

Image-to-Recipe Translation

Deep Learning Chef: CS 7643

Jian-Yuan Lin, Hung-Hsi Lin, Shangci Wang, Takashi Kokubun

Georgia Institute of Technology

jasonlin0211@gatech.edu, hlin56@gatech.edu, swang879@gatech.edu, k0kubun@gatech.edu

Abstract

Although we see many food images on social media today, it's rare that the post also has its cooking recipe. If you want to enjoy the taste of the food, you would like to translate the food photo into a recipe. There have been some deep learning models proposed to solve this problem, but it's not been clear which method is state-of-the-art. This project aims to compare an InceptionV3-based model with a couple of topic modeling choices and a ResNet-based model, exploring the difference between their performance and architectures. We successfully reproduced both models and were able to find several implications of their difference on the performance.

1. Introduction/Background/Motivation

Nowadays, social media has become a popular way for foodies to share their passion for food, which also inspires people who like to cook. Although there are a lot of food images on the platform, oftentimes recipes are not available in the posts. This creates difficulties for others to reproduce the dish in the food images. This study is aimed to convert such images to recipes by using topic modeling in tandem with image analysis.

CNN/VGG models with the assistance of transfer learning and other architectures like ResNet and DenseNet will be investigated for performance comparison [1,2]. The goal is to clarify trade-offs between these approaches and possibly find a way to improve the performance of the state-of-the-art model. Using topic modeling to create extra data labels is one possible direction for performance improvement. Some traditional methods for topic modeling include Latent Semantic Analysis (LSA) and Latent Dirichlet Allocation (LDA). Despite their popularity, these

methods have key limitations. For example, LSA is not efficient in topic number determination and LDA lacks the capability to model relations between topics [2]. Non-negative Matrix Factorization (NMF) is a non-probabilistic method based on matrix decomposition. Previous researchers have reported several advantages of NMF over LDA in topic modelings, such as less execution time and more incoherent topics [3, 4]. Word2Vec (W2V) is a shallow neural network that builds embeddings from large text corpus efficiently and precisely [5]. It also takes the semantic closeness of words into consideration during embedding calculation.

Our work could benefit all the foodies who like to get recipe inspiration from online images. They could get a recipe in a few seconds by simply uploading an image. It would be also useful for people to quickly understand the preparation steps for the dish they see at a restaurant and make it at home.

We used a dataset called Recipe1M+ [6] that has over a million cooking recipes and 13 million food images. The dataset has many recipes collected from various popular recipe websites, which is important for improving the performance and comes with a PyTorch-based implementation that we were able to easily reproduce and compare with the other model.

2. Approach

2.1. Topic Modeling followed by CNN Model

For the first approach, the concept is based on the concept of Serifovic's work [7, 8], which is to apply the topic modeling techniques to reduce the recipe names to different categories, and simplify the task as an image classification problem, then solve it by using a classical CNN model. The first task we focus on is to choose the data source. In the reference work, the food images and recipes were scraped from a German website [1,6], and since our model is expected to be

built in English, the Recipe1M dataset [6] was downloaded and organized into frames that we could easily load. The second step is to reduce the dimensionality for the image titles to avoid the overfitting issue, which is the topic modeling. We have replicated the NMF as the benchmark results, and we also developed a new topic modeling technique that combines W2V with K-means clustering (W2V-Kmeans). The influences of these two different methods on prediction accuracy are also investigated and compared. Finally, we need to train the CNN model and combine it with the features extracted from the VGG 16 model to make the prediction of the recipe categories, and then output the possible recipes. We choose to fine-tune the existing pre-trained model instead of building a model from scratch, and the Inception v3 model structure is applied [9]. After we get the predicted probabilities from the CNN model, we combine them with the features extracted from the VGG 16 model to produce recipe prediction results.

2.1.1. Topic modeling

A set of common text pre-processing methods is applied to recipe titles, including text splitting, lowercase conversion, stop-word filtering, etc. The collection of recipe titles are firstly converted and normalized to a matrix of TF-IDF features. Models with topic numbers of 30 and 300 are developed respectively. Topic terms for each category are extracted based on the weights in the topic-term and term-recipe matrices.

The same pre-processed recipe title corpus is used in W2V-Kmeans. Recipe titles are firstly transformed to W2V embeddings by using the CBOW algorithm, which is then clustered by using the k-means [10,11] with cluster numbers of 30 and 300 as well. The cluster (topic) group is predicted for each recipe and the two most frequent terms in each cluster are considered as the corresponding topic name.

t-SNE is applied to visualize relationships among the recipe groups predicted by NMF and W2V-K-means method, respectively. To reduce the calculation complexity, t-SNE is only used for 30 topic groups with the first 1000 recipes extracted from each group. The first two latent t-SNE dimensions are visualized. Due to space limitations, figures with the better resolution are in Appendix.

