Group 1: Recipe Retrieval System Based on Available Ingredients

By Edward Lu, Xulun Luo and Zihan Liu

Agenda

- Motivation
- Problem Definition
- Dataset
- Methodology, Evaluation and Illustrations
- Conclusion

Motivation

🥦 **Reduce Food Waste**

Help users make use of ingredients they already have.

👩‍🍳 **Simplify Cooking Decisions**

Answer the common question: *“What can I cook with what’s in my kitchen?”*

📈 **Improve Recipe Search Relevance**

Go beyond keyword search by matching recipes to real-time ingredient lists.

NYU

Outline & Problem Definition

Given a set of ingredients provided by the user,
retrieve a ranked list of recipes that:

- Use the available ingredients
- Match dietary preferences (optional)
- Allow substitutions when exact matches are unavailable

NYU

Dataset

- RecipeNLG is a large-scale dataset comprising over 2.2 million cooking recipes, designed for natural language generation tasks. https://www.kaggle.com/datasets/saldenisov/recipenlg

- Each recipe includes a title, a list of ingredients, step-by-step instructions, and named entities extracted via Named Entity Recognition (NER), an NLP technique that identifies and classifies key elements in text into predefined categories

NYU

Dataset

We did the cleaning:

- Remove empty or invalid titles, ingredients, and directions
- Merge ingredients with NER (List of used ingredients) phrases (e.g., "chicken tender" → "chicken")
- Mark ingredient as **core** if it's mentioned in the title
- Add tags: **vegetarian** and **gluten-free** based on ingredient content
- Classify recipe as **main**, **side**, or **dessert** from the title
- The output is in json files for further usage

Methodology - TF-IDF

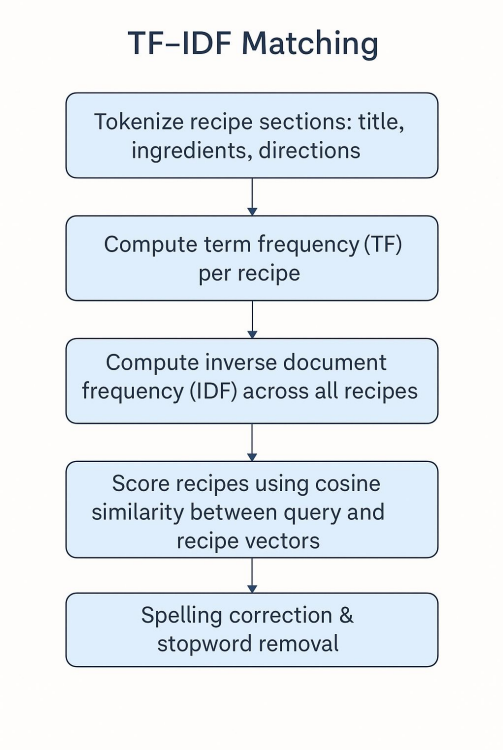
🧾 **Tokenize recipe sections**: title, ingredients, directions

🔠 **Compute term frequency (TF)** per recipe

📊 **Compute inverse document frequency (IDF)** across all recipes

🧮 **Score recipes using cosine similarity** between query and recipe vectors

🧹 **Spelling correction & stopword removal** included for cleaner input



NYU

Fine-Tuning Strategies in version 2

Section-Based Weighting

- Title words receive higher weight (default 2.0)
- Ingredient words receive medium weight (default 1.5)
- Direction/instruction words receive lower weight (default 0.5)

```
# ------------------------------------------------------------
def __init__(self,
             json_path: str  = "cleaned_data/train.json",
             cache_dir: str  = "tfidf_cache",
             title_weight: float = 2.0,
             ingredients_weight: float = 1.5,
             directions_weight: float  = 0.5,
             default_mode: str = "tfidf"  # "tfidf" | "semantic" | "hybrid"
             ):
```

Query Processing

- Tokenization with regex cleaning
- Stopword removal
- Spell correction

```
# basic stop-word list
self.stopwords = {
    'a','an','the','and','or','but','in','on','of','to','from',
    'with','by','for','at','is','are','cup','cups','tablespoon',
    'tablespoons','teaspoon','teaspoons','recipe','about','until',
    'tsp','tsps'
}
```

```
def _tokenize(self, text: str) -> List[str]:
    if not text: return []
    cleaned = re.sub(r'[^a-z\s]', ' ', text.lower())
    return [t for t in cleaned.split() if t and t not in self.stopwords]
```

