ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/362941097

An update on cooking recipe generation with Machine Learning and Natural Language Processing

Conference Paper · June 2022

DOI: 10.1109/AIC55036.2022.9848929

CITATIONS
2

READS
1,160

2 authors:

Nikolaos-Ioannis Galanis
International Hellenic University
4 PUBLICATIONS **11** CITATIONS

SEE PROFILE

George Papakostas
Democritus University of Thrace
235 PUBLICATIONS **3,752** CITATIONS

SEE PROFILE

All content following this page was uploaded by George Papakostas on 10 May 2023.

The user has requested enhancement of the downloaded file.

An update on cooking recipe generation with Machine Learning and Natural Language Processing

Nikolaos-Ioannis Galanis
MLV Research Group
Department of Computer Science
International Hellenic University
Kavala 65404, Greece
nigaean@cs.ihu.gr

George A. Papakostas
MLV Research Group
Department of Computer Science
International Hellenic University
Kavala 65404, Greece
gpapak@cs.ihu.gr

***Abstract*—Cooking is a part of most peoples' daily routine. Yet there's always the same question: "What are we going to cook today?". We will attempt to provide a roundup of Machine Learning approaches whose goal is to provide an answer to that question with recipe generation by using as input just a list of preferred ingredients - or even generating them as well!**

***Index Terms*—Cooking, Machine Learning, Recipe Generation, Natural Language Processing, NLP, Case Based Reasoning**

I. Introduction

Food is essential for human survival, thus making cooking a very important task. Though it sounds like a simple one, it's rather complicated since additional factors are added according to the individual person's needs regarding the taste, caloric content, dietary restrictions and even lifestyle choices, like veganism. And it gets even harder if we add to that the fact that the production of a truly edible dish requires a specific skill set.

Cooking has become nowadays something more than just a biological need. For some it's just another hobby, for others it's a profitable profession and with the prevalence of social media food now has to look great and not just taste good. Furthermore it must be up-to-date regarding the latest trends on materials, flavor combinations and many other factors. It has become so complex that a whole new field emerged under the name "Food Computing", researching every possible aspect - with Machine Learning being a great part of it.

There are attempts to apply Artificial Intelligence in its early stages going back as far as 1986 with CHEF [1], trying to generate new recipes based on existing ones for the Szechuan cuisine. Up until now that's an ever evolving process, taking a leap from simple pattern matching to big data mining with IBM's Chef Watson [2] and Deep Neural Networks.

There's a wide range of tasks as well, from image recognition attempting to reconstruct recipes or calculate the dish nutrients from photographs, to equally complex text generation tasks regarding either just the ingredients or even whole recipes. In the following chapters we'll be focusing on the latter tasks, and the proposals on how to produce new recipes according to the user's needs, whether it's by adapting existing ones or by creating entirely new suggestions. We will first go through a summary of the individual methods and some commonly used datasets and then proceed to a discussion about the issues of the field that still remain.

II. Related Work

There have been some other attempts that refer to developments in the field of cooking and recipes, but they are mostly focusing on other types of tasks or have a more generalized approach. Burca [3] published an overview of cases combining traditional Case Based Reasoning with other Machine Learning approaches, with recipe generation being a small part of it. Trattner [4] compiled a review of food recommendation techniques. Min [5] on the other hand created a review on food computing as a whole "industry", without focusing in any of the specific subtasks.

In this review the ideas that attempt to tackle the problem of recipe generation within the last decade will be summarized, whether it's recipes as a whole with ingredients and execution instructions or just parts of them.

III. Materials and Methods

A. Selecting the appropriate cooking literature

The search for candidate papers begins at Google Scholar and Scopus, using the search term (cook OR recipe OR cocktail) AND (machine OR deep OR nlp OR "natural language processing"), focusing roughly on the last 10 years. After filtering out some initial selections regarding the evolution of recipe generation, the collection was expanded by branching into references and citations of them.

Many of the CBR systems described in the respective section were developed for the Computer Cooking Contest, a contest that aims at comparing CBR system results on cooking as an event that was part of the annual International Conference on CBR [1] - thus making the Conference proceedings another great source.