2.1.2. CNN model

After the corresponding topic clusters (labels) are defined, then the remaining task can be viewed as an image classification problem, namely given an input image, finding the corresponding topic cluster, and then output the recipe. We use a pre-trained CNN model and perform the fine-tuning technique to avoid the model design iteration. In the benchmark work [8], the Inception v3 model showed better accuracy than VGG16, therefore the Inception v3 model is selected in our approach. The first problem we anticipate is the expensive training time and computation resources due to the high complexity of the Inception v3 model. Google Colab PRO is used for prototyping and training with only a small portion of data. Then we utilize a local machine with NVIDIA RTX 3060 to perform the full scale of training. Another problem is the scale of the input data is more than 800k images, and loading the data easily reaches the limit of memory. Therefore, a self-defined dataset combined with PyTorch data loader is implemented to load the data parallelly without consuming the RAM. The pre-trained Inception v3 model without the last fully connected layer is imported, then training/validation images and labels are loaded by the data loader to perform the CNN model training. Lastly, the feature extractor is created by utilizing the last layer of VGG16 and combined with the probability predicted by the CNN model, the predicted recipe category is then predicted. Figure 1 shows the concept of the entire methodology.

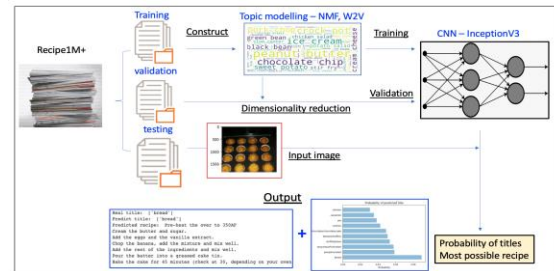


Figure 1. Concept of implementing topic modeling and CNN model for the image to recipe prediction.

2.2. Im2Recipe

Amalia Salvador et al. proposed a model called im2recipe [12]. This model has an architecture shown in Figure 2. The model uses LSTM to encode recipe instructions and ingredients. It uses pre-trained ResNet50 to encode recipe images. Then the whole model is trained to reduce cosine similarity between the recipe embeddings and the image embeddings for

correct pairs of a recipe and an image whereas one for random pairs of them is increased.

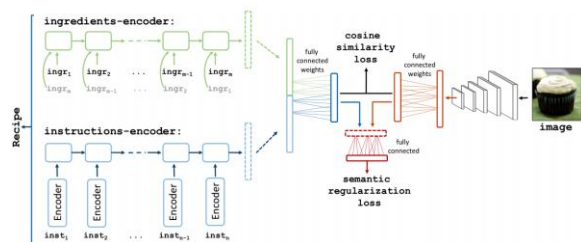


Figure 2. Architecture of im2recipe

We used their im2recipe-Pytorch repository [13] as a baseline. Because the original code has several bugs crashing the training process, such as pointing to wrong paths and insufficient dependency specifications, we fixed them and some are publicly submitted as pull requests to the repository.

Another problem is that their document was not comprehensive and it wasn’t clear what is required to reproduce the whole process. We investigated the implementation and created a document that describes which script requires what files and in what order you need to run them. This would be useful for anybody who will try to reproduce it in the future.

Our other contribution is, while the original repository only has scripts to measure the performance in losses, we created a script to translate a query image into recipes using embeddings generated by the model. This makes it easier to use the model for real-world applications, making the project more practical.

3. Experiments and Results

3.1. Topic modeling

Results are compared between NMF and W2V-K-Means methods with 30 topic groups.

By using NMF, 6.3% of the recipes in the training data are not assigned to any topic group due to their dissimilarity to the rest. The 5 most frequent topic categories in NMF are “salad” (6.6%), “stuffed, shrimp” (6.0%), “chicken” (5.4%), “sauce”(4.3%), and “grilled, chicken” (4.3%). Figure 3 presents the relationships among the 30 topic groups visualized by t-SNE. Interestingly, NMF achieved a very clear separation among the 30 topic groups (Figure 3a). However, recipes with similar topic concepts (e.g. “chicken” vs. “grilled chicken”; “shrimp” vs. “stuffed

shrimp”) are generally located on the opposite side. In addition, the low similarity is noticed for groups that are next to each other in Figure 3a, such as “cake” and “salad”.

