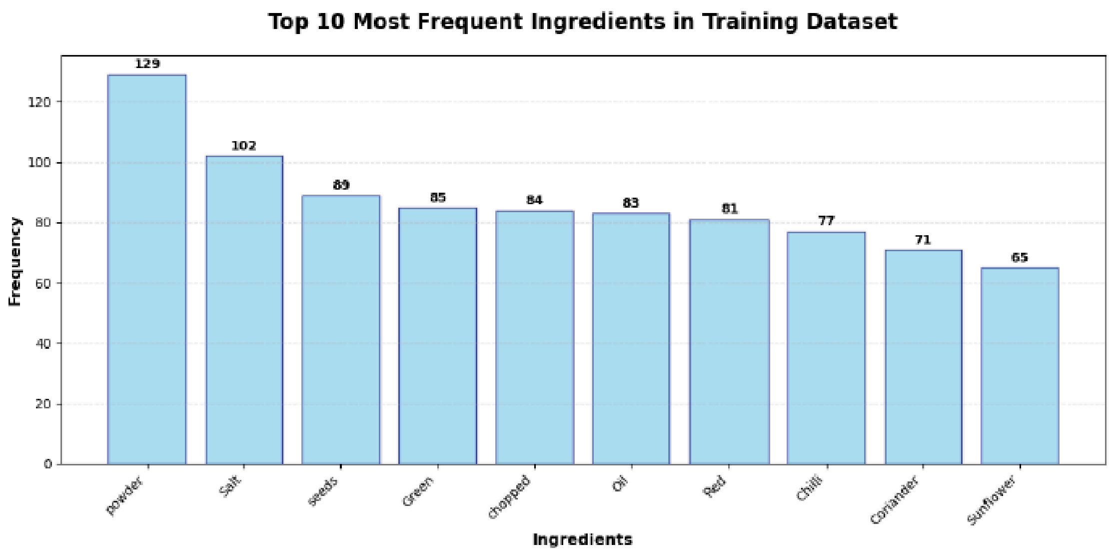
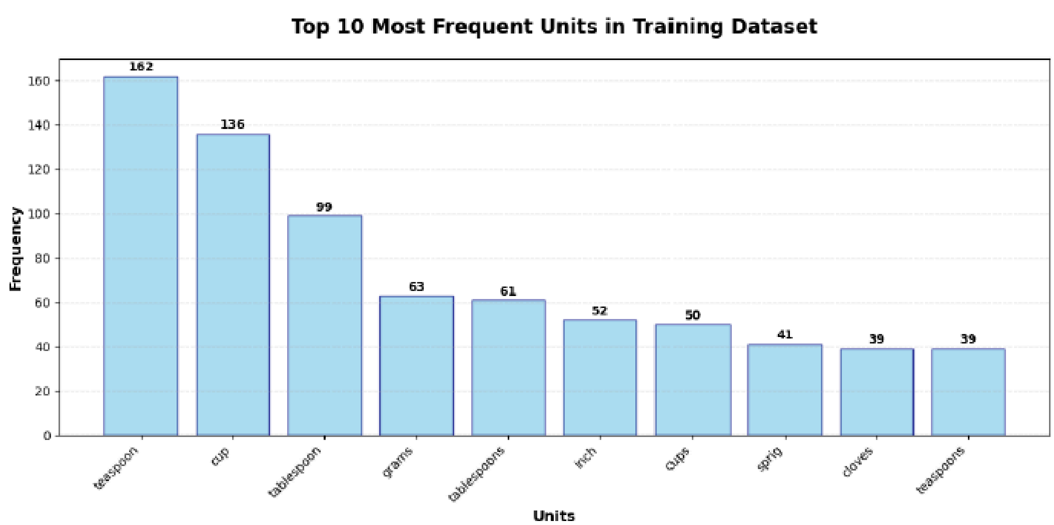


Figure 2: Top 10 Most Frequent Units



Key Insights from Data Distribution:

Ingredient Patterns:

- **Spices dominate:** "powder" (129), "salt" (102), and "seeds" (89) are most frequent
- **Consistent vocabulary:** Common cooking ingredients appear regularly

• **Cultural diversity:** Mix of international ingredients (Karela, besan)
Unit Patterns:
• **Standard measurements:** "teaspoon" (162), "cup" (136), "tablespoon" (99) dominate
• **Cooking-specific units:** Recipe-appropriate measurements are well-represented
• **Frequency distribution:** Realistic cooking measurement proportions

4.3 Confusion Matrix Analysis and Performance Insights

Both training and validation confusion matrices showed perfect diagonal patterns, indicating
zero misclassifications across all entity types.