B. The "Traditional" approach: Case-Based Reasoning

Case-Based Reasoning (CBR) [6] is the adaptation of existing knowledge in order to find a solution to a new problem. Systems with such functionality are based on a

[1]https://iccbr21.org/

Case Base (a collection or database of recipes, in this case), searching for data according to the user's constraints which are usually defined in natural language. When no matches are found, the system attempts to create a new suggestion by replacing ingredients, combining its Domain Knowledge (DK) - commonly a tree relating flavors, ingredients or dish types - and Adaptation Knowledge (AK) - basically the "matching algorithm". The individual representations of the collection entries can be Structured (stored as entities-classes), Textual (stored as free text), or Conversational (stored as series of questions-answers) [7].

Taaable [8] creates a hierarchical tree as seen in Fig.1 with a recipe as its root, with ingredients linked to related classes (regarding cuisine, diet, type of ingredient etc.) and a cost for each edge of the tree. Then, when adaptation is required, it is performed either by using the tree directly combined or not with specific AK rules, or by using nutritional information for individual ingredients as additional variables. Along with the main system, an openly available wiki was generated by the name of WikiTaaable [9] to improve the knowledge acquisition process, a project that was further used by other cooking CBR projects as well.

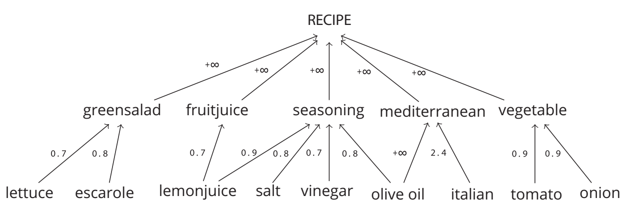

Fig. 1. A sample hierarchical tree from Taaable [8], a concept used by most CBR concepts.

JaDaCook [10] was created using Java, with a straightforward K-Nearest Neighbours approach. It extracts the requested ingredients from the input, rates the recipes of the case base in a normalized [0,1] range depending on whether the ingredients are contained in the title or instructions of each of them, and then selects the top ranking ones. A 2.0 version [11] and a web version [12] were introduced later, with improvements not so much for the core algorithm, but for the interface and its database of ingredients and recipes.

CookIIS [13] is a web based system created with JSP. For the case representation, it's using Structured representations of the recipes, classifying ingredients into specific taxonomies, creating essentially a tree with specific costs for its traversal, and a table of some direct taxonomy relations with predefined costs overriding the ones resulting from the tree. Summing up the weighted relation scores for the ingredients ends up in a similarity score for each recipe in the case base. In order to adapt a recipe the system either replaces missing ingredients with the highest similarity score, or by using crawled data from cooking community websites.

CookingCAKE [14] is also a web based system created with PHP. In order to calculate similarity of ingredients and recipes, they are generalized in specific classes resulting in a tree, like other CBR systems. The similarity score is based on the individual nodes score, which is calculated by using the depth of each node and the height of the tree for each ingredient. A Naive Bayes classifier is used to classify recipes and ingredients by type of meal and cuisine whenever that's required. A latter version of it [15] introduced the usage of and workflows [16] and POQL [17], a language that was introduced to provide a structured representation for the queries of process-oriented case based reasoning tasks [18], attempting to adapt the workflow task as a whole or the sub-tasks and not only the recipe ingredients. An example of such workflow can be seen in Fig. 2

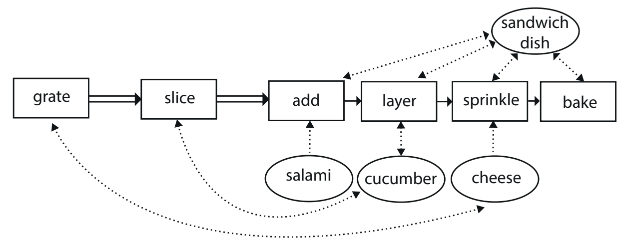

Fig. 2. A workflow example used by CookingCAKE [15].

ColibriCook [19] is actually comparing the input ingredients with the ones contained in recipes of the Case Base, attempting to return recipes that match and using WEKA [20] for classifying accordingly when a cuisine or diet restriction is selected.