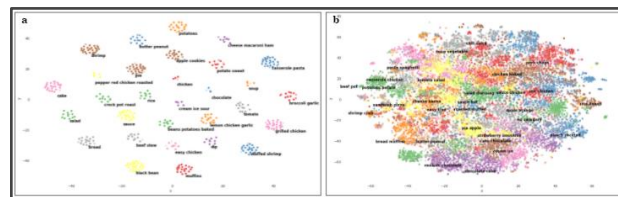


Figure 3. Selected recipes (1k) clustered based on topic groups (n=30) predicted by NMF (a/L) or V2W-Kmeans (b/R) for recipe titles. Refer to Appendix for the better resolution version.

As for V2W-Kmeans, only 3.1% of recipes are in the “no category” group, which is 50% less than that in NMF. “Salad, dressing” (5.4%), “sauce, chicken” (4.8%), “chicken, baked”(4.3%), “cheese bacon” (4.1%), and “roasted, stuffed” (4.0%) are the top 5 recipe categories in V2W-Kmeans. Relationships among topic groups are present in Figure 3b. Dessert-type topics are all clustered at the bottom, against the savory-type topics on top. Cocktail recipes form a unique group that is close to sweet food groups. This makes sense since many cocktail recipes use sweet ingredients such as sugar, fruit, juice, etc. In addition, we also notice that most recipes with “chicken” are next to each other in Figure 3b, such as “sauce, chicken”, “chicken, baked”, and “stir, chicken”. Moreover, these chicken theme groups are right next to “pock, chop”.

Our results suggest that compared to NMF, the V2W-K-means method shows better performance in representing the semantic correlations among the training recipes in this study, which agrees with findings from previous researchers [5, 13, 14].

3.2. Training the CNN Model

Because the full Recipe1M+ dataset is too large and it didn’t fit into our storage, we used their smaller old version of the dataset called Recipe1M. Given that each epoch processes all recipes of Recipe1M in the training set and takes some time, we trained our models with just 30 epochs to leave some space for comparing different hyperparameters.

Using labels generated by topic modeling, we fine-tuned a pre-trained InceptionV3 to predict labels of

categories we prepared. We measured the performance with an accuracy of the label prediction as the original code did.

Our first experiment is that we tried a couple of learning rates for training the model. Figures 1 and 2 show accuracy curves for each learning rate. Loss curves are Figures D1 and D2 in the appendices.

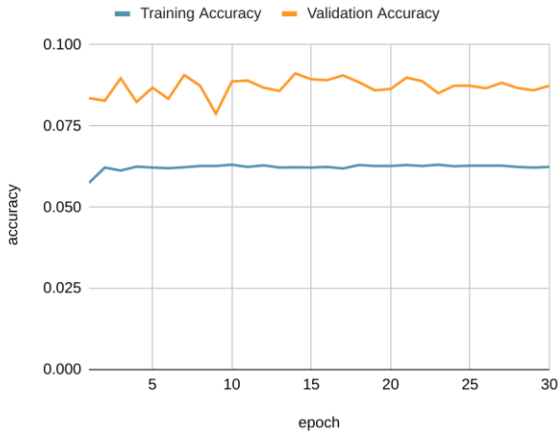


Figure 1: Accuracy curves of the CNN trained with NMF 300 categories and a learning rate = 0.01

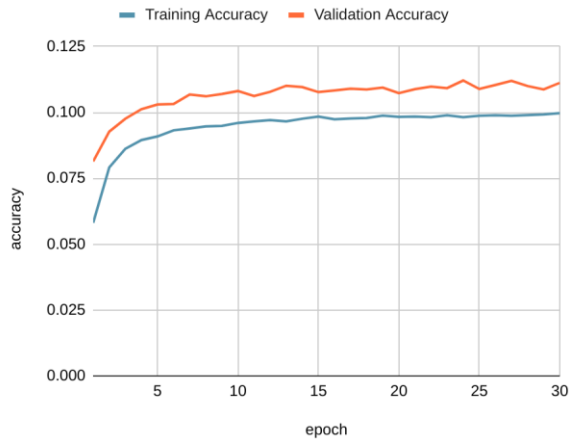


Figure 2: Accuracy curves of the CNN trained with NMF 300 categories and a learning rate = 0.001

0.001 gives relatively steady progress whereas we can't see any improvement with 0.01. In assignments of the class, relatively large learning rates like 0.01 gave better results. However, the difference from it here could be that Recipe1M has a lot of data for training. Even if a small learning rate is used, once many batches are processed, the cumulative changes can be bigger than a model trained with a large learning rate with a small dataset [15].

