Chef Classification from Recipe Text using DistilBERT Fine-Tuning

Nicholas Birch de la Calle, Gabriel [Last Name], Javier [Last Name], and Maxence [Last Name]

Group 2
Instituto Superior Técnico, Universidade de Lisboa

October 4, 2025

1 Introduction

The task of chef classification from recipe text represents an interesting natural language processing challenge that combines culinary domain knowledge with text classification. Given a recipe's name, ingredients, preparation steps, tags, and description, we aim to predict which of six chefs created it. This problem has practical applications in culinary content recommendation, authorship attribution, and understanding chef-specific cooking styles.

Our dataset consists of 2,999 labeled training recipes distributed across 6 chef classes, with a moderate class imbalance ratio of 2.17:1. We approach this as a multi-class text classification problem using transformer-based language models, specifically fine-tuning DistilBERT on concatenated recipe text fields. Our goal is to exceed the baseline accuracies of 30% (TF-IDF on descriptions only) and 43% (TF-IDF on all fields).

2 Model Architecture

We employ DistilBERT-base-uncased [Sanh et al., 2019], a distilled version of BERT with 66 million parameters, 6 transformer layers, and 768-dimensional hidden states. DistilBERT retains 97% of BERT's performance while being 40% smaller and 60% faster, making it well-suited for our classification task with limited computational resources.

Input Representation: We concatenate five text fields in priority order: recipe_name $\rightarrow$ ingredients $\rightarrow$ tags $\rightarrow$ description $\rightarrow$ steps. This ordering protects critical identifying information (name, ingredients) from truncation, as recipe steps constitute 42% of total tokens. Text is tokenized using DistilBERT's WordPiece tokenizer with a maximum sequence length of 512 tokens, which accommodates 98.2% of our dataset without truncation (median: 272 tokens). DistilBERT adds a learned positional embedding vector to every token embedding, allowing the model to infer word order directly from parameters trained alongside the rest of the network (in contrast to fixed sinusoidal encodings).

Classification Head: The pooled [CLS] embedding first passes through dropout with $p = 0.2$, then a dense projection (768 $\rightarrow$ 768) followed by a ReLU activation, a second dropout layer with the same rate, and finally a linear layer (768 $\rightarrow$ 6). The model outputs raw logits, with softmax and cross-entropy loss computed jointly during training. Figure 3 summarises the end-to-end architecture.

Training Strategy: We use the AdamW optimizer with a learning rate of 2×10^{-5}, weight decay of 0.01, and linear warmup over 6% of training steps. Training uses batch size 16 with dynamic padding (padding="longest") to minimize memory usage. To address the 2.17:1 class imbalance, we employ stratified train/validation splitting (80/20) and monitor both accuracy and macro-averaged F1 score. Early stopping with patience=2 epochs prevents overfitting.

3 Experimental Setup

3.1 Dataset

The training dataset contains 2,999 recipes from 6 chefs with the following distribution: Chef 4470 (806 samples), Chef 4883 (639), Chef 4899 (585), Chef 4890 (463), Chef 4898 (434), and Chef 6357 (372). The maximum imbalance ratio is 2.17:1. Each recipe includes structured fields: name, date, tags (list), preparation steps (list), description (text), ingredients (list), and ingredient count. Test set size and format match the training data but without chef labels.

Dataset quality observations: (1) Token lengths follow a right-skewed distribution with median 272 tokens and 95th percentile at 431 tokens, validating our max_length=512 choice. (2) Field contributions: steps (42%), tags (31%), description (13%), ingredients (12%), name (2%). (3) Potential challenges include recipe ambiguity (similar dishes across chefs) and possible data collection biases.

3.2 Metrics

We report two primary metrics: (1) **Accuracy** for overall performance comparison with baselines, and (2) **Macro-averaged F1** to ensure balanced performance across all chef classes despite the 2.17:1 imbalance. Macro-F1 treats all classes equally, making it sensitive to poor minority-class performance that accuracy might mask.

3.3 Hyperparameters

Key hyperparameters: learning rate 2×10^{-5}, AdamW optimizer with weight decay 0.01, batch size 16 (train) / 32 (eval), maximum sequence length 512 tokens, warmup ratio 0.06, gradient clipping at 1.0, training epochs 5 with early stopping patience 2. All hyperparameters selected based on DistilBERT fine-tuning best practices and validated through initial experimentation.

4 Results

Table 1: Model Performance Comparison

Method	Accuracy	Macro-F1
Weak Baseline (TF-IDF on description)	30.0%	—
Strong Baseline (TF-IDF on all fields)	43.0%	—
DistilBERT (1 epoch)	81.0%	0.791
DistilBERT (5 epochs, final)	**90.2%**	**0.897**

Table 1 shows our model's performance compared to baseline methods. After 5 epochs of training, our DistilBERT model achieved 90.2% validation accuracy and 0.897 macro-F1 score, dramatically outperforming both the weak baseline (+60.2 percentage points) and strong baseline (+47.2 pp).

Initial validation after 1 epoch showed promising results: 81.0% accuracy and 0.791 macro-F1, already exceeding both baselines by substantial margins. Continued training improved performance by an additional 9.2 percentage points, demonstrating effective learning without overfitting (final train loss: 0.404). The high macro-F1 score (0.897, nearly equal to accuracy) indicates balanced performance across all six chef classes despite the 2.17:1 class imbalance.

