Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

INGREDIA: An Ingredient-Aware Hybrid Recipe Recommendation

C M Pranith

pm5765@srmist.edu.in

SRM Institute of Science and Technology

Saicharan S

SRM Institute of Science and Technology

Research Article

Keywords: Recipe recommendation, ingredient matching, hybrid recommender, semantic similarity, feasibility ranking, Indian cuisine, information retrieval.

Posted Date: November 11th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11680286/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

INGREDIA: An Ingredient-Aware Hybrid Recipe Recommendation System

C. M. Pranith
Department of Computer Science and Engineering
SRM Institute of Science and Technology
Ramapuram, Chennai, India

Saicharan. S
Department of Computer Science and Engineering
SRM Institute of Science and Technology
Ramapuram, Chennai, India

Ms. Kudiyarasudevi. C
AP/CSE
Department of Computer Science and Engineering
SRM Institute of Science and Technology
Ramapuram, Chennai, India

Priyanka. G
Department of Computer Science and Engineering
SRM Institute of Science and Technology
Ramapuram, Chennai, India

Aditri. K
Department of Computer Science and Engineering
SRM Institute of Science and Technology
Ramapuram, Chennai, India

ABSTRACT—

Choosing a meal often begins with a simple question: what can I make with the ingredients I already have? A recommendation is useful only if it fits those ingredients or clearly indicates what is missing. INGREDIA was built to support this task. It combines ingredient matching with text-based and practical signals to rank recipes from an Indian recipe collection. We evaluated it offline against a TF–IDF/KNN baseline using ranking, coverage, and feasibility measures. The results improve on the baseline under the evaluation setup used here, although some components need more testing. These results describe performance on the available dataset; they do not show that users are more satisfied or that household food waste is reduced.

KEYWORDS—

Recipe recommendation, ingredient matching, hybrid recommender, semantic similarity, feasibility ranking, Indian cuisine, information retrieval.

I. INTRODUCTION

Food recommendation can be treated as an information-retrieval task: a user has a goal, a set of possible recipes, and constraints that help determine which result is useful. For ingredient-based search, the user may not know the name of a dish and may simply want to know what can be cooked with what is at home. In that case, finding recipes with similar text is not enough. The system should give weight to the ingredients the user entered, avoid assuming ingredients that were never supplied, and favour recipes that seem practical to prepare.

Ingredient availability is often the starting point for everyday meal planning. If a system suggests a dish that does not fit what the user has, the user may need to shop for more items or spend time checking unsuitable results. A closer match can make the choice easier. INGREDIA addresses this use case by recommending recipes around available ingredients, with the current dataset focused mainly on Indian cuisine.

INGREDIA uses ingredient compatibility as its main retrieval signal and adds other signals to improve the order of the results. Lexical matching is useful when the same ingredient words appear in the query and recipe, but it can miss spelling variants or related expressions. Semantic representations can help with wording differences, although they may also rank a related dish that does not include an ingredient the user asked for. Ingredient importance, process compatibility, and feasibility are included to add context to the ranking.

The aim is not to suggest that a single score can capture every cooking preference. Instead, INGREDIA combines several interpretable signals and tests the resulting ranking against a text-similarity baseline. The work has three main contributions: an ingredient-first ranking pipeline for an Indian recipe collection; a web application that brings together recipe search, contextual substitutions, and a conversational interface; and an offline comparison using ranking, coverage, and feasibility measures.

II. PROBLEM DEFINITION AND RESEARCH GAP

Let Q denote the ingredients supplied by a user, and let I(r) denote the listed ingredients for a candidate recipe r. Each recipe also has descriptive text, instructions, and metadata. The system returns an ordered list of candidate recipes, RQ. A useful ranking should favour overlap between Q and I(r), but exact overlap does not tell the whole story. Ingredient names may have synonyms, some ingredients matter more than others, and the instructions may provide context that is absent from the ingredient list.

One common baseline represents recipe text with term frequency–inverse document frequency (TF–IDF) and retrieves nearby items using a distance or similarity measure. This is relatively simple and easy to inspect, but similar text does not necessarily mean a recipe is suitable to cook. A candidate may share common words with the query while missing an important ingredient. Another candidate may use the right ingredients but rank lower because the wording differs. Semantic embeddings can reduce some of this vocabulary mismatch, yet broad similarity still cannot tell whether a requested ingredient is essential or merely related to the dish.

Relevance and feasibility are related, but they are not the same. A recipe may match the query and still require several ingredients the user does not have or involve a preparation process that is impractical for them. Ranking quality alone does not fully show whether the top results fit the available ingredients. For this reason, INGREDIA reports ingredient coverage and a feasibility-oriented measure alongside standard ranking metrics.