Then we compared a couple of topic modelings we implemented: NMF and W2V-Kmeans. In terms of the number of categories to be generated by the topic modelings, we used 300 categories as a starting point. Figures 2 and 3 show accuracy curves for each topic modeling with the 0.001 learning rate.

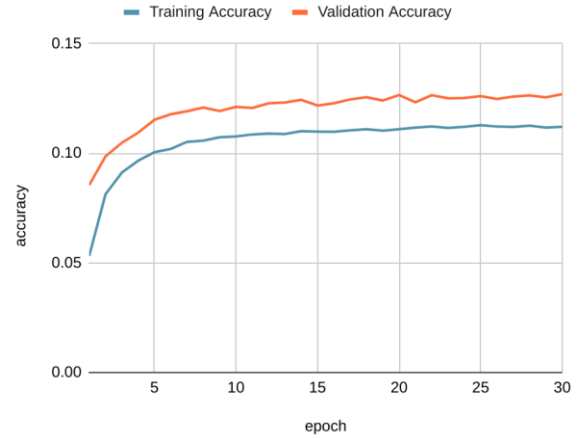


Figure 3: Accuracy curves of the CNN trained with W2V 300 categories and a learning rate = 0.001

Both of these figures have a similar curve presumably thanks to the 0.001 learning rate. But W2V-Kmeans almost always outperforms NMF. We believe that this is because Word2Vec accounts for the semantic closeness of words whereas words are just independent values of a matrix in NMF.

Because the accuracy doesn't look that good with 300 categories, we tried some other numbers of categories and found 30 categories give much better accuracy. Figures 3 and 4 show accuracy curves for 300 categories and 30 categories generated by W2V with the 0.001 learning rate.

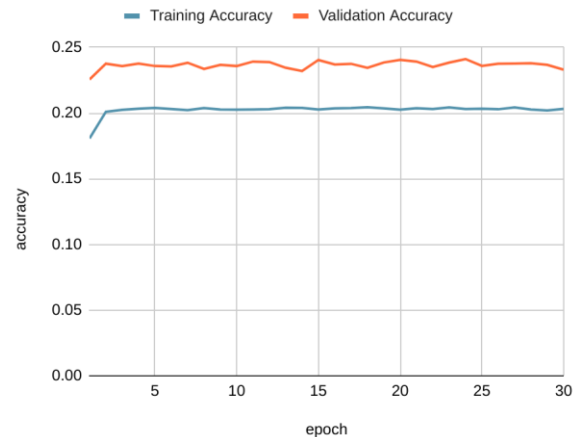


Figure 4: Accuracy curves of the CNN trained with W2V 30 categories and a learning rate = 0.01

The accuracy nearly doubled by reducing the number of categories. However, improving the accuracy alone doesn't mean anything because the model's output is also impacted by the other part of the model that considers the similarity of images when the category is the same. We'll visit how this impacts the final performance at 3.4.

3.3. Training the Im2Recipe Model

We trained the im2recipe model with the default parameters, such as a 0.0001 learning rate, for 30 epochs to compare it with the CNN model. Then we evaluated the performance of this model with models trained by the original authors with different settings, such as larger epochs and/or using full Recipe1M+. The im2recipe outputs embeddings for images and recipe texts respectively, and their cosine loss gives a loss of associating correct pairs of recipes and images. Figure 5 shows the cosine loss curve. The loss seems a bit unstable, but the trend seems to show a constant decrease. We think this instability is made because the trainer gives -1 as a target value for random pairs and it could possibly pair a recipe and an image that are actually very similar and ideally should give 1 as the target value. The fact that the correctness of the training relies on randomly generated pairs could negatively impact the training speed. We take a look at the performance comparison with the other model using the same epochs at 3.4.

