Final Report - Recipe Recommender - Team #34: Team EU

***Team Members / Authors**: Lee, Reddemreddy, Yan, Ghoneim, Fountain, Kamanga*

1. Motivation:

As the coronavirus pandemic continues to affect most parts of the world, people are forced to stay and work at home wherever possible. This has had a huge impact on the lifestyles of people, especially in their eating habits. More people have been cooking at home than the past. Though the measures have been relaxed in some parts of the world, some surveys conducted among consumers [R1] [R2] have found that the intention to keep up with home cooking is especially strong among younger demographics. People have reported that their cooking skills have improved and they are more confident in cooking themselves the food they like.

We would like to enable the younger demographics who are less experienced with apt recommendation of food recipes based on ingredients at hand. We intend to do this using the shared knowledge of recipes that are available on the internet. It is known by research that cooking skills directly contribute to health [3] of individuals. We would like our users to experiment confidently in the kitchen and would like to nudge them towards healthier living.

2. Problem Definition:

Today, there are many applications which enable home cooks to explore different recipes and follow laid out instructions to cook desired recipes. So far, applications mostly take only ingredients as an input and recommend a few recipes based on other user reviews [1], which has the problem of returning many non-applicable solutions in the context of the what the user has available at the moment as the algorithm rather focuses on past ratings. Furthermore, current solutions do not provide users with alternative ingredients based on their preferences, health concerns or food intolerances/ allergies [2]. Sometimes, it also becomes challenging to keep adding ingredients available randomly and look for desired recipes.

We intend to solve these problems by developing an application which provides the user a list of recipes which can be prepared using the ingredients available at hand. The novel recipe recommender uses alternate ingredients to find recipes and simulates the common tendency of professional cooks to use alternate ingredients which match certain characteristics of the ingredient being replaced in the original recipe. Lastly, we intend to develop an interactive UI which provides the user most commonly used ingredients based on the main ingredient. The user can then select the ones that are available and hence reducing the effort and improving the quality of interaction on the application in order to find a desired recipe.

3. Survey:

[5]: Mohammedali proposes a recommender based on a CSV file using Neo4j, which is applicable in our work since it supports multi-criteria searching and real-time querying. Further dataset manipulating is needed since graph database does not optimized for high-volume queries. [6]: Neo4j is one of the most popular graph databases and many useful training materials are available online [7]: Batra et al. combines data from various sources into a customized dataset. Our recommender would also perform data extracting and mapping from existing sources. This paper did the implementation in relational database hence transformation is needed. [8]: In their approach Cielen et al. proposes general development procedures for graph databases applying to recommendation engines. [9]: In their work Kuo et al. developed a recommender based on collected recipes from food.com which captures co-occurrence relationships between different recipes. [10]: The paper shows how to handle data discrepancies based on content-based filtering as we also need to parse and group different yet correlated data from stored recipes in our recommender. [11]: Feature selection and dimension methods such as PCA will be implemented to reduce the number of variables. Only those attributes which contribute the most to users' pick on recipes will be chosen and evaluated

in depth. Thus, some features will be muted out. [12]: Collaborative filtering works by matching a user to other users who've expressed similar preferences in the past. Its approach works by running content-based filtering on the results of collaborative filtering. [13]: Content-based Filtering does not require other users' data during recommendations to one user but relies on 'item-to-item' correlation to predict ratings on unseen items. [14]: Burke suggested a different approach by which content-based filtering is used to bootstrap collaborative filtering results while there is still not enough data. However, we believe that our approach in graph-based method adds more flexibility and modularity as a simple graph search task enables easy tuning, combination as well as alternation between the two recommendation approaches. [15], [16]: The similarity model defines similarity between recipes by common ingredients and proposes similarity measures such as cosine similarity. However, this approach will not be able to handle unknown words like new ingredients. [17]: Non-Negative Matrix Factorization reduces the dimensionality of filtering databases and improves the performance of collaborative filtering. The reconstructed matrix serves as a basis to the recommendation. It may have problems if the values are not independent. [18]: K-nearest neighbour graph search predicts the preferences of only the k-nearest neighbours and thereby shortens the execution time in collaborative filtering algorithm without sacrificing the level of accuracy. [19]: Yamagishi et al. proposed a recommender based on co-occurrence relation between ingredients used in the recipes part of the database. This will support our alternate ingredients algorithm as our database has extensive collection of recipes which could result in a superior quality outcome. One of the shortcomings is that all the ingredients, irrespective of whether they are primary ingredient, are included. [20]: Using naive Bayes filtering as mentioned in this paper, we intend to find an alternate ingredient to the one chosen by the user. We then find recipes which uses the alternate ingredient in place of user defined ingredient. [21]: It shows a similar attempt was made using deep learning, but the shortcomings are like one discussed on Yamagishi's approach. [22]: It shows the ingredients have relationships. It may add a feature that will have users being able to user alternate ingredients that have a strong co-relation with the ingredients in the recipe selected. [23]: It shows relationships between ingredients can also be clustered into flavour compounds to build a network. It plays a huge role in enhancing already existing recipes. This is by way of adding a certain taste, altering colour, adding nutritional value or even altering firmness of the food or mechanical stability. [24]: Smyth and Cotter have used both collaborative and content-based filtering to independently generate two recommendation sets which are then combined at a later stage.