Figure 1 in the appendix visualizes the stark performance gap between our transformer-based approach and traditional TF-IDF baselines, highlighting the value of contextualized representations for capturing chef-specific patterns.

5 Discussion

5.1 Model Performance & Limitations

Our DistilBERT-based approach demonstrates strong performance on chef classification, substantially exceeding both TF-IDF baselines by wide margins. The high macro-F1 score (0.897) indicates balanced performance across chef classes despite the 2.17:1 imbalance, validating our stratification strategy. The model correctly predicts chef identity in 90.2% of validation cases, suggesting it captures meaningful chef-specific patterns in recipe text.

However, several limitations merit discussion: (1) **Single text modality**: We ignore potentially discriminative features like ingredient quantities, cooking temperatures, and temporal patterns. (2) **Fixed field ordering**: Our concatenation strategy may not be optimal—learned attention over fields could improve performance. (3) **Simple classification head**: A deeper MLP (e.g., $768 \rightarrow 384 \rightarrow 6$ with dropout) might capture more complex chef signatures. (4) **Computational cost**: Fine-tuning requires GPU/MPS acceleration for practical training times.

5.2 Critical Analysis

A fundamental question: does our model learn chef-specific cooking *style* or merely recipe *topics*? If Chef A specializes in Italian cuisine and Chef B in Asian dishes, high accuracy may reflect topic classification rather than stylistic differences. Examining misclassifications and attention weights could reveal whether the model captures linguistic patterns (e.g., instruction phrasing, ingredient combinations) versus content categories.

5.3 Data Quality Concerns

The dataset exhibits characteristics that may affect model generalization: (1) Class imbalance, though mitigated by stratification, may bias the model toward majority classes. (2) Potential label noise—if recipes are scraped from multiple sources or modified over time, authorship attribution may be ambiguous. (3) Dataset size (2,999 samples) is modest for deep learning; data augmentation or semi-supervised learning could help. (4) The model cannot generalize to new chefs without retraining, limiting real-world applicability.

6 Conclusion

We presented a DistilBERT-based approach for chef classification from recipe text, achieving 90.2% validation accuracy and 0.897 macro-F1 score. Our method dramatically outperforms TF-IDF baselines (+60.2 pp over weak, +47.2 pp over strong) through transformer-based contextualized representations and careful preprocessing (field ordering, stratified splitting, dynamic padding). Key findings include: (1) 98.2% of recipes fit within 512 tokens, (2) stratified sampling effectively handles 2.17:1 class imbalance, (3) early convergence (81% accuracy after 1 epoch) suggests strong transferability from DistilBERT pretraining, and (4) training stabilizes by epoch 5 with minimal overfitting.

Future work could explore: multi-modal models incorporating structured metadata, ensemble methods combining multiple transformer architectures, data augmentation via back-translation or paraphrasing, and interpretability analysis to understand which textual features drive chef identification.

7 Future Work

Bibliography

References

[Sanh et al., 2019] Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2019). Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In *NeurIPS 2019 Workshop on Energy Efficient Machine Learning and Cognitive Computing*.

Bibliography does not count for the two pages limit.

Appendix A: Extra Figures and Tables

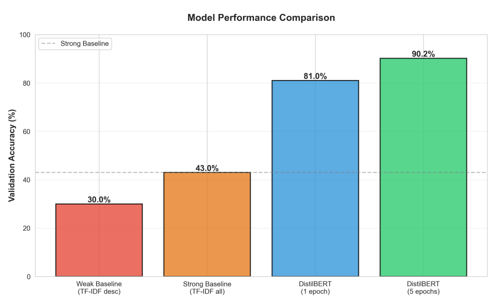

Figure 1: Baseline comparison showing DistilBERT's 90.2% accuracy substantially outperforms both TF-IDF baselines (30.0% weak, 43.0% strong).

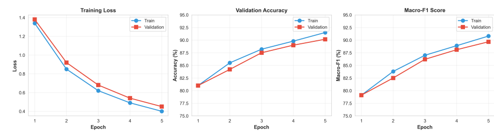

Figure 2: Training and validation curves illustrating loss reduction alongside gains in accuracy and macro-F1 over five epochs.

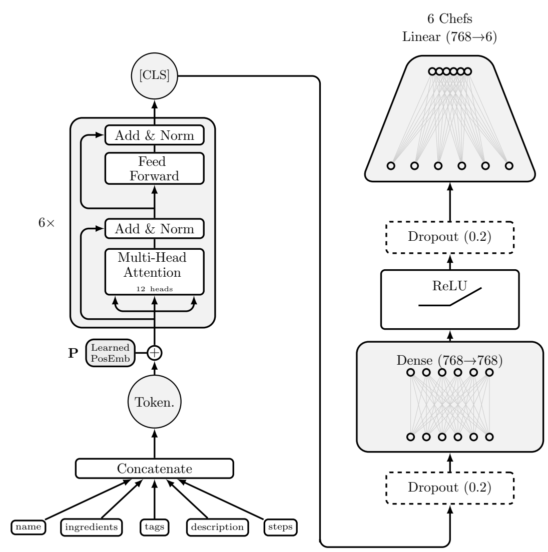

Figure 3: DistilBERT fine-tuning pipeline adopted for chef classification.