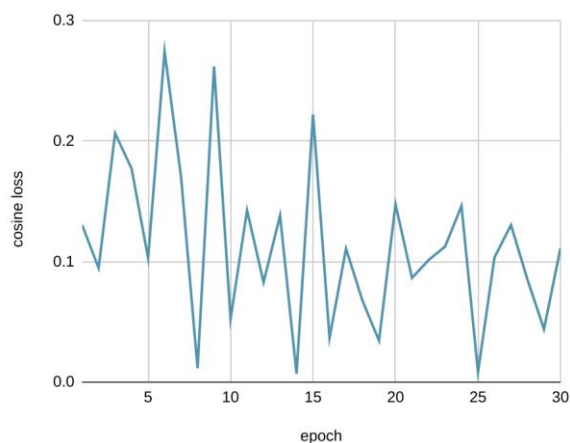


Figure 5: Cosine loss curve of the im2recipe model

For validation, the authors used an interesting metric called MedR (median rank) score. To calculate a MedR score, you first randomly sample 1000 recipes. Then for each recipe in it, you create a ranking of the similarity between the image of the selected recipe and recipe texts of the 1000 recipes, which can be calculated by matrix multiplication of these embeddings. It looks for only the rank of the recipe text embedding that was originally paired with the selected recipe, and the median of 1000 ranks becomes MedR. The original recipe text should give the best similarity for the query image, so MedR becomes 1 if the model performance is perfect, and 1000 if otherwise. Figure 6 shows the MedR curve of our training.

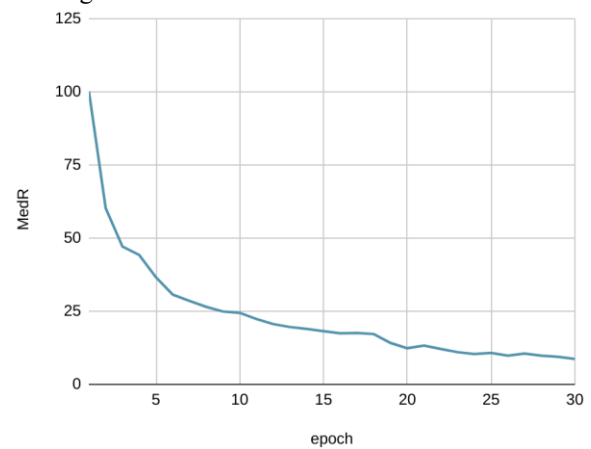


Figure 6: MedR curve of the im2recipe model

The MedR curve is stabler than the cosine loss curve. This seems to be because the MedR calculation accounts for only the top actual pair and says nothing about other recipes, which does not rely on randomly generated labels like the loss. However, the problem with this metric is that it's non-differentiable; if we were able to use this as the loss function and back-propagate the gradient, it would have a faster learning speed.

3.4. Comparing the CNN Model and Im2Recipe

Because losses of individual models cannot be used for evaluating both models, comparing the performance of different models is a challenge. We originally planned to use the above MedR score to compare those two models. However, unlike the im2recipe model, the CNN model predicts recipe images using a query recipe image. Since the original

recipe has the same features and it gives the best similarity and the same category prediction, MedR always becomes 1 by its nature. So it's not fair to use MedR for comparison.

We propose a method that can compare any model solving the image-to-recipe translation problem. That is to let humans choose the best prediction selected from a test dataset, using a set of query images that are in neither the training dataset nor the test dataset. Model details are blinded by 1-digit code (e.g. model 1,2,3) to human testers to avoid expectation bias. Then we count the number of times each model becomes the best and uses that as a score of the model. To ensure the quality of the evaluation, we all did the evaluation. If more reliable results are needed, the evaluation can be delegated to a service called Amazon Mechanical Turk [14] to crowdsource the evaluation. Figures E1 and E2 in the appendices show the test data that we used for the evaluation.

While the primary purpose of the evaluation using the framework was to compare im2recipe and the CNN and therefore we included these, we also wanted to rely on this framework to compare the overall performance of the CNN where CNN's accuracy is improved by reducing the number of categories but doesn't necessarily mean it improves the final prediction. Therefore we tested im2recipe, the CNN trained by 300 categories of W2V-Kmeans, and the CNN trained by 30 categories of W2V-Kmeans. We did not include any other combination because then we would need to test more recipes to get the same granularity of the scores and it's time-consuming for humans and potentially decreases the evaluation quality when a human gets bored to complete it. Table 1 shows the scores of the three models.

Model	Score
im2recipe	9.25
CNN W2V 300 categories	68
CNN W2V 30 categories	44.5

Table 1: The performance of the models trained with 30 epochs evaluated by ourselves

The higher the score is, the better the model performs. The CNNs significantly outperform im2recipe. The training speed issue of im2recipe discussed in 3.3 does seem to impact the comparison with the same number of epochs here. However, we observe that when im2recipe gives a better result than the CNNs, it seemed to recognize correct ingredients whereas the CNNs fail to do so. That's presumably because im2recipe can recognize any number of recipe categories while the CNN models cannot distinguish categories that do not fit in the fixed number of categories. It also explains the effectiveness of the CNN with 300 categories where we pick the most frequent 300 categories and thus it can cover a large number of popular recipes.