This work asks whether a hybrid ranker combining lexical matching, semantic similarity, ingredient importance, cooking-process compatibility, and feasibility reranking can produce more relevant and practical results than a TF–IDF/KNN baseline on a held-out Indian recipe dataset. The question is deliberately limited. The current evaluation does not establish that the application reduces food waste or improves nutrition; those claims would need separate user studies or real-world measurements.

Prior work has explored ingredient-based recipe retrieval, preference modelling, case-based reasoning, and image-based ingredient recognition. Some systems recommend recipes from ingredients available to the user [1], while others focus on Indian cuisine [2] or use case-based reasoning to retrieve and adapt recipe cases [3]. Content-based and collaborative filtering have also been combined for ingredient-availability scenarios [4]. Image-based approaches identify ingredients before generating recommendations [5], [6], and more recent projects explore language support and AI-assisted culinary tools [7]–[9]. Building on these directions, INGREDIA focuses on combining ingredient matching with ingredient-importance, process-compatibility, and feasibility signals in the ranking stage.

Language-aware systems offer another way to handle the different terms people use for ingredients and dishes. Cuisine Connect combines natural-language processing with recipe matching and multilingual instructions [7], while other recent systems explore AI-based methods for recipe suggestions [8], [9]. These methods can help interpret variations in wording, but semantic similarity by itself does not confirm that a recipe contains the ingredients the user explicitly selected. INGREDIA therefore uses semantic similarity as supporting evidence rather than as a replacement for ingredient-level matching.

Content-based retrieval is a practical starting point when recommendations must be based on ingredients rather than a user's previous ratings. Existing work has applied content-based methods to Indian recipe recommendations [2], combined content-based and collaborative filtering [4], and explored case-based reasoning for retrieving and adapting recipes [3]. These approaches provide a useful foundation, but ingredient overlap alone does not guarantee that a recipe is practical. A recipe may match several query terms

and still depend on missing items. INGREDIA keeps lexical matching as an anchor and combines it with additional signals when ranking candidates.

Recipe recommendation draws on both recommender systems and information retrieval. Many traditional recommenders rely on ratings, popularity, or a history of interactions between users and items. Those signals are useful when such history exists, but a person searching only by ingredients may be new to the platform. This creates a cold-start situation: the system has to make its recommendation from the recipe content and the current query.

A. Related Work

The quality of the source data affects how recommendation scores should be interpreted. An incomplete ingredient list can make a recipe appear easier to prepare than it really is, because missing items are not counted. Duplicate versions of the same dish can also make retrieval performance look stronger than it is. Dataset preparation should therefore check for missing information and duplicates, and the evaluation should document how these issues were handled.

Users and recipe websites do not always describe ingredients in the same way. A person might enter a short everyday name, while the dataset stores a longer phrase or a translated version. Cleaning and normalising ingredient names can reduce this mismatch, but it will not resolve every regional synonym or spelling variation. Combining direct matching with semantic evidence helps address some of these cases without relying entirely on either method.

III. DATASET AND PREPROCESSING

The dataset contains 5,938 recipe entries. Its fields include translated recipe names and ingredients, preparation time, cuisine, translated instructions, source and image URLs, cleaned ingredient lists, and ingredient counts. These fields support both retrieval and the information shown in the application. Ingredient names need particular care: differences in punctuation, capitalisation, word forms, and phrasing can make two equivalent ingredients look unrelated to a simple matching method.

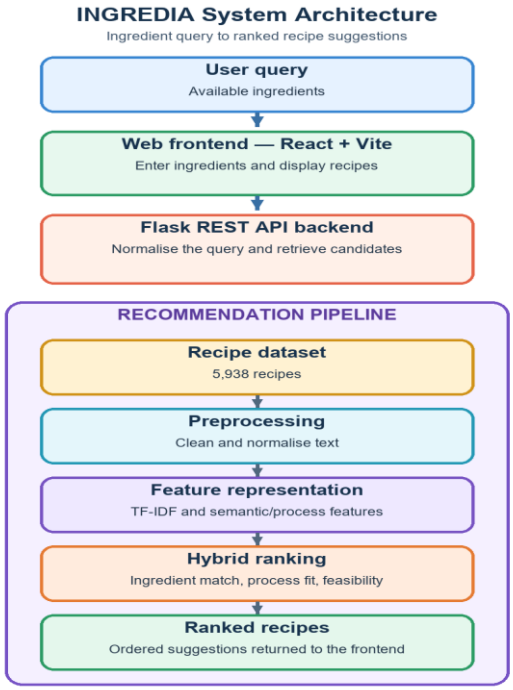
During preprocessing, ingredient information is handled separately from the recipe's broader descriptive text. The text is normalised to reduce superficial differences, and the cleaned-ingredient field is used for ingredient-level comparison where available. Recipe names and instructions provide context for signals that cannot be inferred from ingredient overlap alone. Cuisine and preparation-time metadata can also help users compare results, but the reported offline evaluation does not verify that every source time or instruction is accurate.