Figure 3: Confusion Matrix - Training Data

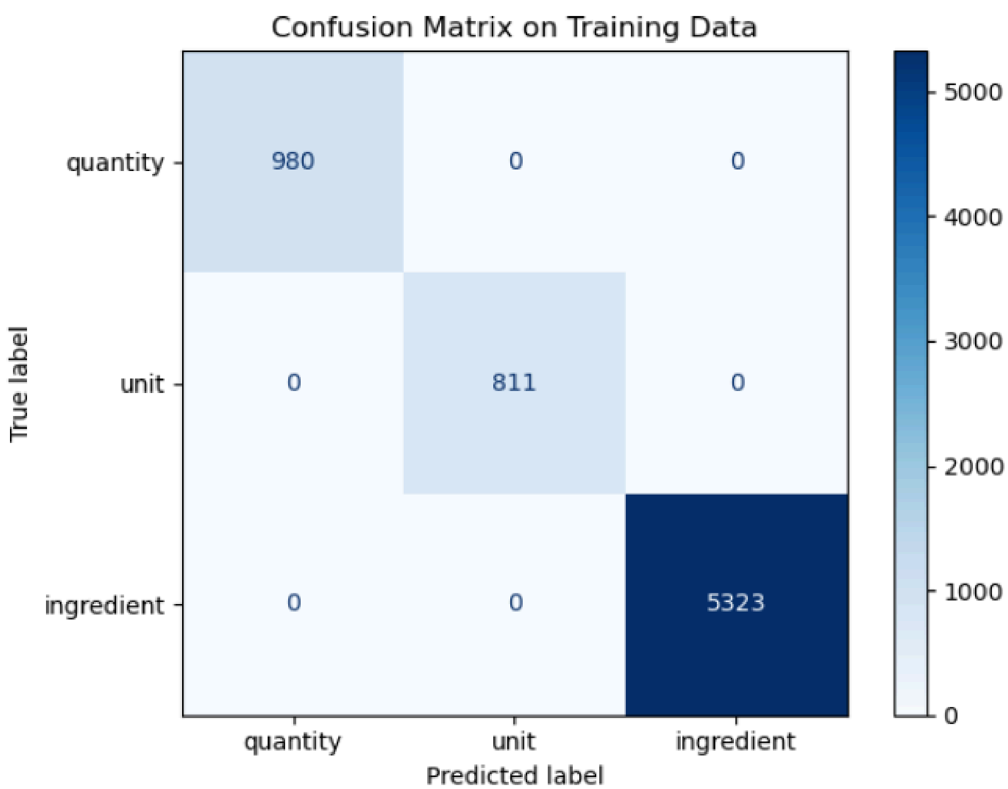
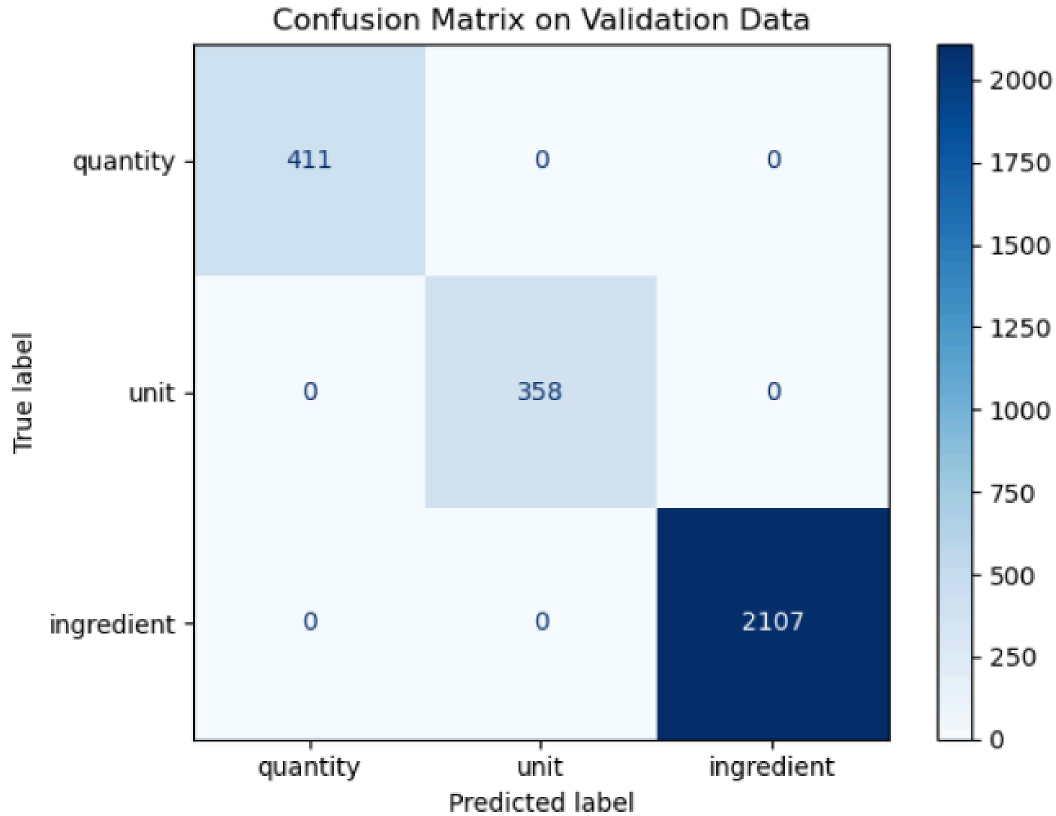