NYU

Fine-Tuning Strategies in version 3

Dietary Restriction Filtering

- Built-in dictionary mapping dietary restrictions to forbidden ingredients
- Supports vegan, gluten-free, and diabetic filtering

```python
# Dietary restriction → forbidden ingredient keywords
DIETARY_FORBIDDEN = {
    "vegan": {
        "egg", "milk", "butter", "cheese", "cream", "yogurt",
        "honey", "meat", "chicken", "beef", "pork", "fish", "shrimp"
    },
    "gluten_free": {
        "flour", "wheat", "barley", "rye", "bread", "pasta",
        "noodle", "crumbs", "couscous", "semolina"
    },
    "diabetic": {
        "sugar", "honey", "maple", "syrup", "jam", "jelly",
        "brown", "powdered"
    },
}
def _check_diet(self, rec: dict, diet: str) -> bool:
    bad = self.DIETARY_FORBIDDEN.get(diet, set())
    text = ' '.join(i['name'] if isinstance(i,dict) else str(i)
                    for i in rec.get('ingredients', [])).lower()
    return not any(w in text for w in bad)
```

Spelling Correction

- Uses `SpellChecker` library for typo correction
- Falls back to `difflib` for finding close matches in the vocabulary

```python
def _correct_spelling(self, tokens: List[str]) -> List[str]:
    fixed = []
    for t in tokens:
        corr = self.spell.correction(t)
        if corr and corr != t:
            fixed.append(corr)
        else:
            close = difflib.get_close_matches(t, self.vocabulary, n=1, cutoff=0.7)
            fixed.append(close[0] if close else t)
    return fixed
```

NYU

Fine-Tuning Strategies in version 4(Final)

Multi-Modal Search Options

- Pure TF-IDF mode
- Pure semantic search mode
- Hybrid mode combining both approaches

```
def search(self,
           query: str,
           top_n: int = 5,
           debug: bool = False,
           dietary: Optional[str] = None,
           mode:   Optional[str] = None      # "tfidf" | "semantic" | "hybrid"
           ) -> List[dict]:

    mode = mode or self.default_mode
```

NYU

Illustration - TF-IDF Baseline

```
Your query: ramen egg meat

Found 5 matching recipes:
1. Not Just Ramen (Similarity: 0.4971)
   Ingredients:
     - ramen noodles
     - 1 egg
     - 1 cup leftover meat, cooked and chopped (any type of meat will do)
   Directions (first 3 steps):
     - Cook noodles according to package directions.
     - While noodles cook, beat egg in separate bowl.
     - Once noodles are soft and water is boiling, add egg and stir.
2. Saniatan's Fried Noodles (Similarity: 0.4836)
   Ingredients:
     - ramen noodles
     - 1 teaspoon sesame oil
     - 2 teaspoons vegetable oil
     - 1 teaspoon soy sauce
     - 1 green onion
     - 1 ounce fresh ginger, minced
     - 2 cloves garlic, minced
     - 3 ounces meat (chicken, pork, steak, shrimp, even ground beef will do)
   Directions (first 3 steps):
     - Drop ramen into boiling water for*exactly* two minutes.
     - Scramble egg in a small bowl.
     - Drain ramen and rinse twice with cold water.
3. Sesame Ramen Cakes (Similarity: 0.4588)
   Ingredients:
     - ramen noodles
     - 1 large egg, lightly beaten
     - 2 teaspoons soy sauce
     - sesame oil
     - sesame oil
     - Kosher salt
     - sesame oil
     - Sliced scallion greens
   Directions (first 3 steps):
     - Cover the ramen with boiling water and let stand 5 minutes, then drain.
     - Whisk together the egg, soy sauce, sesame oil, sesame seeds, and 1/4 teaspoon salt.
     - Toss the ramen with the egg mixture.
```

The 2nd and 3rd recipe are too complicated to make with ‘ramen egg meat’

—> This shows the model needs fine-tuning.