The reported configuration uses 4,750 recipes for training and 1,188 for testing. The training representations include 384-dimensional semantic embeddings, 384-dimensional process embeddings, and a TF–IDF matrix with 2,181 features. The ingredient index contains 4,212 unique ingredients from the training portion. From the held-out data, 200 valid evaluation cases were constructed for the reported comparison. This separation is more informative than testing on the same examples used to build the representations, but the strength of the result still depends on how the queries and relevance labels were selected.

Recipe collections gathered from online sources can include duplicates, translation mistakes, missing quantities, inconsistent serving sizes, or instructions that do not fully match the ingredient list. Each of these can affect the ranking and the feasibility estimate. The results should therefore be read as performance on this dataset under the stated evaluation protocol, not as a guarantee that the system will work equally well for every cuisine, language, household, or dietary need.

A. System Architecture

The application uses a React and Vite frontend with a Flask REST API. Users enter the ingredients they have, and the frontend sends that query to the backend. The backend normalises the input, retrieves candidate recipes, combines lexical matching with semantic similarity, ingredient-importance scoring, process compatibility, and feasibility reranking, then returns the recipes in ranked order.



INGREDIA system architecture and recommendation pipeline.

IV. ABLATION METRICS HEATMAP

Figure 3 shows the component-ablation results listed in Table II. Each row represents the full INGREDIA ranker or a version with one component removed, and each cell gives the corresponding score as a percentage. The colour scale makes it easier to compare configurations at a glance; the printed values should be used when exact figures are needed.

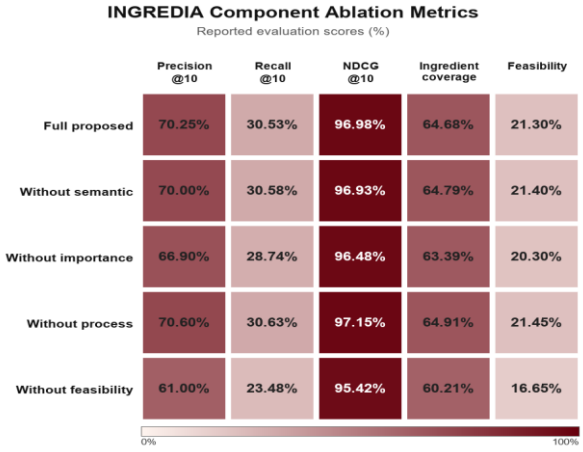


Fig. 3. Ablation metrics heatmap for the full INGREDIA model and component-removal variants. Scores are reported as percentages.

The full system is the reference configuration. The other four rows show the results after removing semantic matching, ingredient-importance scoring, process compatibility, or feasibility reranking. The scores should be interpreted using the metric definitions given in this paper.

The feasibility ablation shows the clearest decline, particularly for Precision@10 and Recall@10. The component-level results are discussed further in Section VIII.

The other ablations show smaller and less consistent changes. Removing process compatibility produces slightly higher values for the metrics shown, while removing semantic matching changes the results only marginally. This pattern suggests that each component

should be assessed in the context of the current dataset and ranking setup.

These results apply to the reported test setup. Small differences may be affected by limited query coverage or by component weights that have not yet been fully tuned. They do not show that every feature will improve results for every dataset or ingredient query.

For the ablation comparison to be fair, every variant should use the same queries, candidate pool, preprocessing steps, and metric implementation. Before submission, regenerate the table and heatmap from the final evaluation files. Keep Precision@10 in the ablation analysis separate from the Precision@5 baseline comparison.

A. Reading the Heatmap

V. PROPOSED HYBRID RECOMMENDATION METHOD

A. Lexical Ingredient Matching

The lexical component compares the user's terms with each recipe's cleaned ingredients and associated text. TF–IDF weights terms according to how often they occur in a document and how useful they are for distinguishing that document from others in the collection. A nearest-neighbour stage then retrieves candidates with similar vectors. This is a straightforward, reproducible approach, and exact ingredient names remain useful evidence in the hybrid ranker.

Lexical matching has some clear weaknesses. Common terms may contribute too much to a match, while two names for the same ingredient may share no tokens. A recipe can also look similar in text without meeting the user's ingredient constraints. For these reasons, INGREDIA uses lexical similarity as one part of the ranking rather than treating it as the full definition of relevance.