4. Experience

4.1. Challenges and Changes in Approach

In this section, several challenges and changes in approaches are summarized.

4.1.1. Expensive Training Time

Computation resources and training time are both expensive due to large data size and it results in running out of RAM and crashes easily. We first loaded the datasets in a list for small-scale prototyping, which works fine. However, we found out that it easily reaches the limitation of RAM as the data scale becomes larger. Besides, even if the training can be conducted, it will be extremely slow without optimization and parallelization. Therefore, we found out that with the data loader module, the data loading can be parallelized (up to 4 workers) and the training speed increased significantly. Meanwhile, since we use a data loader to load the data on the fly, we don't need to save anything before the training starts, and we never encounter the crash of the runtime caused by RAM.

4.1.2. Semantic regularization of im2recipe

Related to the above challenge of expensive training time, im2recipe takes much more training time than Murgio's CNN, which made it impossible to explore hyperparameters of im2recipe with our schedule by the deadline.

One of the hyperparameters of im2recipe we were interested in trying is a flag to disable semantic regularization proposed by Amaia et al. [12]. It is to

account for semantic categories, which is somewhat similar to what Murgio’s model does. It is enabled by default and therefore we chose a better version, but it would be interesting to train the model with semantic regularization disabled under our settings and discuss its performance implications.

4.1.3. Missing and Imbalanced Data

Some of the recipes have either no corresponding images or multiple images, which makes the original dataset imbalanced. To solve this issue, we filtered out those recipes which have no corresponding images and conducted the training on the rest of the dataset, and it worked better than just importing originally non-organized data. From our studies, some preprocess including data cleaning, data augmentation like applying the focal loss to adjust the imbalance of the dataset could also be a potential way for training accuracy improvement.

4.1.4. Mislabeled Data

We found out that some images in Recipe1M were not relevant to the recipe or food content. For example, an image of a woman is observed in a food category. We believe that there might be more images exhibiting wrong content or have little connections to their categories, which mislead our model and potentially downgrade the accuracy. In addition, a certain amount of data have recipe titles irrelevant to food image or recipe content, especially those in the “no category” topic group, such as “Unbelievable Stain Remover”, “Lemon Charge”, “The Big Guido” etc. Recipes with non-English words as titles create another layer of challenge and these recipes generally fall into the “no category” group. Overall, these noisy labels could significantly affect our model performance.

4.1.5. Prediction on Recipe with Known Topic

The challenge comes from how to output the most related recipe in the case that we already successfully predict the topic of this image. For NMF topic modeling, since each object is “hardly” classified into one cluster, it is not possible to find which recipes in the cluster match better. One way we did is to find the recipe which appears most frequently, but it is not a reliable method. Therefore, we implement W2V for topic modeling, since we could also have the information of “distance” between the input image with all other objects located in the same cluster. Thus,

we can find which recipe in the training set has the best similarity with the input testing image.

4.2. Project Success

We had originally planned to measure success through metrics called MedR that we explained in 3.3. However, we later noticed it doesn’t make sense for Murgio’s CNN because it always gives the best performance as we discussed in 3.3. We then came up with an idea to evaluate models by humans and found our proposed approach performs better than the competitors. The result also verified what we originally guessed, which is that low accuracy in category prediction with many categories doesn’t necessarily mean overall bad performance.

5. Conclusion

We proposed a topic modeling approach called W2V-Kmeans that lets Murgio’s CNN [8] outperform the CNN trained with NMF topic modeling they originally applied. We presented a method to evaluate models that use different loss functions by humans, and it showed our CNN trained with Recipe1M+ [6] and W2V-Kmeans outperforms the im2recipe model [7] that is implemented by the authors of Recipe1M+. However, it remains to be a challenge that we don’t have a formal, automated way to evaluate the whole pipeline of Murgio’s CNN.

6. Work Division

Summary of contributions is provided in Table x.

7. References

- [1] German food recipe website: CHEFKOCH <https://www.chefkoch.de/>
- [2] Rubayyi Alghamid and Khakid Alfalqi. A Survey of Topic Modeling in Text Mining. *International Journal of Advanced Computer Science and Applications*, Vol. 6, No. 1, 2015.
- [3] Derek O’Callaghan, Derek Greene and Joe Carthy, Pádraig Cunningham. An analysis of the coherence of descriptors in topic modeling. *Expert Systems with Applications*. Vol.42 5645-5657, 2015.
- [4] Evan Sandhaus. The New York Times Annotated Corpus. *Linguistic Data Consortium*. Oct. 2008

[5] Magnus Sahlgren. Rethinking Topic Modelling: From Document-Space to Term-Space. *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2250–2259 November 16 - 20, 2020.

