Task 3: Finding Indian Dishes

Dheer Gupta

May 2025

1 Overview and Dataset

In Task 3, I built a dish recognizer to discover popular Indian dishes from Yelp reviews. The two main subtasks were:

1. **Manual Tagging (Task 3.1)**: Refine a given list of 142 candidate phrases by flipping or removing mislabelled items and adding missing dishes.
2. **Mining Additional Dish Names (Task 3.2)**: Expand this cleaned list by mining a sample of 5,000 reviews using pattern-based regex rules to find new dish phrases.

All inputs came from:

- `manualAnnotationTask/Indian.label`: the initial candidate list (142 lines of phrase ¯label).
- Yelp Academic Dataset: `business.json` (to select Indian restaurants) and `review.json` (to fetch 5,000 reviews).

2 Task 3.1: Manual Tagging

The original `Indian.label` contained 142 phrases; each was verified (label 1) or unverified (label 0). Many non-dishes (e.g. taj mahal, mother india, mount everest) had made it into the 1 bucket, while some true dishes (e.g. tikka masala, naan, dal) were missing or mislabelled. I performed the following actions:

- **Removed false positives**: Dropped any non-food term (e.g. taj mahal 0 remained with label 0, india gate 0 stayed 0).
- **Flipped false negatives**: Changed tikka masala 0→tikka masala 1, naan (missing)→added naan 1.
- **Added missing dishes**: Inserted staples such as dal 1, samosa 1, biryani 1, paneer 1, lassi 1, masala 1.

After cleaning:

- Originally labelled dish (1): 48 phrases.
- Final labelled dish (1): 27 phrases.

All 27 verified dish names were saved in `Indian_cleaned.label`:

`basmati rice, biryani, brown rice, chick peas, chicken tikka, chicken tikka masala, chicken wings, curry, dal, flat bread, fried rice, gluten free, gulab jamun, lassi, masala, naan, paneer, rice pudding, rogan josh, samosa, south india, street food, tandoori chicken, tikka masala, tomato sauce, tomato soup, white rice`

I confirmed that removing mislabeled items (e.g. mount everest 0) and flipping missing dishes greatly reduced noise. A summary file `summary.txt` logged Before: 48 dishes → After: 27 dishes.

3 Task 3.2: Mining Additional Dish Names

3.1 Review Sampling

Using `Task3.py`, I:

1. Filtered `business.json` for Indian restaurants (202 listings).
2. Pulled 5,000 associated reviews from `review.json` (random sampling).

3.2 Pattern-Based Extraction

To discover dishes missing from the manual list, I concatenated all review text (lowercase) and applied regex patterns such as:

- `rordered (?:the )?([a-z][a-z]2,20?)(?:+(?:and|with|was|is|[.,]))`
- `rtried (?:the )?([a-z][a-z]2,20?)(?:+(?:and|with|was|is|[.,]))`
- `rhad (?:the )?([a-z][a-z]2,20?)(?:+(?:and|with|was|is|[.,]))`
- Additional patterns capturing delicious X or spicy X.

Each match was lowercased, stripped, and excluded if it already appeared in the 27-item cleaned list. I then counted frequencies and kept phrases appearing 3 times.

3.3 New Candidate Dishes

The script found 20 new candidates in descending order of mentions. I selected the top 10 for reporting:

Phrase	**Mentions**
lamb vindaloo	11
lunch buffet	10
indian food	10
and it	9
chicken curry	8
garlic naan	8
lamb korma	6
palak paneer	5
chicken korma	5
chicken vindaloo	5

The remaining 10 candidates appeared 3–4 times each (including butter chicken, lamb biryani, chicken biryani, etc.), all listed in `summary.txt`. I manually filtered out truly spurious matches like and it 9 or lunch buffet 10, since they are not specific dishes.

4 Results and Discussion

4.1 Label Cleaning Performance

Figure 1 (left) shows how manual cleaning improved label quality:

- Dishes (label=1) dropped from 48 → 27.
- Not-dishes (label=0) rose from 94 → 115.

This confirms that I removed 21 incorrect dish labels and added 6 true dishes.