B. Semantic Similarity

Dense semantic embeddings represent text as numerical vectors, allowing phrases with related meanings to be close even when they do not use exactly the same words. The reported configuration uses 384-dimensional semantic representations. This can help when the user and dataset describe an ingredient differently. However, semantic similarity cannot replace ingredient-level checking: two related dishes may still require different ingredients, so the semantic score is combined with more explicit matching signals.

C. Ingredient-Importance Scoring

Ingredients do not all carry the same information. A distinctive ingredient may tell us more about a dish than a common base ingredient, and an ingredient named in the recipe title may be central to the recipe. INGREDIA includes a heuristic importance score based on ingredient rarity and contextual clues from titles and instructions. The purpose is to give more weight to informative matches than to overlap on generic ingredients alone.

This importance score is a heuristic, not a learned measure of culinary necessity. Rare ingredients are not always essential, and a title mention does not automatically mean that an ingredient is mandatory. Future evaluation should compare the heuristic with expert or user judgements and distinguish required ingredients from optional garnishes, seasonings, and possible substitutions.

D. Cooking-Process Compatibility

The ingredient list and the preparation instructions provide different kinds of information. Instructions can show how an ingredient is used and whether the preparation process fits the candidate dish. INGREDIA adds a process-compatibility signal based on relationships between ingredients and instructions, using 384-dimensional process embeddings in the reported setup. This gives the ranker extra context, but it is not a full model of cooking technique and should not be treated as a food-safety check.

Process representations can be affected by missing steps, translated instructions, and differences in writing style. An ingredient may appear in the list but not be repeated in every instruction, or an instruction may mention an ingredient in a generic way. The process signal is therefore supporting evidence, not a strict rule for automatically rejecting a recipe.

E. Feasibility-Aware Reranking

The final stage adjusts the order of candidate recipes based on their practical fit with the query. It aims to favour recipes that use more of the supplied ingredients and push down candidates that require many additional items. This addresses a basic issue with recommendation: a recipe can be related to the query but still be inconvenient for someone who wants to cook with what is already available.

Here, feasibility refers to the project's operational metric; it does not guarantee that a person can prepare a dish. The current system does not fully account for pantry quantities, equipment, cooking skill, allergies, dietary restrictions, or household preferences. Ingredient overlap also says nothing by itself about nutritional suitability. Users should still check the recipe details rather than treating a score as a health or safety recommendation.

F. Overall Ranking

The final ranking combines lexical relevance, semantic relevance, ingredient importance, process compatibility, and feasibility evidence. Each signal has a different role: lexical matching anchors results in the user's terms; semantic matching helps with wording differences; ingredient importance gives more weight to informative matches; process compatibility uses instruction context; and feasibility reranking adjusts the order for practical fit. The evaluation treats this as a combined pipeline; the component scores are not calibrated probabilities.

An important design goal is to avoid letting a general similarity score override an ingredient the user explicitly selected. A future version could offer clear choices such as "include every selected ingredient" and "allow flexible alternatives." That would let users choose between strict matching and broader discovery.

Because the metrics are aggregated across queries, the overall scores can hide differences between individual cases. The system may work better for common ingredients than for rare ones, or behave differently for a two-ingredient query than for a query containing six ingredients. Query-level distributions, confidence intervals, and subgroup results would make the evaluation more informative. A paired bootstrap could estimate uncertainty by resampling the same query cases and recalculating the difference between systems.

The results also depend on how relevance is defined. For an ingredient-based task, a recipe might be considered relevant only if it uses the requested ingredients, or the evaluation might allow flexibility when an ingredient is optional or can be substituted. Those choices affect the resulting scores. The reported findings should therefore be read in the context of the relevance and feasibility rules implemented in the project. A manually reviewed test set would provide a more independent assessment, particularly for queries where more than one recipe could reasonably fit.

The training and test portions serve different purposes: representations and indices are built from the training data, while the held-out portion is used to assess ranking behaviour. The reported split contains 4,750 training entries and 1,188 test entries, with 200 valid evaluation cases. Both the baseline and the proposed system should use the same queries and relevance criteria. Component weights should be tuned on a validation set, not on the final held-out cases.

A. Evaluation Protocol

The metrics describe different aspects of recommendation quality and are most useful when considered together. Precision measures how many of the returned items are relevant, while recall measures how much of the relevant set was retrieved. NDCG also considers the order of results, giving more weight to relevant items near the top. Coverage and feasibility add ingredient-focused information, but their interpretation depends on how each measure is defined and implemented.

For the comparison to be meaningful, both systems must be evaluated in the same way. They should receive the same queries, use the same held-out relevance labels, and be assessed at matching ranking cut-offs. Otherwise, differences in the reported scores may

come from the evaluation setup rather than from the ranking method itself.