4. Proposed method:

The proposed method is better than the state of the art in 3 aspects. Firstly, our recommendation engine uses collaborative and content-based filtering for recommendation rather than just text-based search. Secondly, our alternate ingredient algorithm suggests alternate ingredients for missing ingredients and lastly our interactive UI allows the users to select the most relevant ingredients based on its relevance to the main ingredient. These 3 innovative aspects of our application make our proposal unique. 3 key areas of the application are the database, the recommendation system and the user interface.

a. Database:

Neo4j, a graph database, is chosen as the database management tool for our implementation. Graph database is commonly adopted in recommendation engines due to its emphasis on node-node relationships. Neo4j is applicable in our work since it supports multi-criteria searching and real-time querying. The first phase of the work on the Graph database entailed fitting the raw ".csv" data into a data model that meets the requirements of the recommendation engine and the visualization. The recommendation engine requires a three-layer graph with User to User, Recipe to Recipe, and Recipe to Ingredient relationships between all three types of nodes. The basic data pipeline involved joining all the id fields to their requisite name fields through joins and normalizing the recipe and ingredient id fields into atomic single value per row format. We added additional fields such as calories, nutrition, steps taken etc. to meet the visualization requirements.

We decided to use "Food.com Recipes and Interactions" dataset published on Kaggle [25]. After processing, there were 178,265 recipe nodes, 8024 ingredient nodes, and 25,076 user nodes. We used the Neo4j admin import tool to load data after defining the nodes, edges, and properties through ".csv" headers. Figure 1 shows a small snippet of the database. Using the different approached that will be discussed in the next section, similarities between the users, recipes and ingredients are created and the same is updated on the db as relationships.

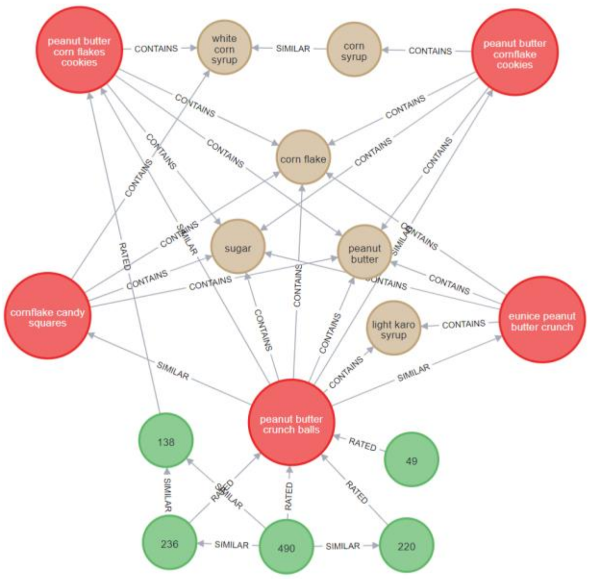


Figure 1: Snippet of the graph structure in Neo4j Database

b. Recommendation engine:

Our approach is to use a graph-based method of combining collaborative and content-based filtering incorporating recipe-to-recipe correlation & user-to-user correlation. Our approach comprises of three stages.

In the First stage we define how similar users are to each other based on their ratings on recipes. As shown in below matrix, each row represents the rating history of each user on each recipe. Out of 25,076 unique users, 'user number 94' rated the most recipes, namely 6,437 out of 178,265 recipes in total, while there are more than 4,000 users who rated only 2 recipes.

	rec 1	rec 2	rec 3	....	rec 178265
user 1	4	2	0	....	4
user 2	0	5	1	....	4
user 3	3	5	2	....	3
....					
user 25076	1	4	0	....	1

Then, we can identify similarities between users using a 'cosine similarity measure'. Assuming each row in the user-recipe matrix is a separate vector, similarities of two vectors (x, y) will be computed as follows:

$$sim(x,y) = \frac{x \cdot y}{\| x \| \| y \|}$$

$$where \ \| x \| = \sqrt{x_1^2 + x_2^2 + \cdots + x_p^2}.$$

The result of pairs of vectors from the user-recipe rating matrix, we can get a similarity matrix of which size equals to $(Number\ of\ users \times Number\ of\ users)$. The similarity scores will be used to predict preferences of similar users. A known issue with this method is the cold start problem [4], where we don't have a user's history.