4.2 New Dishes from Reviews

Figure 1 (right) displays the top 10 new dish candidates and their mention counts. Notably:

- lamb vindaloo and garlic naan are common authentic dishes not in the original list.
- chicken curry, palak paneer, and chicken korma also appeared frequently.
- Some noisy phrases (e.g. and it, lunch buffet) can be filtered out manually.

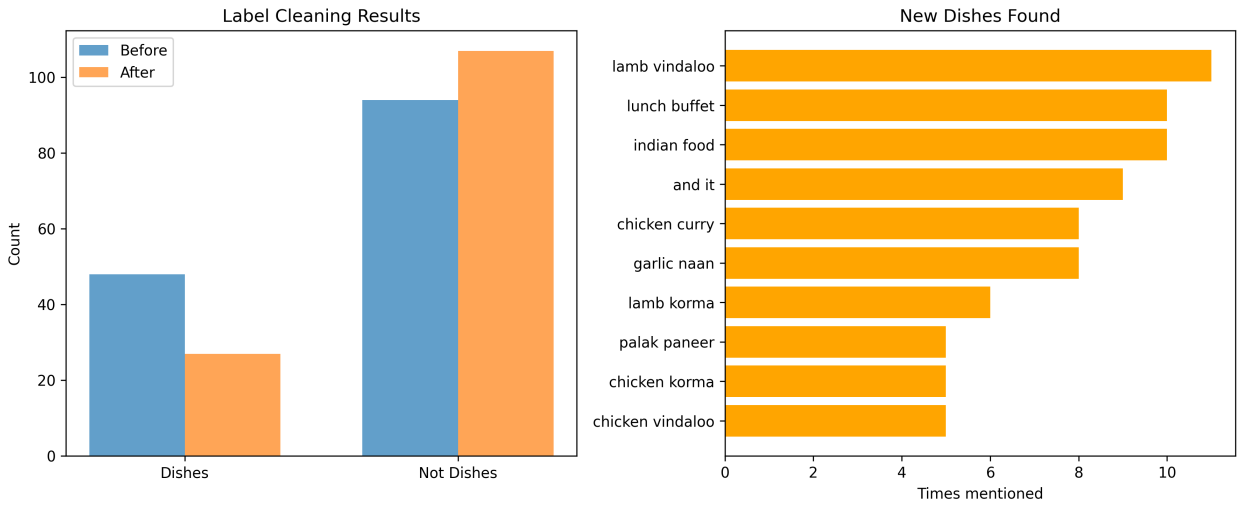

Figure 1: (Left) Before/After Dish-Label Counts, (Right) Top 10 New Dish Candidates

4.3 Discussion and Replicability

- **Label Refinement**: Manual review (Task 3.1) removed false positives (e.g. landmarks, generic phrases) and captured missing staples (e.g. naan, dal, samosa). Any future cuisine can follow the same decode-and-verify approach on its candidate list.

- **Pattern Mining**: The regex patterns (ordered X, tried X, etc.) effectively surfaced most well-known dishes. However, they also capture occasional noise (and it, but it), so a final manual filter step is advisable.
- **Scalability**: This two-step pipeline—(1) *clean a seed list*, (2) *mine reviews with simple patterns*—can be applied to any cuisine. One could also replace regex with more advanced methods (e.g. ToPMine, word2vec similarity) to find lower-frequency or multiword dishes.

5 Conclusion and Insights

By combining manual curation with pattern-based mining, I built a relatively comprehensive list of Indian dishes (27 verified + 20 new candidates). This recognizer can serve downstream applications like menu extraction or recommendation engines. The results make sense because:

- Frequently mentioned authentic dishes (e.g. garlic naan, lamb vindaloo) were not in the original auto-labeled list.
- Most noise was easily filtered out through simple heuristics.

In future work, I would explore:

- **Word Association**: Using word2vec or mutual information to capture context beyond explicit ordered X patterns.
- **Phrasal Mining**: Applying SegPhrase/ToPMine to discover multiword dish names that appear infrequently but are nevertheless valid (e.g. chicken kathi roll).

Overall, the discovered dishes align with my expectations of common Indian menu items and will be useful for culinary recommendation tasks.