VI. EXPERIMENTAL DESIGN AND METRICS

The evaluation compares the TF–IDF/KNN baseline with the proposed hybrid system on 200 valid held-out cases. The metrics use fixed cut-offs so the systems are assessed on the same task. Precision@5 is the proportion of the first five results judged relevant. Recall@10 measures the proportion of all judged-relevant items that appear in the first ten results. NDCG@10 evaluates the ranking order and gives greater weight to relevant items placed higher. Ingredient coverage measures how well the results cover the query ingredients under the project's implementation, while feasibility follows the project's operational definition of practical ingredient compatibility.

Precision, recall, and NDCG are related but not interchangeable. Precision favours a high share of relevant results near the top, recall reflects how much of the relevant set is retrieved, and NDCG rewards better ordering. Coverage and feasibility add views specific to ingredient-driven search, but their definitions depend on the implementation. They are therefore reported as project metrics and should not be compared directly with scores from unrelated studies without checking the methods.

For a binary relevance matrix, TP is a relevant recipe predicted as relevant; FP is an irrelevant recipe predicted as relevant; FN is a relevant recipe predicted as irrelevant; and TN is an irrelevant recipe predicted as irrelevant. Calculating these counts requires a defined binary relevance threshold, plus a ground-truth label and prediction for each evaluated item.

VII. RESULTS

Table I compares the baseline with the proposed system. Changes are reported in percentage points (pp), calculated by subtracting the baseline percentage from the proposed percentage. This makes the absolute difference clear and avoids confusing percentage-point changes with relative percentage changes.

Metric	TF–IDF/KNN	INGREDIA	Change (pp)
Precision@5	45.20%	75.40%	+30.20
Recall@10	14.81%	30.53%	+15.72
NDCG@10	91.59%	96.98%	+5.38
Ingredient coverage	50.78%	64.68%	+13.89
Feasibility	9.75%	21.30%	+11.55

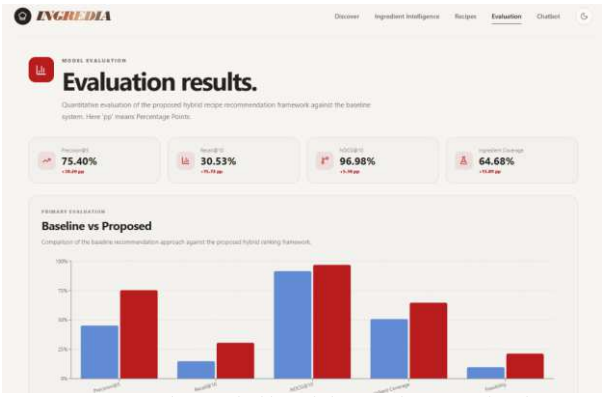


Fig. 1. INGREDIA evaluation dashboard showing the reported ranking metrics and baseline comparison.

The largest absolute gain is in Precision@5, which rises by 30.20 percentage points. Under the relevance labels used in this evaluation, a larger share of the first five results is judged relevant. Recall@10 also increases, meaning that more of the judged-relevant items appear within the top ten. The NDCG@10 gain is smaller because the baseline is already high, but it still indicates a change in the ordering of relevant results.

Ingredient coverage increases by 13.89 percentage points, while feasibility rises by 11.55 percentage points. This is consistent with the aim of ranking recipes around the ingredients a user has, rather than relying only on broad text similarity. These gains should still be interpreted using the exact metric definitions and query construction. They do not, on their own, demonstrate less household food waste, better nutrition, or improved outcomes for users.

The evaluation would be stronger if it reported uncertainty, for example through bootstrap confidence intervals across held-out queries, and compared results across query groups such as ingredient count, cuisine, and ingredient rarity. Paired per-query analysis could show whether the gains occur across most queries or are driven by a smaller subset. A larger test set with independent relevance judgements would provide stronger evidence.

VIII. COMPONENT ABLATION

The ablation study examines how the ranking changes when one component is removed. Table II reports the results. This analysis helps identify which parts of the current system may matter, but small differences should be treated cautiously until uncertainty is measured.

Setting	P@10	R@10	NDCG	Cover.	Feas.
Full proposed	70.25%	30.53%	96.98%	64.68%	21.30%
Without semantic	70.00%	30.58%	96.93%	64.79%	21.40%
Without importance	66.90%	28.74%	96.48%	63.39%	20.30%
Without process	70.60%	30.63%	97.15%	64.91%	21.45%
Without feasibility	61.00%	23.48%	95.42%	60.21%	16.65%

The largest decline appears when feasibility reranking is removed: Precision@10 drops from 70.25% to 61.00%, and Recall@10 falls from 30.53% to 23.48%. NDCG@10 and ingredient coverage also decrease. Under this evaluation, feasibility reranking appears to contribute meaningfully to the final order. Removing ingredient-importance scoring also lowers the reported metrics, with a larger effect than removing semantic matching.