In the second stage we represent recipes as bag-of-words vectors using their actual ingredient information. The output of this step is a sparse MxN matrix where M is the number of recipes and N is the number of ingredients across all recipes. In our datasets M = 178,265 and N = 8024.

	ingr 1	ingr 2	ingr 3	....	ingr 8024
rec 1	1	0	1	....	1
rec 2	0	0	1	....	0
rec 3	1	1	0	....	1
....					
rec 178265	1	1	0	....	1

Then we compute the similarity between recipes by applying Jaccard Similarity measure, generating a symmetric MxM matrix where each cell represents the Jaccard similarity score between each two corresponding recipes.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

$$where\ A, B\ are\ recipe\ vectors$$

In the Third stage we create an ingredient-to-ingredient similarity matrix.

Word embedding represents words using dense, semantically-meaningful N-dimensional vector space (where N refers to the dimensions of the vector) such that words that share common contexts and semantics are located in close proximity to one another in the space. The word vectors are chosen carefully such that a simple mathematical function (cosine similarity) indicates the level of semantic similarity between the words represented by those vectors.

Word2Vec is one of the most popular techniques to learn word embeddings, it was developed by Tomas Mikolov in 2013 at Google [26]. The Word2vec algorithm uses a shallow neural network model to learn word associations from a large corpus of text with the objective of maximizing the probability of the next words given the previous words.
For our ingredient similarity algorithm, we used Spacy's pre-trained word2vec model [27]. Spacy is a natural language processing library for Python designed to have fast performance, and with word embedding models built in. The large English language word2vec model in Spacy is trained on a huge corpus of text such as blogs, news and comments [28], and contains 685K static vectors of 300 dimensions each.

Firstly, using the word2vec model we represented the 8K unique ingredients in our dataset into an 8000x300 dense matrix of ingredient representations.

	dim 1	dim 2	dim 3	...	dim 300
ingr 1	0.03036	-0.02487	0.06298	...	-0.32548
ingr 2	0.15442	0.01939	-0.24432	...	-0.00124
ingr 3	0.05523	0.19296	0.08134	...	0.0145
...					
ingr 7993	0.22145	-0.02458	0.021458	...	0.1247

Secondly, we computed the pairwise cosine similarity between all ingredients in our matrix, hence extracting the top 3 most similar ingredients for every ingredient in our database. The similar ingredients were then used to enhance the user experience through recommending alternative ingredients in case the user wants to try a recipe but does not have all required ingredients.

$$Cosine\ sim(x, y) = \frac{x \cdot y}{\| x \| \| y \|}$$

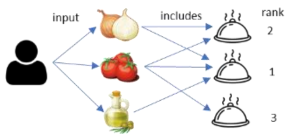
(Pick top 3 similar ingredients)

we model recipes, ingredients and users in an extended graph (shown in Figure 1) where nodes represent users, ingredients and recipes, while edges represent similarities/correlations between them.

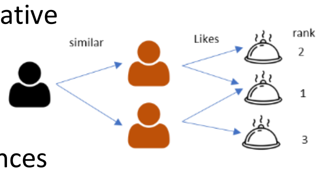
Lastly, we generate a complementary ingredient selection query in case the user wants to add more ingredients which are relevant and available. The query basically searches for the most commonly occurring ingredient for the currently selected ingredient set which is ranked and presented to the users. The user can then select the available ingredients and once done, the application fetches the relevant recipes.

The sequence of activities from user input to recipe recommendation is illustrated in Figure 2.

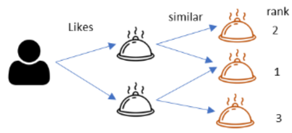
Step 1: We first find the matching recipes in our database purely based on ingredient match.



Step 2: We use collaborative filtering to get another set of recipes which will be ranked based on number of occurrences within the user network.



Step 3: We use content-based filtering to find recipes based on the liking of the user and its similarities which is again ranked.



Step 4: Complementary ingredient selector and alternate ingredients will be presented separately in the UI.