Goethe-Shaker [21] was created to provide recipes too, yet not for cooking but for cocktails. The ingredients are being scored three ways: their co-occurrence in recipes, the number of edges linking to an ingredient while also being classified in categories for its profile (e.g. sour, sweet etc.). Then the Case Base candidates with the most wanted ingredients defined in the input are selected, and if there are unwanted or extra ingredients they're replaced according to these scores.

C. The "Modern" approach: (Deep) Neural Networks

Generating new recipes is mostly a text generation task. As such, the usage of certain types of architectures would be expected: generative models like the Variational Autoencoder [22] (presented in its simplest form in Fig. 3), Recurrent Neural Networks and LSTMs that can maintain the context of the text being generated, or even the more recent Transformers [23] that excel in multiple NLP tasks. WHile other generative models like GANs are used in similar tasks, they were not included since they're mostly used to generate food images from their recipe, and the task at hand is to generate lists of ingredients or actual cooking instructions. Many of the solutions in this section do not provide an "end-to-end" recipe but only part of it - i.e. just the ingredients, with or without required measurements, or only cooking instructions).

Bostan [24] handles the recipe generation as a translation task, attempting to "translate" the input ingredients to the recipe itself. After creating the word embeddings of the ingredients, they are passed through a **seq2seq architecture**

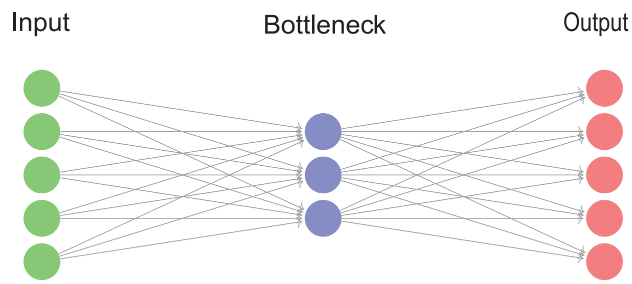

Fig. 3. A simple Autoencoder architecture, commonly used for generative tasks.

[25] of 2 LSTMs, to produce cooking steps for the target recipe.

Another similar approach [26] attempts to generate a recipe from a set of ingredients, with a seq2seq model combined with an attention mechanism [27]. It consists of an encoder that turns a series of ingredient embeddings into a representation of the input recipe, and a decoder that converts these to a series of cooking instructions.

Gona [28] suggests a combination of **bi-directional LSTMs** that attempts to find recipes fit for the user's taste and diet, while also generating new ones based on them. The first part consists of three LSTM classifiers: One with an output of 0.5 or higher when the user likes the recipe, a second one where acidic recipes get a value of 0.5 or higher, while the third one outputs a value in the range [0,1] depending on how high the caloric content is. The 3 outputs are combined through a circuit of logic gates, resulting in a 1 or 0 depending on whether the recipe is suitable for the user or not. The suitable recipes are then passed through a Variational Autoencoder, producing new ones similar to the initial input.

Yu [29] proposed a method where an **LSTM encoder** is used to get a representation of the input ingredients which are then passed through a routing module [30]. By calculating the category vectors and routing weights for each of them, they are clustered into the categories ”Low Sugar”, ”High Fiber”, ”Low Fat”, ”Grilling” and ”Frying”. Then, by using an attention-based [27] **LSTM decoder**, the final recipe is generated.

Majumder [31] describes a model that receives a recipe name, a list of ingredients and a desired caloric level at its input. These are passed through a **two-layer bidirectional GRU** [32], and then attention is applied on the GRU's output. Finally, another two-layer GRU is used as a decoder producing the recipe output, while also using a secondary attention mechanism for the current user's previous recipes.

Aljbawi [33] attempts to create a model partially based on **Transformers**. The candidate ingredients are selected in three different ways: by calculating the co-occurrences of ingredients in the input dataset, by using a small neural network with 3 hidden layers, or by fine tuning a pre-trained BERT model [34] for the Masked Language Model task - where one ingredient is masked in predefined ingredient lists. Then the list is filtered by caloric content, in order to filter out the most healthy ones. Finally, the selected ingredients are used as input to a GPT-2 model [35] in order to generate the text for the actual recipe.

In another similar proposal [36], a **GPT-2** model was fine tuned using a combination of crawled data sources along with the introduction of additional control tokens in order to generate recipes directly from the user's input ingredients.