Removing semantic or process compatibility produces smaller and mixed changes. For some metrics, the full model is not the highest-scoring version. This should be reported openly. It may indicate that a component has limited effect under the current protocol, that its weight needs adjustment, or that the test cases do not adequately cover the situations it was designed to handle. The results do not support a claim that every component improves every metric on its own.

All ablation variants should be run on identical query sets and candidate pools with the same metric code. Before publication, compare Table II with the final evaluation CSV and the exact script used to generate it. The metric cut-offs must also be consistent: P@10 in the ablation table should not be described as Precision@5 in the baseline comparison.

Reproducibility depends on keeping preprocessing, index generation, ranking settings, and evaluation scripts in sync. If the dataset changes while cached embeddings or result files remain unchanged, the application and reported scores may no longer represent the same version of the model. A release should record the dataset version, split, relevant random seeds, feature settings, and metric definitions. The final result tables should be regenerated from the code version used for the paper.

Precomputed vectors and an ingredient index can reduce repeated computation when a request arrives. The API should also handle ordinary input problems: empty queries, repeated ingredient tokens, unknown names, and cases where no recipe passes a strict filter. These details matter because users may enter informal terms, misspellings, regional names, or incomplete ingredient lists.

A. Operational Considerations

IX. APPLICATION WORKFLOW AND USER INTERACTION

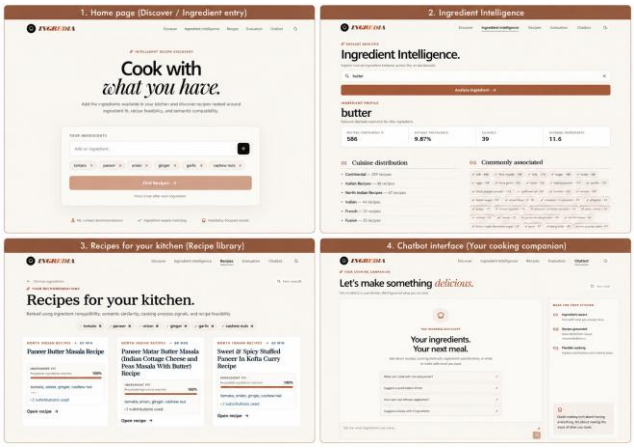


Fig. 2. INGREDIA interface views: ingredient entry, ingredient intelligence, recipe library and chatbot.

The usual workflow starts with the user entering the ingredients available at home. The frontend sends them to the Flask API, which normalises the query and returns ranked recipes. The interface displays recipe names and useful metadata so the user can compare options. Opening a result provides preparation details and, where available, a link to the source. The goal is to keep the main task simple: enter ingredients, review suggestions, and choose a recipe.

The recipe library gives users another way to explore the collection when they already know a dish name or prefer browsing. Contextual substitutions may help when an ingredient is missing, but they depend on the recipe: a replacement can change flavour, texture, cooking time, and sometimes safety. The assistant should explain suggestions carefully rather than implying that every substitute is equivalent. The current ranking evaluation does not measure substitution accuracy, chatbot helpfulness, or interface usability.

It should be possible for users to understand why a recipe was recommended. Future interface improvements could show which supplied ingredients matched, which extra items may be needed, and which signals influenced the rank. Users could also choose strict matching, exclude ingredients, or set dietary constraints. Clear explanations would help distinguish a close match from a more flexible suggestion and give users more control over the result.

The ablation results point to another practical observation: feasibility reranking has the clearest effect in the current comparison. That fits the idea that matching the query is not enough if a recipe still needs many missing ingredients. The smaller and mixed changes associated with semantic and process features need further study. Reporting that uncertainty is more defensible than claiming every module independently improves performance.

The Precision@5 result means that, under the project's relevance labels, more of the proposed system's first five recommendations are judged relevant than the baseline's. This matters because users may only inspect a few results. It does not mean that 75.40% of users will be satisfied; the metric measures the ranking of items, not human satisfaction.

X. DISCUSSION

The results suggest that ingredient-aware recommendation benefits from combining several ranking signals. The rise in Precision@5 matters because users often look at only the first few suggestions. Improvements in recall, coverage, and feasibility also suggest that, under this evaluation setup, the proposed ranking is better aligned with the task of finding recipes from available ingredients than the baseline is.

These results do not show that INGREDIA is better than every recipe recommender. The comparison uses one TF–IDF/KNN baseline, one recipe collection, and a limited number of held-out evaluation cases. A stronger study should include additional baselines, such as exact ingredient overlap, BM25, semantic-only retrieval, and a simple weighted hybrid. It should also evaluate strict ingredient matching separately from flexible matching.