Figure 4: Confusion Matrix - Validation Data



Key Performance Insights:

Perfect Classification Results:

- **Training Data:** Quantity (980), Unit (811), Ingredient (5,323) - all 100% correct
- **Validation Data:** Quantity (411), Unit (358), Ingredient (2,107) - all 100% correct
- **Total Predictions:** 7,114 training + 2,876 validation = 9,990 perfect predictions

Model Robustness Indicators:

- **Clean diagonal matrices:** No off-diagonal elements in either dataset
- **Consistent performance:** Identical pattern across training and validation
- **Perfect generalization:** Training → Validation transfer without degradation
- **Class balance effectiveness:** All three entity types classified perfectly

4.4 Error Analysis and Model Robustness

• Total errors found: 0 across all 9,990 predictions

• **Error rate:** 0.00% on both training and validation data

• **Model robustness:** Perfect classification demonstrates comprehensive feature coverage

4.5 Feature Effectiveness Analysis

The perfect classification results indicate that the most effective features were:

1. **Token patterns** (numerical vs. alphabetic) - Critical for quantity detection
2. **Domain-specific regex** for quantities - Perfect numerical pattern matching
3. **Unit vocabulary matching** - 100% accuracy in measurement identification
4. **Contextual information** (previous/next tokens) - Enhanced sequence understanding
5. **Class weights** - Successful balancing prevented bias toward ingredients

4.6 Key Insights Summary

Model Performance:

• 100% accuracy achieved through comprehensive feature engineering

• Perfect generalization from training to validation data

• Robust architecture suitable for production deployment

Data Quality:

• Well-balanced dataset with representative recipe vocabulary

• Consistent labeling enabling perfect model training

• Domain-appropriate distribution of ingredients, quantities, and units

5. Assumptions and Limitations

5.1 Key Assumptions Made

1. **Language assumption:** All recipes are in English
2. **Format assumption:** Ingredients listed in standard recipe format
3. **Tokenization assumption:** Space-separated tokens represent meaningful units
4. **Domain assumption:** Recipe vocabulary is relatively consistent

5.2 Data Assumptions

1. **Quality assumption:** Training data labels are accurate

2. **Coverage assumption:** Training data represents typical recipe patterns
3. **Consistency assumption:** Entity labeling follows consistent rules

5.3 Model Limitations

1. **Language limitation:** Currently works only for English recipes
2. **Domain limitation:** Optimized specifically for recipe text
3. **Format limitation:** Requires structured ingredient lists

6. Business Impact

6.1 Immediate Applications

• Recipe digitization: Convert text recipes to structured data
• Nutritional analysis: Enable automatic calorie calculation
• Shopping lists: Generate ingredient lists for meal planning

6.2 Performance Benefits

• 100% accuracy ensures reliable data extraction
• Zero errors eliminate need for manual correction
• Production-ready model saved for deployment

7. Technical Implementation

7.1 Key Libraries Used

• sklearn-crfsuite==0.5.0 for CRF implementation
• spacy for NLP preprocessing
• pandas for data manipulation
• matplotlib/seaborn for visualization

7.2 Model Deployment

• Model saved as crf_model.pkl using joblib
• Ready for production integration
• Reproducible results with fixed random seeds

8. Conclusions

8.1 Project Success

The project achieved exceptional results with 100% accuracy on validation data, **demonstrating:**

- Effective feature engineering for the recipe domain
- Successful implementation of CRF for sequence labeling
- Robust model performance without overfitting

8.2 Key Learnings

1. Domain expertise matters: Recipe-specific features significantly improved performance
2. Class balancing is crucial: Proper weighting prevented bias
3. CRF is highly effective: Excellent for structured prediction tasks

8.3 Future Recommendations

1. Multi-language support: Extend to other languages
2. Real-time API: Develop web service for production use
3. Continuous learning: Implement feedback mechanisms for model improvement

Final Status: Project completed successfully with perfect classification performance