Supplementary Material for "MCEN: Bridging Cross-Modal Gap between Cooking Recipes and Dish Images with Latent Variable Model"

A. Training Details

The image encoder is initialized with the pretrained ResNet-50 of PyTorch implementation [1]. During training, the images are resized to 256 pixels in their shortest side and random crops of 224 $\times$ 224 are taken. During inference, we simply use the central 224 $\times$ 224 pixels. We use dropout on top of the encoder output (Equation 1) and the final output layer (Equation 8) with dropout rate 0.2.

On the recipe side, we keep a maximum of 20 ingredients and 25 instruction sentences per recipe. Moreover, we truncate each sentence to a maximum of 30 words. The implementation of attention-based RNN decoder follows the classic implementation Dl4mt [2], where the previous hidden state is fed to a GRU before being fed to the decoder. Dropout is also applied with the similar strategy as the image encoder.

During training, we first freeze the parameters of ResNet and only optimize the rest parameters for 30 epochs. Then we unfreeze the ResNet weights and fine-tune the entire model for another 100 epochs. Early stopping strategy is utilized to select the best model with R@1 on validation for both periods.

B. Scalability and Stability

We compare the MedR score of MCEN and other baselines against subsets larger than 10K to investigate the scalability of our model. As shown in Figure 1, it can be observed that MCEN outperforms all baselines on all test sets. Moreover, as the size of test set increases, the performance gap between MCEN and baselines becomes larger, indicating the robustness of MCEN. Especially on the largest 50K set, MCEN achieves a significant improvement, ranking the true positive by 15.3 positions ahead compared with ACME.

Furthermore, we also list the results of MCEN and some baselines on different sampled subsets. We can observe that the R@1 scores of MCEN on different test sets are more stable than those of ACME and Adamine. This comparison proves that the proposed model is more robust than the baseline and is capable to adapt to various settings.

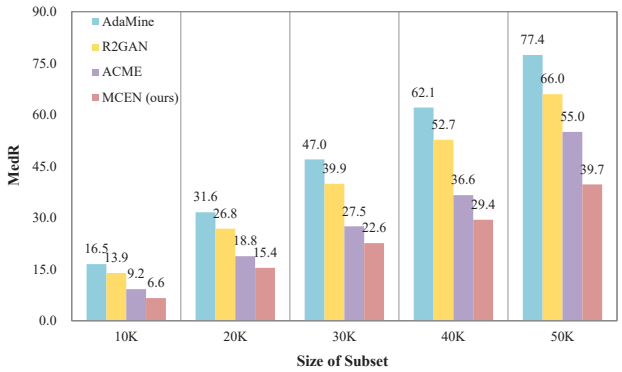

Figure 1. Scalability comparison between different models for image-to-recipe retrieval. Median Rank (MedR, lower is better) is used as the evaluation metric.

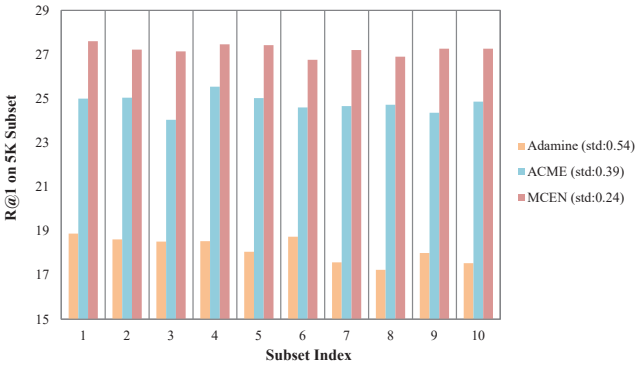

Figure 2. Stability comparison between ACME and MCEN (ours) on im2recipe task. Recall at 1 (R@1, higher is better) is used as the evaluation metric.