Lastly, **RecipeGPT** [37] is a web based application with an underlying **GPT-2** model that was fine tuned using the Recipe1M+ dataset [38] in order to generate either ingredient lists from a title, or execution instructions when a title and the ingredients are provided. A sample recipe generated by the model can be seen in Fig. 4

Ingredients
Salt,Baking soda,Eggs,Chocolate chips,Bananas,Sugar,Flour,Vanilla,Butter
Instructions

1) Preheat oven to 350f
2) Grease and flour a 9x5x3 inch loaf pan and set aside
3) Cream butter and sugar until light and fluffy
4) Add eggs one at a time, beating well after each addition
5) Stir in vanilla and mashed bananas
6) In a separate bowl, combine flour, salt, baking soda, and chocolate chips
7) Add flour mixture into creamed mixture, stirring until just combined
8) Pour into loaf pan
9) Bake for 60 to 70 minutes, until a wooden pick inserted near center comes out clean
10) Cool in pan for 10 minutes, remove and cool completely on a wire rack

Fig. 4. A sample recipe generated with RecipeGPT [37] given the title “delicious double chocolate banana bread”.

D. The “Hybrid” approach: Best of both worlds?

Q-Chef [39] combines CBR with a Variational Autoencoder in a dual-cycle implementation, attempting to bring the best of both approaches. During the first cycle the input is forbidden and required ingredients along with a desired level of “surprise”, producing multiple sets of ingredients and calculating the surprise and plausibility of each set. Then one of these sets is passed into the second step, picking a similar recipe from the case base and adapting the ingredients accordingly, providing though only a set of ingredients while leaving the amounts and preparations steps required as future work.

E. What about the data?

Many proposed models come along with their own self-scraped datasets from cooking websites, usually openly available to any researcher. They are usually though pre-processed according to each project's needs, rendering them unusable for significantly different projects.

Kaggle [2] is a common source for datasets, though usually rather small in size but diverse enough to train and test Proof of Concept models and solutions.

RecipeQA [40] is one of the smaller sets of about 20.000 cooking recipes and 36.000 question-answer pairs. Though as its name suggests it's mostly targeted on Question Answering tasks, the recipes are well structured with separate ingredients and instructions and can be used for other types of applications as well.

[2]https://www.kaggle.com/

Recipe1m+ [38] is a dataset of over 1 million recipes and 13 million food images, along with relations between images and recipe texts - making it one of the most commonly used ones for food related tasks. Based on it, **RecipeNLG** [41] took it one step further by adding over 1 million new recipes while focusing on the actual recipe texts rather than their links to images.

IV. Discussion

Though recipe generation and suggestion has come a long way, there's still no solution that could be completely reliable and there are some important issues to be addressed.

A. Maintaining the analogies

Most suggestions are based on recipe adaptation in order to produce their output results, with a quite often direct replacement of ingredients. While that might seem like a reasonable idea, the new ingredient might have significantly different properties (e.g. acidity, texture, etc.) resulting in a non plausible recipe. This issue is even worse when there are other limitations as well, like for instance diet needs where the caloric content of the ingredients is important. Some of the methods attempt to resolve this problem by adjusting the quantity as well and not just the ingredient itself depending on nutritional information, but a more "holistic" approach might be required that adapts the whole ingredient list and not just replacing some of them.

B. Cultural diversity & ingredient combinations

Sometimes the ingredients aren't a good combination, either because their individual flavor profiles are not a good match, or just because it's an unexpected taste. For instance, an Asian recipe might be very surprising for a western country and vice versa. That "surprise" might even be a requirement in some cases, but the user usually expects a meal with a familiar taste. While this issue cannot be resolved for all cases, filtering out the input recipes according to the user's taste - whether it's from a case base or training material for a model - could help.

C. Incoherent recipes

Sometimes the ingredient lists or the cooking steps that are described might not make sense, usually when the output is a brand new recipe from a generative model and not just an adaptation. For instance, it might be for a weird ingredient combination or a cooking step that just can't be performed - like "cut the water in pieces". While this is an issue of Natural Language Processing and text generation in general and it might not be as important in other tasks like automated translation where the user can actually understand the content based on the overall context, in this case the problem is crucial since the output recipe can't be executed at all.