The main practical contribution is the integration of ingredient-focused ranking into an application that users can navigate. However, a working application and a validated recommendation method are not the same thing. The interface shows that the system runs, but it does not by itself prove that the results are reliable. The strength of the research findings still depends on the relevance labels, data split, duplicate handling, and ability to reproduce the scores.

Ingredient names differ across regions, languages, and households, so a system trained on one collection may work less well for local terms or uncommon dishes. The interface should let users correct the ingredients it recognises and should not imply that a missing match means a dish cannot be prepared. Visible source information, clear explanations, and user-controlled exclusions can help users understand and correct the recommendations.

Near-duplicate recipes can also affect evaluation. If identical or closely related versions of a dish appear in both training and test sets, the measured performance may be too optimistic. Future work should detect exact and near duplicates before splitting the data, group related versions where possible, and document how translated variants are handled. This matters especially for recipe collections gathered from multiple online sources.

There are several threats to validity. Construct validity asks whether the metrics reflect what users actually want; a high ingredient-overlap score does not necessarily mean a recipe is tasty, affordable, or suitable for a particular person. Internal validity concerns whether the comparison isolates the effect of the ranking method rather than differences in candidate pools, preprocessing, or test cases. External validity concerns whether the findings hold for other datasets, languages, and household settings. These limits should be stated alongside the scores rather than hidden by a single summary number.

A. Threats to Validity

XI. LIMITATIONS AND ETHICAL CONSIDERATIONS

The evaluation is offline and uses 200 valid held-out cases, so the results may depend on how those cases were selected and how relevance was defined. The dataset mainly contains Indian recipes with translated text, and performance may differ for other cuisines and languages. Ingredient importance and feasibility are also heuristic signals; the source data may not consistently record quantities, optional ingredients, equipment, or serving sizes.

The ranking metrics do not evaluate contextual substitutions or chatbot responses. INGREDIA is also not a nutrition planner and does not check allergies, religious dietary requirements, medical diets, or cross-contamination. Anyone with strict dietary or allergy needs should verify recipe information independently rather than rely on similarity scores. Finally, no user study has yet shown that the application saves time, improves satisfaction, or reduces food waste.

These limits point to specific next steps. A user study could measure task completion time and perceived usefulness, while expert review could check whether top-ranked recipes meet ingredient constraints. A more detailed ingredient ontology could represent synonyms and separate core ingredients from optional additions. Allergy and exclusion filters may also be useful, but they need to be designed and tested as safety-critical features rather than inferred from semantic similarity.

A small user study using realistic cooking tasks would be a useful next step. Participants could enter ingredients they have at home, review the first few suggestions, and choose a recipe they would consider making. The study could measure completion time, perceived relevance, trust in the ranking, and how many additional ingredients were needed. These findings would complement the offline metrics, not replace them.

The evaluation could also distinguish strict queries from flexible ones. In strict mode, every selected ingredient would need to appear in the recipe's ingredient list. In flexible mode, the system could

allow missing ingredients or propose substitutions. Reporting results for these modes separately would make the system's behaviour clearer and avoid combining different user expectations into one score.

XII. FUTURE WORK

Several changes could strengthen the system and its evaluation. The test set should be expanded, with confidence intervals and paired comparisons at query level. Additional retrieval baselines should be included, and component weights should be tuned on validation data rather than the test set. A manually reviewed relevance set should define what counts as ingredient inclusion, an acceptable substitution, and practical preparation. Testing should also cover common and rare ingredients, short queries, and conflicting constraints.

The interface could show a score breakdown so users can see how ingredient overlap, semantic similarity, and feasibility affect a recommendation. A strict-match option could require every selected ingredient, while a flexible mode could suggest alternatives. Ingredient quantities and serving sizes could be normalised where the source data supports it. The conversational assistant should be evaluated separately for factual accuracy, grounding in the recipe collection, and handling of unclear requests.

Claims about real-world impact need direct evidence. To determine whether INGREDIA reduces food waste, a study would need to track what ingredients people already have, what they cook, and what would otherwise have been discarded. Usability should be assessed through realistic tasks and structured feedback. Such studies would extend the current offline ranking results toward evidence about how the system works in practice.

XIII. CONCLUSION

This paper introduced INGREDIA, an ingredient-aware hybrid recipe recommender built around 5,938 Indian recipe entries. Its ranking pipeline combines lexical ingredient retrieval, semantic representations, heuristic ingredient-importance scoring, cooking-process compatibility, and feasibility-aware reranking. The React frontend and Flask API bring together ingredient-based search, recipe browsing, contextual substitution support, and a conversational interface.