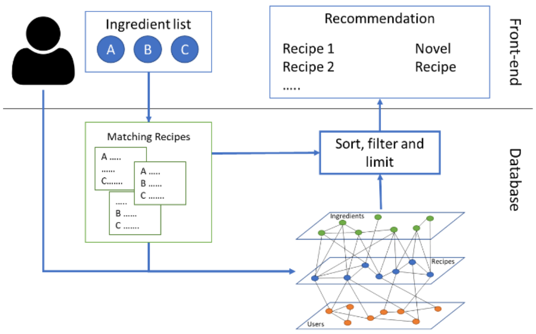


Figure 2: Flow chart of the application

c. User Interface:

The UI for our recipe recommender is user-based, in which different users can login to their account and search for recipes. The workflow of the UI is described in the following. User can login with his/her username and password in the beginning, this helps us to retrieve the user review records to optimize the search result. As soon as the user login, he/she will reach the main page of the recommender. It shows the short cuts to the user's record, as well as the input fields for the main ingredient and side ingredients. User can decide to have the search result in graph/table presentation by selecting "Visual/Text" search option.

For the text search, users can input their main and side ingredients. By selecting one of the recipes, a details page is shown which displays all the information, such as steps, ingredients, nutrition details, tags and rating details, of the recipe. If the user decided not to use this recipe, he/she can return to search result and select another recipe.

For the visual search, users can view the relevant recipes for the given ingredients in graph with nodes and edges. Users can also filter the recipes according to their review history or similar users' reviews, which adopted our content-based and collaborative filtering approach respectively. In additions, users can go to the additional ingredients tab to add complementary ingredients to optimize the search result. By selecting one of the recipe's node, the graph will be updated with the selected recipe connected with its related ingredients or user ratings. A tooltip will be presented top 3 similar ingredients as alternative on each ingredient node, similar tooltip is also displayed on each rating node with the user id and rating score. Finally, when users double-click the selected recipe node, the same details page for the recipe will be shown.

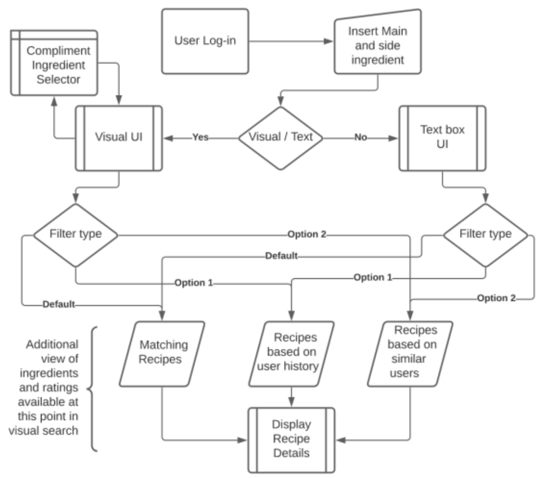


Figure 3: User Journey Chart of the Application

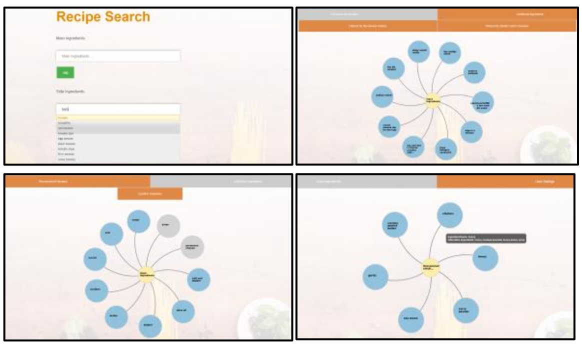


Figure 4: Screenshots of the UI in action

[Click here for a Demo of our application (2min. video)]()

5. Experiments / Evaluation:

In order to test our application which is on device and not hosted online, we had to take 2 approaches. Firstly, a survey is conducted to determine whether the recommendations suggested by our engine are apt. The survey contained 2 sections. Section 1 covered 3 questions related to the alternate ingredient algorithm. We framed 3 random outputs from our algorithm into questions where the respondents could judge subjectively if they thought that the suggestion was apt or not. They were asked to rate the suggestion on a scale of 1 to 10. Similarly, we developed 3 questions to get a feedback on the content-based similarity algorithm. Figure 5 shows a question from each section.