V. Conclusion

The search for the "perfect recipe" looks like an ongoing quest, both for amateurs and professionals in the kitchen. With the rise of social media and pop culture, cooking is one of the subjects currently on the spotlight and everyone is trying to create innovative, tastier, healthier, easy to make recipes.

Applying Machine Learning for the task can prove helpful, whether it's to find the right recommendations within huge databases, adapting them, or even producing exciting new combinations of ingredients. The process might not have the desired results for all possible cases - and it can be hilarious at times - but it can definitely provide the context and inspiration to use as a foundation for the actual solution.

And while AI seems to have a long way to create a "robotic chef", using Deep Learning might the thing that will bring us a step closer to that. With NLP going through its "Golden Age", and recipe generation being a subgroup of that field, it remains to be seen what the next leap will bring.

Acknowledgement

This work was supported by the MPhil program "Advanced Technologies in Informatics and Computers", hosted by the Department of Computer Science, International Hellenic University, Kavala, Greece.

References

[1] K. J. Hammond, "Chef: A model of case-based planning." in *AAAI*, vol. 86, 1986, pp. 267–271.

[2] L. R. Varshney, F. Pinel, K. R. Varshney, D. Bhattacharjya, A. Schörgendorfer, and Y.-M. Chee, "A big data approach to computational creativity: The curious case of chef watson," *IBM Journal of Research and Development*, vol. 63, no. 1, pp. 7–1, 2019.

[3] D. Burca, M. Schüller, and J. Zlabinger, "Case-based reasoning and machine learning," 2018.

[4] C. Trattner and D. Elsweiler, "Food recommender systems: important contributions, challenges and future research directions," *arXiv preprint arXiv:1711.02760*, 2017.

[5] W. Min, S. Jiang, L. Liu, Y. Rui, and R. Jain, "A survey on food computing," *ACM Computing Surveys (CSUR)*, vol. 52, no. 5, pp. 1–36, 2019.

[6] J. L. Kolodner, "An introduction to case-based reasoning," *Artificial intelligence review*, vol. 6, no. 1, pp. 3–34, 1992.

[7] R. Bergmann, K.-D. Althoff, S. Breen, M. Göker, M. Manago, R. Traphöner, and S. Wess, *Developing industrial case-based reasoning applications: The INRECA methodology*. Springer Science & Business Media, 2003.

[8] A. Cordier, V. Dufour-Lussier, J. Lieber, E. Nauer, F. Badra, J. Cojan, E. Gaillard, L. Infante-Blanco, P. Molli, A. Napoli *et al.*, "Taaable: a case-based system for personalized cooking," in *Successful Case-based Reasoning Applications-2*. Springer, 2014, pp. 121–162.

[9] F. Badra, J. Cojan, A. Cordier, J. Lieber, T. Meilender, A. Mille, P. Molli, E. Nauer, A. Napoli, H. Skaf-Molli *et al.*, "Knowledge acquisition and discovery for the textual case-based cooking system wikitaaable," in *8th International Conference on Case-Based Reasoning-ICCBR 2009, Workshop Proceedings*, 2009, pp. 249–258.

[10] P. J. Herrera, P. Iglesias, D. Romero, I. Rubio, and B. Díaz-Agudo, "Jadacook: Java application developed and cooked over ontological knowledge." in *ECCBR Workshops*, 2008, pp. 209–218.

[11] P. J. Herrera, P. Iglesias, A. M. G. Sánchez, and B. Díaz-Agudo, "Jadacook 2: Cooking over ontological knowledge," in *8th International Conference on Case-Based Reasoning (ICCBR-09)*, 2009, pp. 279–288.

[12] M. Ballesteros, R. Martın, and B. Dıaz-Agudo, "Jadaweb: A cbr system for cooking recipes," in *Workshop Proceedings*, 2010, p. 179.

[13] A. Hanft, R. Newo, K. Bach, N. Ihle, and K.-D. Althoff, "Cookiis–a successful recipe advisor and menu creator," in *Successful Case-based Reasoning Applications-I*. Springer, 2010, pp. 187–222.

[14] C. Fuchs, C. Gimmler, S. Günther, L. Holthof, and R. Bergmann, "Cooking cake," in *ICCBR*, vol. 9, 2009, pp. 259–268.