NYU

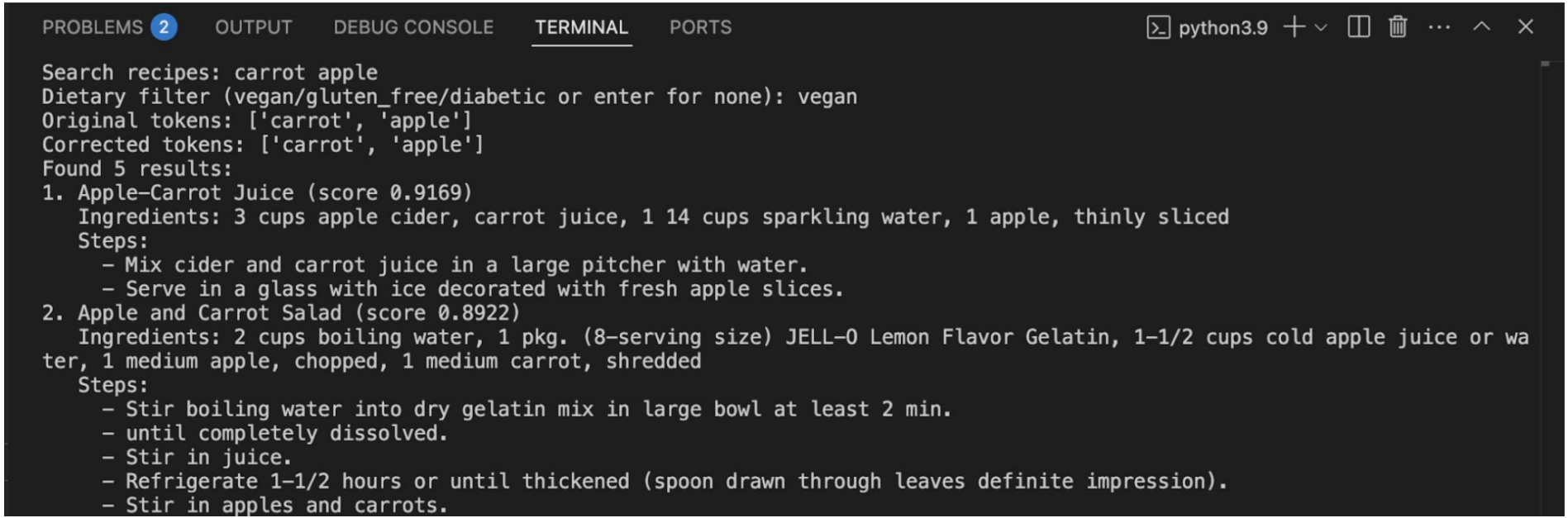
Illustration- TF-IDF V2

```
Search recipes: ramen egg meat

Found 5 matching recipes:

1. Not Just Ramen (Match score: 0.662)
Main ingredients:
  - ramen noodles
  - 1 egg
  - 1 cup leftover meat, cooked and chopped (any type of meat will do)
First cooking step:
  - Cook noodles according to package directions.

2. Ramen in Minutes (Match score: 0.575)
Main ingredients:
  - 3 cups water
  - 1 pkg. (3 oz.) beef-flavored ramen noodle soup mix, divided
  - 2 cups loosely packed baby spinach leaves
  - 2 eggs
  - 2 KRAFT Singles King Sooper's 1 lb For $3.99 thru 02/09
First cooking step:
  - Bring water to boil in medium saucepan.

3. Ramen in Minutes (Match score: 0.574)
Main ingredients:
  - 3 cups water
  - 1 pkg. (85 g) beef-flavoured ramen noodle soup mix, divided
  - 2 cups loosely packed baby spinach leaves
  - 2 eggs
  - 2 Kraft Singles Cheese Slices
First cooking step:
  - Bring water to boil in medium saucepan.
```

The 2nd and 3rd recipe are ‘Ramen in Minutes’ with less ingredients —> easy to make and more accurate result.

NYU

Illustration - TF-IDF V3



NYU

Testing dietary requirements: vegan

Ongoing Enhancement - BERT

- Extracted ingredient names from recipe data and encoded them using Sentence-BERT for semantic comparison.
- Built a FAISS index to enable fast nearest-neighbor search among embedded ingredients. Designed a hybrid search engine that supports keyword-based (TF-IDF), semantic-based, and combined query modes.
- Integrated typo correction, stopword filtering, and cosine similarity scoring to improve query relevance.
- Supported dietary filtering (vegan, gluten-free, diabetic) and returned full recipe details: title, top ingredients, and step-by-step directions.