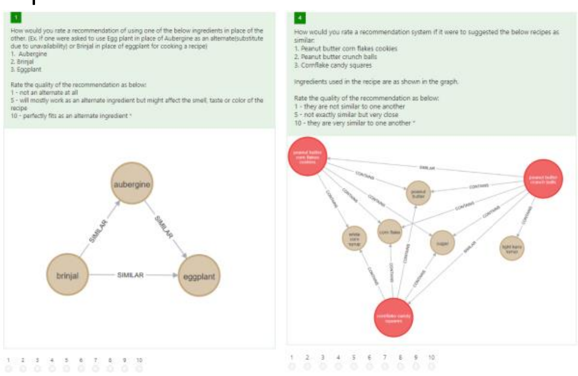


Figure 5: Few questions from the Survey conducted

The survey results includes responses from 41 respondants. Figure 6 shows the plot of the ratings. The results as shown is slightly above average and points out that the algorithm need to do much better to get better user satisfaction. While analyzing the results, it was clear that some of the recommendations had very high score and some lower. Some improvement measures will be discussed in the conclusion section.

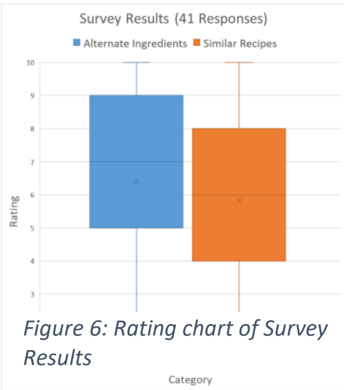


Figure 6: Rating chart of Survey Results

Secondly, we also conducted a UX survey for the application but the sample size was very low as the survey was taken by family members of the team. Figure 7 shows the survey questions as well as the results. Responsiveness of the application was specifically pointed out as a an area of improvement. Most of the users were delighted to use our visual search and thought it was very innovative and intuitive.

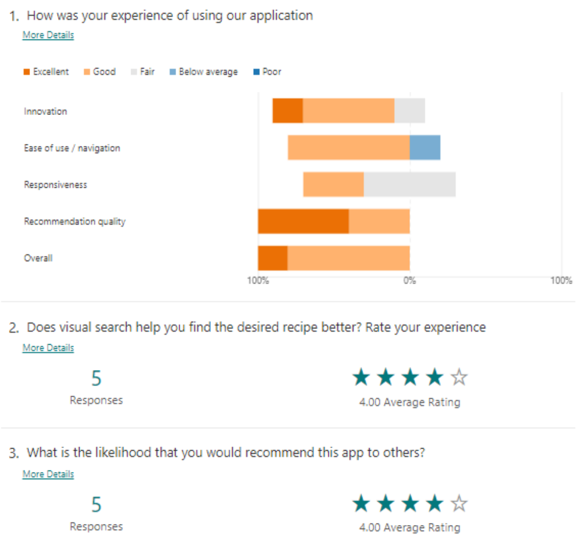


Figure 4: Overview of the results from the UI survey

6. Conclusion and discussion:

In summary, our team have built an application from scratch using new technologies such as graph database (neo4j), flask framework, D3 for visualizations accompanying with various filtering and similarity algorithms to make recipe recommendations.

While the project was very successful with a very robust database, interactive UI and a coherent recommendation engine, there is still a lot of room for improvements.

Firstly, we couldn't complete the novel recipe recommender which would use our alternate ingredient information to create novel recipes. The algorithm we envisioned was technically very challenging and it needed a lot of compute power to reach optimized results which was not at our disposal. Additionally with 1 non-contributing team member, we couldn't pursue it beyond the first failed attempt. We could still just replace the main ingredient with our alternate ingredient and make a search but the output would be suboptimal and the team opined not to take that route.

The recommendations have a lot of room for improvement and the feedback from the survey that was discussed clearly states the same. Our dataset includes many additional fields such as nutrition values, tags and ratings which were not included in our algorithms. Including them can improve the quality substantially. Additionally, categorizing the ingredients into actual ingredients and seasoners (salt, water, pepper etc) will help improve the quality of the recommendations. There are several papers which discusses strategy to do the same.

The responsiveness of the application can further be improved by using neo4j indexes for search instead of node properties / labels. We took this approach at the end of the project for some of the queries which took more time but we did not improve all the cypher queries in this manner.

The application itself can be improved to profile the users better. This was not in scope for our project and hence we used existing user data from the dataset. The application can also be improved by linking pictures for the ingredients and recipes from the web (pictures were not part of our dataset).

As a team we see a lot of scope to improve the application and ultimately meet our purpose of enabling every user to cook healthy meals on their own and to lead a healthy lifestyle in these difficult times.

7. Project plan:

Below is the activity plan and all the members contributed equally during all phases of the project.

	Timeline	Lee	Ghoneim	Reddy	Kamanga	Yan	Fountain
Phase-I	W1-W3	Reco. Engine		Alternate ingredient		Dataset & database	
Phase-II	W4-W7	Recommendation Engine			UI		Database
Phase-III	W8-W11	Alternate Ingredient Algorithm			UI		

Unfortunately, Anton Sodia who is part of our team was unable to contribute due to personal issues for most part of the project. After consultation with Prof. Polo, members of Team034 and Anton, we have decided to allocate 35% of the project grades to Anton (Proposal and Progress report) and no credit for the final project submissions which is equivalent to 65%.