[15] G. Müller and R. Bergmann, "Cookingcake: A framework for the adaptation of cooking recipes represented as workflows." in *ICCBR (Workshops)*, 2015, pp. 221–232.

[16] P. Schumacher, M. Minor, K. Walter, and R. Bergmann, "Extraction of procedural knowledge from the web: A comparison of two workflow extraction approaches," in *Proceedings of the 21st International Conference on World Wide Web*, 2012, pp. 739–747.

[17] G. Müller and R. Bergmann, "Poql: A new query language for process-oriented case-based reasoning." in *LWA*, 2015, pp. 247–255.

[18] M. Minor, S. Montani, and J. A. Recio-Garcia, "Process-oriented case-based reasoning," *Information Systems*, vol. 40, pp. 103–105, 2014.

[19] J. DeMiguel, L. Plaza, and B. Díaz-Agudo, "Colibricook: A cbr system for ontology-based recipe retrieval and adaptation." in *ECCBR Workshops*. Citeseer, 2008, pp. 199–208.

[20] E. Frank, M. Hall, G. Holmes, R. Kirkby, B. Pfahringer, I. H. Witten, and L. Trigg, "Weka-a machine learning workbench for data mining," in *Data mining and knowledge discovery handbook*. Springer, 2009, pp. 1269–1277.

[21] M. Keppler, M. Kohlhase, N. Lauritzen, M. Schmidt, P. Schumacher, and A. Spät, "Goetheshaker-developing a rating score for automated evaluation of cocktail recipes," in *Computer Cooking Contest Workshop, Cork, Ireland*, 2014.

[22] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[24] L. Bostan, "Ingredient-driven recipe generation using neural and distributional models," Ph.D. dissertation, MA thesis. Trento, Italy: University of Trento, 2017.

[25] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *Advances in neural information processing systems*, vol. 27, 2014.

[26] D. B. T. Teisberg, "Recipe for disaster: A seq2seq model for recipe," 2018.

[27] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.

[28] S. N. R. Gona and H. Marellapudi, "Suggestion and invention of recipes using bi-directional lstms-based frameworks," *SN Applied Sciences*, vol. 3, no. 5, pp. 1–17, 2021.

[29] Z. Yu, H. Zang, and X. Wan, "Routing enforced generative model for recipe generation," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 3797–3806.

[30] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," *Advances in neural information processing systems*, vol. 30, 2017.

[31] B. P. Majumder, S. Li, J. Ni, and J. McAuley, "Generating personalized recipes from historical user preferences," *arXiv preprint arXiv:1909.00105*, 2019.

[32] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[33] B. Aljbawi, "Health-aware food planner: A personalized recipe generation approach based on gpt-2," 2020.

[34] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[35] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[36] M. Bień, M. Gilski, M. Maciejewska, and W. Taisner, "Cooking recipes generator utilizing a deep learning-based language model," Ph.D. dissertation, 02 2020.

[37] H. H. Lee, K. Shu, P. Achananuparp, P. K. Prasetyo, Y. Liu, E.-P. Lim, and L. R. Varshney, "Recipegpt: Generative pre-training based cooking recipe generation and evaluation system," in *Companion Proceedings of the Web Conference 2020*, 2020, pp. 181–184.

[38] J. Marin, A. Biswas, F. Ofli, N. Hynes, A. Salvador, Y. Aytar, I. Weber, and A. Torralba, "Recipe1m+: A dataset for learning cross-modal embeddings for cooking recipes and food images," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 1, pp. 187–203, 2019.

[39] K. Grace, M. L. Maher, D. C. Wilson, and N. A. Najjar, "Combining cbr and deep learning to generate surprising recipe designs," in *International Conference on Case-Based Reasoning*. Springer, 2016, pp. 154–169.

[40] S. Yagcioglu, A. Erdem, E. Erdem, and N. Ikizler-Cinbis, "Recipeqa: A challenge dataset for multimodal comprehension of cooking recipes," *arXiv preprint arXiv:1809.00812*, 2018.

[41] M. Bień, M. Gilski, M. Maciejewska, W. Taisner, D. Wisniewski, and A. Lawrynowicz, "Recipenlg: A cooking recipes dataset for semi-structured text generation," in *Proceedings of the 13th International Conference on Natural Language Generation*, 2020, pp. 22–28.

View publication stats