[6] Marin, Javier and Biswas, Aritro and Ofli, Ferda and Hynes, Nicholas and Salvador, Amaia and Aytar, Yusuf and Weber, Ingmar and Torralba, Antonio. Recipe1M+: A Dataset for Learning Cross-Modal Embeddings for Cooking Recipes and Food Images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2019.

[7] Muriz Serifovic: Image-to-Recipe Translation with Deep Convolutional Neural Networks <https://towardsdatascience.com/this-ai-is-hungry-b2a8655528be>

[8] Murgio's GitHub: Food-Recipe-CNN <https://github.com/Murgio/Food-Recipe-CNN>

[9] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathan Shlens and Zbigniew Wojna. Rethinking the Inception Architecture for Computer Vision. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[10] VukGlisovic. Classification combining LDA and Word2Vec. 2017.

6<https://www.kaggle.com/vukglisovic/classification-combining-lda-and-word2vec>

[11] Saket Garodia. Topic Modelling using Word Embeddings and Latent Dirichlet Allocation. 2020. <https://medium.com/analytics-vidhya/topic-modelling-using-word-embeddings-and-latent-dirichlet-allocation-3494778307bc>

[12] Salvador, Amaia and Hynes, Nicholas and Aytar, Yusuf and Marin, Javier and Ofli, Ferda and Weber, Ingmar and Torralba, Antonio. Learning Cross-modal Embeddings for Cooking Recipes and Food Images. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[13] Javier Martin Tur et al. im2recipe: Learning Cross-modal Embeddings for Cooking Recipes and Food Images. <https://github.com/torralba-lab/im2recipe-Pytorch>

[14] Romain Gauchon, Jean-Pascal Hermet. Individuals tell a fascinating story: using unsupervised text mining methods to cluster policyholders based on their medical history. 2019. final-02356449f

[15] D. Randall Wilson and Tony R. Martinez. The Need for Small Learning Rates on Large Problems. *Proceedings of the 2001 International Joint*

Conference on Neural Networks (IJCNN'01), 115-119. 2001.

[16] Mohsen Asghari, Daniel Sierra-Sosa and Adel Elmaghraby. Trends on Health in Social Media: Analysis using Twitter Topic Modeling. *2018 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*. 6-8 Dec. 2018

[17] Amazon Mechanical Turk website:

<https://www.mturk.com/>

[a-1]

VukGlisovic. Classification combining LDA and Word2Vec. 2017.

<https://www.kaggle.com/vukglisovic/classification-combining-lda-and-word2vec>

[a-2]

Saket Garodia. Topic Modelling using Word Embeddings and Latent Dirichlet Allocation. 2020.

<https://medium.com/analytics-vidhya/topic-modelling-using-word-embeddings-and-latent-dirichlet-allocation-3494778307bc>

[15] D. Randall Wilson and Tony R. Martinez. The Need for Small Learning Rates on Large Problems. *Proceedings of the 2001 International Joint Conference on Neural Networks (IJCNN'01)*, 115-119. 2001.

A. Project code Repository

The GitHub repository for our final project is at:

https://github.gatech.edu/jlin323/DL_GroupProject_2021.

Please contact jasonlin0211@gatech.edu if your account has not been invited.

B. Work Division

Name	Duty
Jian-Yuan Lin	<ul style="list-style-type: none"> - Prepare the Recipe 1M dataset - Integrate the topic modeling with CNN training - Conduct the training for the CNN prototype model (Small scale)
Hung-Hsi	- Preprocess and organize the

Lin	training/validation/test data. - Help developing topic modeling methods - Integrate the topic modeling with CNN training
Shangci Wang	- Develop the topic modeling techniques(NMF, W2V-Kmeans) - Visualization of the corresponding categories topic
Takashi Kokubun	- Im2Recipe approach evaluation, bug fixes, training, and testing - Extract features by VGG16 and integrate all of the team member's code for the CNN - Run the full-scale dataset training on a local machine - Provide the test framework for team members to judge which model is better

Table x: Contributions of team members.