References:

[1] Jagithyala, Anirudh. “Recommending recipes based on ingredients and user reviews.” 2014.

[2] Meng Chen, Xiaoyi Jia, Elizabeth Gorbonos, Chinh T. Hoang, Xiaohui Yu, Yang Liu. "Eating healthier: Exploring nutrition information for healthier recipe recommendation". Information Processing & Management 2020; 57(6):102051.

[3] M. Stead, M. Caraher, W. Wrieden, P. J. Longbottom, Karen Valentine, A. Anderson. "Confident, fearful and hopeless cooks: Findings from the development of a food-skills initiative". British Food Journal 2004; 106:274-287.

[4] Fayyaz, Z, Ebrahimian, M, Nawara, D, Ibrahim, A, Kashef, R. "Recommendation Systems: Algorithms, Challenges, Metrics, and Business Opportunities". Applied Sciences 2020; 10(21):7748.

[5] Mohammedali, N. "Recommendation System Based on Graph Database Techniques". International Research Journal of Engineering and Technology 2019; 6(10):754-763.

[6] Lal, M. "Neo4j Graph Data Modeling". Packt Publishing 2015.

[7] Batra, D, Diwan, N, Upadhyay, U, Kalra, J, Sharma, T, Sharma, A, Khanna, D, Marwah, J, Kalathil, S, Singh, N. "Recipedb: A resource for exploring recipes". Database 2020; 2020.

[8] Cielen, D, Meysman, A, Ali, M. "Introducing Data Science: Big Data, Machine Learning, and more, using Python tools". Manning Publications 2016.

[9] Kuo, FF, Li, CT, Shan, MK, Lee, SY. "Intelligent menu planning: Recommending set of recipes by ingredients". In Proceedings of the ACM multimedia 2012 workshop on Multimedia for cooking and eating activities 2012; 1–6.

[10] Van Meteren, R, Van Someren, M. "Using content-based filtering for recommendation. In Proceedings of the machine learning in the new information age". MLnet/ECML2000 workshop 2000; 47–56

[11] Lei, SA. Feature Selection Method Based on Information Gain and Genetic Algorithm. In Proceedings of the 2012 International Conference on Computer Science and Electronics Engineering; 02(2012): 355–358

[12] J. Salter, N. Antonopoulos. "CinemaScreen recommender agent: combining collaborative and content-based filtering". IEEE Intelligent Systems 2006; 21(1):35-41.

[13] Salter, J, Antonopoulos, N. "CinemaScreen Recommender Agent: Combining Collaborative and Content-Based Filtering". 2006; 21(1):35–41.

[14] Burke, R. "Integrating knowledge-based and collaborative-filtering recommender systems". In Proceedings of the Workshop on AI and Electronic Commerce 1999; 69–72

[15] Haifeng Liu, Zheng Hu, Ahmad Mian, Hui Tian, Xuzhen Zhu. "A new user similarity model to improve the accuracy of collaborative filtering". Knowledge-Based Systems 2014; 56:156-166.

[16] R. Ji, Y. Tian, M. Ma. "Collaborative Filtering Recommendation Algorithm Based on User Characteristics". In 2020 5th International Conference on Control, Robotics and Cybernetics (CRC) 2020; 56-60.

[17] Y. Kim, S. Choi "Weighted nonnegative matrix factorization". In 2009 IEEE International Conference on Acoustics, Speech and Signal Processing 2009; 1541-1544.

[18] Y. Park, S. Park, S. Lee, W. Jung. "Fast Collaborative Filtering with a k-nearest neighbor graph". In 2014 International Conference on Big Data and Smart Computing (BIGCOMP) 2014; 92-95.

[19] Ryosuke Yamanishi, Naoki Shino, Yoko Nishihara, Junichi Fukumoto, Aya Kaizaki. "Alternative-ingredient Recommendation Based on Co-occurrence Relation on Recipe Database". Procedia Computer Science 2015; 60:986-993.

[20] P. Valdiviezo-Diaz, F. Ortega, E. Cobos, R. Lara-Cabrera. "A Collaborative Filtering Approach Based on Naïve Bayes Classifier". IEEE Access 2019; 7:108581-108592.

[21] Bień, M, Gilski, M, Maciejewska, M, Taisner, W. Cooking recipes generator utilizing a deep learning-based language model (Doctoral dissertation).