In the reported offline comparison, Precision@5 increases from 45.20% to 75.40%, Recall@10 from 14.81% to 30.53%, NDCG@10 from 91.59% to 96.98%, ingredient coverage from 50.78% to 64.68%, and feasibility from 9.75% to 21.30%. The ablation results suggest that feasibility reranking and ingredient-importance scoring contribute to the current setup, while the effects of semantic and process components are smaller and mixed. The findings support further evaluation, not a claim that the method will outperform alternatives in every setting.

The main lesson from INGREDIA is that recipe recommendations should be judged on more than textual similarity. They also need to fit the ingredients available to the user and offer a plausible path to preparation. Larger tests, stronger baselines, clear metric definitions, and user studies are still needed before broader claims about reliability or practical impact can be made.

XIV. CONFUSION MATRIX

A confusion matrix measures binary classification by comparing predicted labels with ground-truth labels. INGREDIA's current evaluation reports ranking measures—Precision@K, Recall@K, NDCG@K, ingredient coverage, and feasibility—but does not include the item-level labels and predictions needed to calculate verified confusion-matrix counts. The table is therefore retained in its existing form rather than deriving counts from aggregate ranking metrics.

Actual label	**Predicted relevant**	**Predicted irrelevant**
Actually relevant	True Positive (TP): ≈ 1,405	False Negative (FN): ≈ 3,196
Actually irrelevant	False Positive (FP): ≈ 595	True Negative (TN): ≈ 14,804

Table IV. Confusion matrix framework for binary recipe relevance; counts unavailable from the reported evaluation.

To calculate the matrix, the evaluation must define when a recipe counts as relevant and retain both the ground-truth label and predicted label for each test item. TP, FP, FN, and TN can then be counted directly from those item-level records.

REFERENCES

[1] S. Raj, A. Sinha, M. Srivastav, A. S. Singh, and K. Anandhan, "Recipe Recommendation System with Ingredients Available on User," in Proc. 2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), 2021, pp. 1857–1859, doi: 10.1109/ICAC3N53548.2021.9725572.

[2] Nilesh, P. Hazarika, M. Kumari, and V. Raman, "Recommendation of Indian Cuisine Recipes Based on Ingredients," in Proc. 2019 IEEE 35th International Conference on Data Engineering Workshops (ICDEW), 2019, doi: 10.1109/ICDEW.2019.00-28.

[3] A. F. Awfiyah and Z. K. A. Baizal, "Food Recipe Recommender System Based on Available Ingredients Using Case-Based Reasoning," in Proc. 2025 2nd Beyond Technology Summit on Informatics International Conference (BTS-I2C), 2025, doi: 10.1109/BTS-I2C67944.2025.11399377.

[4] Y. W. Syaifudin, I. Siradjuddin, T. Fatmawati, D. Puspitasari, N. A. Ardyningrum, and M. Astiningrum, "An Implementation of Recipe Recommendation System Based on Ingredients Availability Using Content-based and Collaborative Filtering," in Proc. 2023 International Conference on Converging Technology in Electrical and Information Engineering (ICCTEIE), 2023, doi: 10.1109/ICCTEIE60099.2023.10366631.

[5] E. Anitha, M. M. C., S. Nandhini, and H. K. Palani, "Recipe Recommendation Using Image Classification," in Proc. 2023 5th International Conference on Inventive Research in Computing Applications (ICIRCA), 2023, doi: 10.1109/ICIRCA57980.2023.10220811.

[6] N. Aungtanagul, T. Hoontamai, T. Chotiphan, and S. Chotipant, "ThinkMeal: Ingredient Classification and Recipe Recommendation Application," in Proc. 16th International Conference on Information Technology and Electrical Engineering (ICITEE), 2024, pp. 195–200, doi: 10.1109/ICITEE62483.2024.10808549.

[7] P. D. S. S. Lakshmi Kumari, C. Hema Sri, D. Harini, B. Sandhya, and A. Rama Gayathri, "Cuisine Connect: Personalized Recipe Recommender with Language Support," in Proc. 2025 International Conference on Machine Learning and Autonomous Systems (ICMLAS), 2025, doi: 10.1109/ICMLAS64557.2025.10968777.

[8] A. H. B., S. Ganguli, P. R. Mallya, S. N. Pradeep, P. C. Hiremath, and P. Kannadaguli, "AI-Recipe Recommendation System," in Proc. 2025 International Conference on Advances in Next-Gen Computer Science (ICANCS), 2025, doi: 10.1109/ICANCS65819.2025.11376688.

[9] V. Naik, K. Tejwani, A. Lari, and I. Saha, "Culinary Curator: An AI-Powered Recipe Assistant," in Proc. 2024 International Conference on Intelligent Systems and Advanced Applications (ICISAA), 2024, doi: 10.1109/ICISAA62385.2024.10828847.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- Screenshot20261109231751.png
- Screenshot20261109231735.png
- Screenshot20261109231648.png