C. Analysis of Learned Embedding

We depict the learned image embeddings obtained by MCEN using a *t*-SNE visualization in Figure 3. The items are sampled from 5 of the most frequent classes and the color of each point indicates the category it belongs to. It can be observed that the embeddings with the same label are quite close while items in different classes are relatively

[1] https://download.pytorch.org/models/resnet50-19c8e357.pth

[2] (https://github.com/nyu-dl/dl4mt-tutorial)

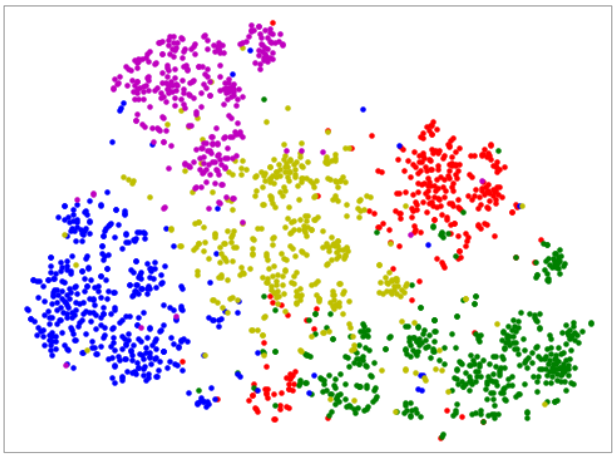

Figure 3. *t*-SNE visualization of learned embeddings.

Figure 4. Attention visualization of polysemous cases, where multiple images (on the same line) correspond to one recipe.

far away from each other.

An interesting observation of Recipe1M dataset is the high variations in images where multiple images may correspond with single recipe since dishes cooked by diverse users can be very different. However, current methods do not explicitly address this problem. As shown in Figure 4, the proposed cross-modal attention approach is capable to force the model to focus more on common and important regions shared by various images. With this technique, images corresponding to the same recipe can be mapped to similar embeddings.

D. Case Study

We visualize some retrieval results of MCEN on im2recipe and recipe2im tasks in Figure 5 and Figure 6 respectively. For image-to-recipe, we can find that the retrieved recipe has several common ingredients with the true one. For recipe-to-image task, it can be observed that the retrieved images are visually similar to the ground truth one.

True Ingredients: pasta, tomatoes, olive oil, parmesan cheese, parsley, garlic, basil, salt

Retrieved Ingredients: chicken breasts, olive oil, lemon juice, pepper, salt, parmesan cheese, pasta, onion, tomatoes, basil, parsley, mozarella cheese

True Insturctions: Cook pasta according to package directions. In a bowl , combine remaining ingredients. Rinse and drain pasta and add to tomato …

Retrieved Insturctions: Combine all seasoning for chicken except mayonaise and lemon juice on a plate. Brush chicken lightly with mayonnaise / lemon …

True Ingredients: lettuce, romaine, garlic, ginger, pepper, sesame oil, soy sauce, water, sesame seeds

Retrieved Ingredients: lettuces, oakleaf, romaine, olive oil, garlic, salt, pepper, lemon, water, lemon juicey

True Insturctions: Cut romaine crosswise into 2-inch pieces and put in a bowl. Cook garlic, ginger, and red-pepper flakes in sesame oil in a small skillet …

Retrieved Insturctions: Put 4 salad plates in the freezer to chill. Assemble all the ingredients as well as a garlic press, a big salad bowl, and a strainer …

Figure 5. Sampled results of image-to-recipe retrieval on 50K test set.

Ingredients: 2 medium acorn squash, 4 tbsp. olive oil, kosher salt, pepper, 1 c apple cider, 1 tbsp red wine vinegar, 1 tbsp . whole - grain mustard…
Insturctions: Boil the potatoes in salted water for about 12 minutes, drain and keep aside. Heat the ghee in a wok over a medium flame. When hot add the mustard ...

Ingredients: 1 Lb turkey sausage, the breakfast kind 12 large eggs, 12 teaspoon salt, 12 teaspoon black pepper, 1 cup cheddar cheese , shredded, 1 cup salsa …
Insturctions: In a large non-stick pan, crumble and cook turkey until no longer pink, about 8 minutes. Remove from pan. Wipe pan clean with paper towel and ...

Figure 6. Sampled results of recipe-to-image retrieval on 50K test set. The top-5 retrieved images are shown from left to right, and the ground truth image is boxed in red.