[22] Chun-Yuen Teng, Yu-Ru Lin, Lada A. Adamic. "Recipe recommendation using ingredient networks". 2012:3-6.

[23] Yong-Yeol Ahn, Sebastian E. Ahnert, James P. Bagrow, Albert-Laszlo Barabasi1. "Flavor network and the principles of food pairing". 2011:2-4.

[24] Cotter, P, Smyth, BPtv: Intelligent personalised tv guides. In AAAI/IAAI 2000; 957–964.

[25] Shuyang Li. Food.com Recipes and Interactions.

[26] Mikolov, Tomas and Sutskever, Ilya and Chen, Kai and Corrado, Greg and Dean, Jeffrey. “Distributed representations of words and phrases and their compositionality”. arXiv:1310.4546

[27] Spacey-models. https://github.com/explosion/spacy-models/releases

[28] Weischedel, Ralph, et al. OntoNotes Release 5.0 LDC2013T19. Web Download. Philadelphia: Linguistic Data Consortium, 2013

[R1] https://www.bloomberg.com/news/articles/2020-07-07/newly-minted-home-chefs-mark-another-blow-to-u-s-restaurants

[R2] https://www.consultancy.uk/news/25412/cooking-at-home-becomes-major-trend-coming-out-of-covid-19

Survey Questionnaire:

Survey - Recipe Recommender Application

A short survey to get user feedback on the quality of the recommendation from our recommendation application. This application was developed as a part of project work for one of the courses in Masters program of Georgia Tech.

* Required

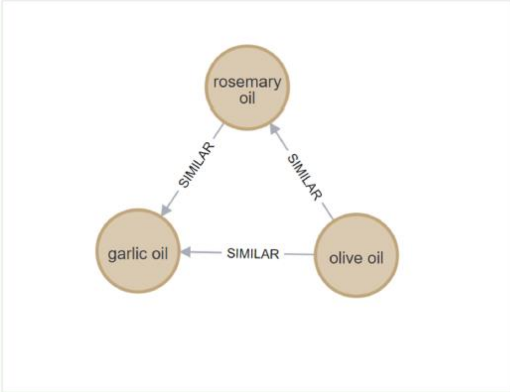
Alternate Ingredient Recommendations

Below is a list of few random recommendations given by our recommendation engine for alternate ingredients. We use this engine to suggest to the user alternate ingredients that could be used to cook a recipe when an ingredient is unavailable.

2

Similar to the previous question. Do you think these ingredients are similar:
1. Olive Oil
2. Garlic Oil
3. Rosemary Oil

Rate the quality of the recommendation if one of these were given as an alternate to the other while cooking
1 - not an alternate at all
5 - will mostly work as an alternate ingredient but might affect the smell, taste or color of the recipe
10 - perfectly fits as an alternate ingredient *

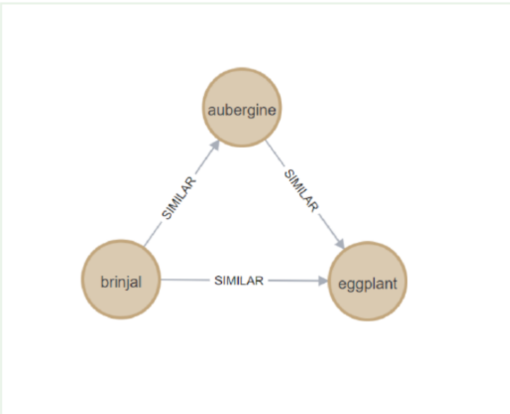


1 2 3 4 5 6 7 8 9 10

1

How would you rate a recommendation of using one of the below ingredients in place of the other. (Ex. If one were asked to use Egg plant in place of Aubergine as an alternate(substitute due to unavailability) or Brinjal in place of eggplant for cooking a recipe)
1. Aubergine
2. Brinjal
3. Eggplant

Rate the quality of the recommendation as below:
1 - not an alternate at all
5 - will mostly work as an alternate ingredient but might affect the smell, taste or color of the recipe
10 - perfectly fits as an alternate ingredient *

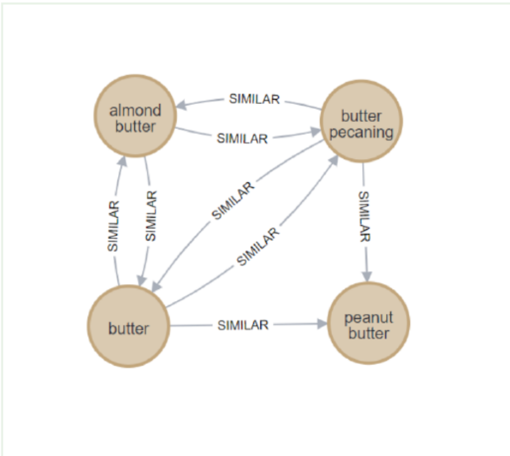


1 2 3 4 5 6 7 8 9 10

3

Similar to the previous questions. Do you think these ingredients are similar:
1. Butter
2. Peanut butter
3. almond butter
4. Butter pecaning