C. Topic Modeling Figures

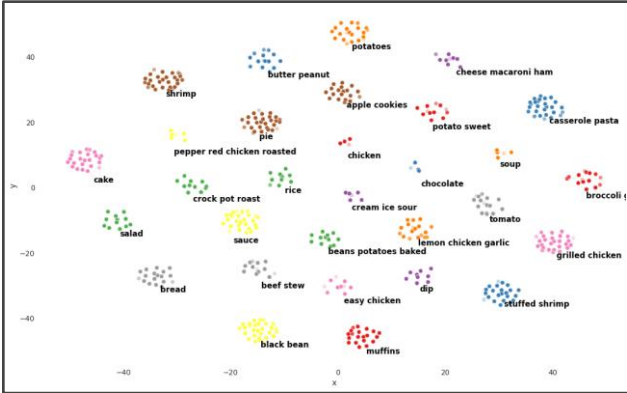


Figure C1. Selected recipes (1k) clustered based on topic groups (n=30) predicted by NMF for recipe titles. Refer to Appendix for the better resolution version.

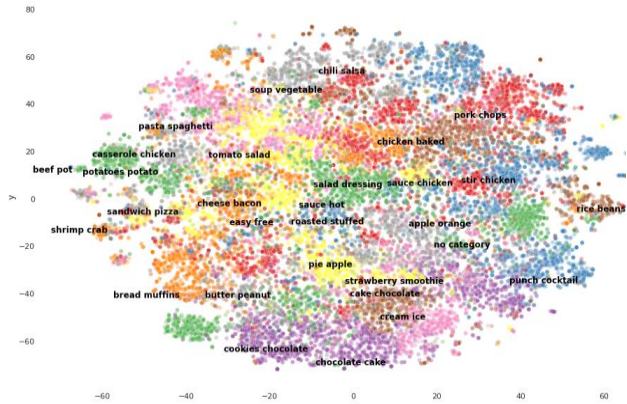


Figure C2. Selected recipes (1k) clustered based on topic groups (n=30) predicted by V2V-Kmeans for recipe titles. Refer to Appendix for the better resolution version.

D. Loss Curves for the training of the CNN with NMF

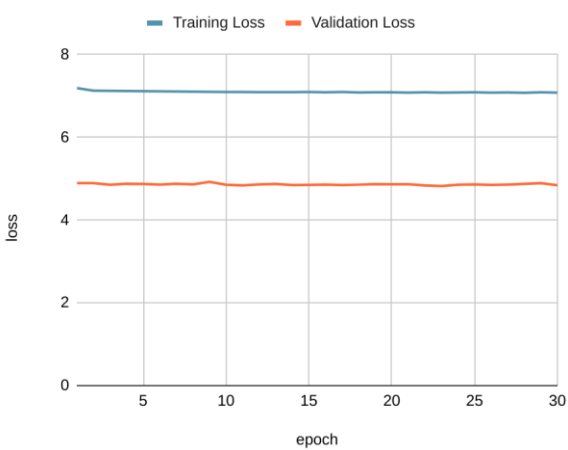


Figure D1: Loss curves of the CNN trained with NMF 300 categories and a learning rate = 0.01

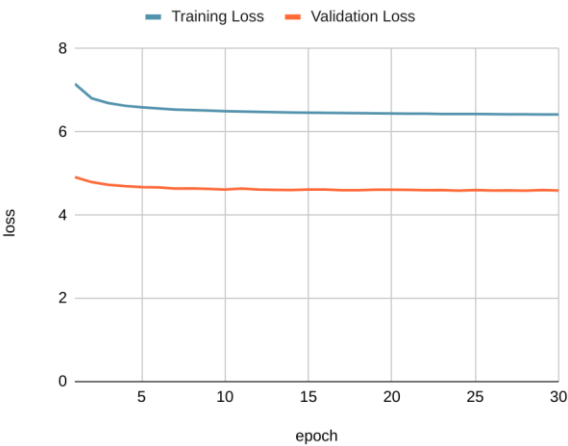


Figure D2: Loss curves of the CNN trained with NMF 300 categories and a learning rate = 0.001

E. The test framework used for the final evaluation



idx	query	model1	model2	model3
1		Red Cabbage, Blue Cheese, and Walnut Empanadas	Beer Cheese Soup IV	Beer Cheese Soup IV
2		Granola	Creamy Noodles	Penne in Parmesan Cream Sauce

Figure E1: The test cases presented to humans.
Results are linked to online recipes.

idx	model1	model2	model3
1		1	1
2		1	
3		1	
4	1		
5			1

Figure E2: A spreadsheet to fill out an evaluation by a human. Multiple models may get 1 when they give the same prediction.