Illustration - TF-IDF V4(with BERT)

```
Search (q=quit): Chinese-style food
Mode tfidf / semantic / hybrid (enter=default): semantic
Diet (vegan/gluten_free/diabetic or enter):
Batches: 100% 1/1 [00:00<00:00, 180.52it/s]
[
  {
    "id": 1882084,
    "title": "Soya Egg Rice With Chinese Sausage",
    "score": 0.7266,
    "ingredients": [
      "rice",
      "water",
      "garlic",
      "Chinese sausage",
      "chestnuts",
      "sesame oil",
      "dark soya sauce",
      "eggs"
    ],
    "steps": [
      "Combine all the ingredients for the Chinese sausage rice in a rice cooker and cook for 10 minutes.",
      "Meanwhile, lightly crack open the quail eggs, and soak them in sesame oil and soya sauce for 5 minutes.",
      "Top the soya eggs on the cooked sausage rice",
      "Garnish with coriander leaf and fried shallots.",
      "Serve."
    ]
  },
  {
```

NYU

No ingredients, Chinese-style food, semantic search, no Dietary restrictions —> Fried rice ✅

Illustration - TF-IDF V4(with BERT)

NYU

Evaluation

- Method: Manually tested search results using diverse user queries
- Use Cases Covered:
  - Queries with misspelled ingredients
    Inputs containing only meat items
    Inputs containing only vegetables etc.
- Evaluation Criterion:
  For each query, check if the top returned recipe(s) fully include all input ingredients

NYU

Evaluation

Testing stages:

chikcen garlic butter

beef onion pepper salt

pornk celery pepper

buter sugur (This misspel of butter was not correct)

2cups rice milk

1tbsp olive oil garlic

milk flour

`` (empty string)

tomato basil vinegar

carrot celery potato

frid chicken recipe (fird became fried)

salt pepper garlic

flour egg sugar

potatoe herb butter

rice chicken carrots

13/15 Passed

NYU

Evaluation

```
chicken (Basic single-ingredient search)✅
tomato garlic onion (Multiple ingredient combination)✅
chocolate chip cookies (Specific dish search)✅
thai curry (Cuisine-specific search)✅
grilled salmon (Cooking method + ingredient)✅
vegan dessert (Dietary restriction + dish type)✅
saffron risotto (Rare ingredient + specific dish)✅
spageti sauce (Misspelled query)✅
quick dinner (Time constraint + meal type)✅
creamy mushroom chicken pasta with white wine sauce (Long complex query)✅
summer salad (Seasonal association)✅
pie (Short ambiguous query)✅
spicy vegetarian soup easy (Mixed attribute query)✅
slow roasted (Technique-focused query)✅
breakfast (Meal type search)✅
```

15/15 Passed

Evaluation

beef (Basic single-ingredient search)

onion carrot potato (Multiple ingredient combination)

banana bread (Specific dish search)

mexican tacos (Cuisine-specific search)

baked cod (Cooking method + ingredient)

gluten-free cake (Dietary restriction + dish type)

black garlic pasta (Rare ingredient + specific dish)

chikcen soup (Misspelled query)

easy lunch (Time constraint + meal type)

roasted butternut squash with sage and brown butter (Long complex query)

winter stew (Seasonal association)

toast (Short ambiguous query)

spicy vegan chili fast (Mixed attribute query)

grilled slow lamb (Technique-focused query)

brunch (Meal type search)

13/15 Passed

NYU

Evaluation

Accuracy:

First 15: 13/15 = 0.867

Second 15/15 = 1

Third 13/15 = 0.867

Average Accuracy:

(0.867+1+0.867)/3 = 0.911

Conclusion & Future

What we did:

- Developed a recipe retrieval system using **TF-IDF** and **semantic similarity**
- System handles **typos**, **dietary filters**, and **ingredient substitution**
- Manual tests show high match accuracy across varied user queries

Future outlook:

- Incorporate nutritional analysis for health-conscious users
- Add natural language support (e.g., “I want something low-carb with eggs”)
- Deploy as a web or mobile app for broader accessibility

NYU

Thank You