Rate the quality of the recommendation if one of these were given as an alternate to the other while cooking
1 - not an alternate at all
5 - will mostly work as an alternate ingredient but might affect the smell, taste or color of the recipe
10 - perfectly fits as an alternate ingredient *



1 2 3 4 5 6 7 8 9 10

Similar Recipe Recommendations

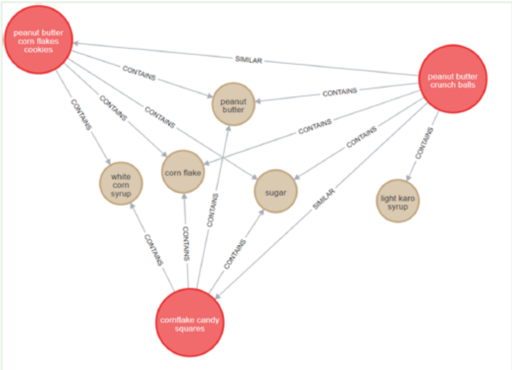
Below is a list of few random recommendations given by our recommendation engine for similar recipes. We use this engine to suggest to the user recipes that are similar to ones the user has already consumed before and rated. This helps us to rank recipes which are closer to the liking of the user higher than the others.

4

How would you rate a recommendation system if it were to suggested the below recipes as similar:
1. Peanut butter corn flakes cookies
2. Peanut butter crunch balls
3. Cornflake candy squares

Ingredients used in the recipe are as shown in the graph.

Rate the quality of the recommendation as below:
1 - they are not similar to one another
5 - not exactly similar but very close
10 - they are very similar to one another *



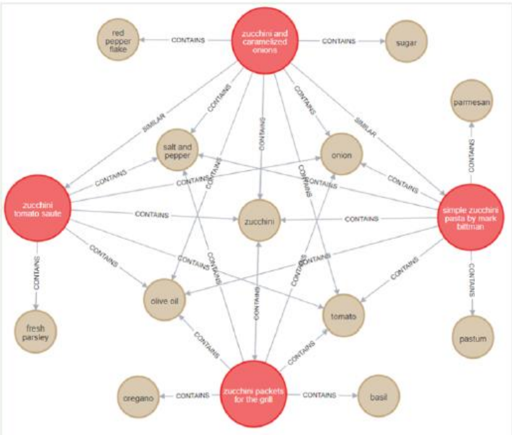
1 2 3 4 5 6 7 8 9 10

5

Similar to the previous question, do you think these recipes are similar:
1. Zucchini and caramelized onions
2. Simple zucchini pasta
3. Zucchini packets for the grill
4. Zucchini tomato saute

Ingredients used in the recipe are as shown in the graph.

How would you rate the recommendation system if it suggested these recipes as similar
1 - they are not similar to one another
5 - not exactly similar but very close
10 - they are very similar to one another *



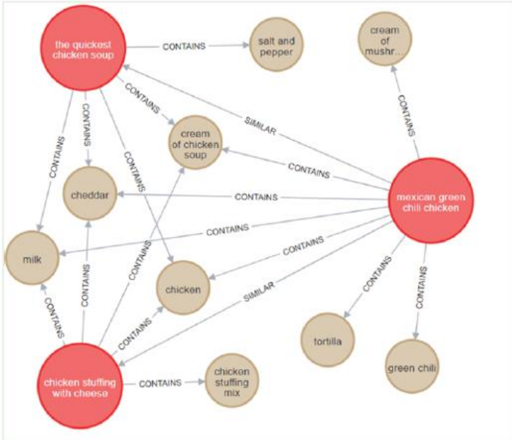
1 2 3 4 5 6 7 8 9 10

6

Similar to the previous question, do you think the below recipes are similar to Mexican green chilli chicken:
1. Chicken soup
2. Chicken Stuffing with cheese

Ingredients used in the recipe are as shown in the graph.

How would you rate the recommendation system if it suggested these recipes as similar
1 - they are not similar to one another
5 - not exactly similar but very close
10 - they are very similar to one another *



1 2 3 4 5 6 7 8 9 10

Thank you for your feedback !

Regards - Team 034 - Recipe Recommender