

Self-Attention and Ingredient-Attention Based Model for Recipe Retrieval from Image Queries

Matthias Fontanellaz

`matthias.fontanellaz@students.unibe.ch`
ARTORG Center, University of Bern
Bern, Switzerland

Stergios Christodoulidis

`stergios.christodoulidis@artorg.unibe.ch`
ARTORG Center, University of Bern
Bern, Switzerland

Stavroula Mougiakakou

`stavroula.mougiakakou@artorg.unibe.ch`
ARTORG Center, University of Bern
Bern, Switzerland

ABSTRACT

Direct computer vision based-nutrient content estimation is a demanding task, due to deformation and occlusions of ingredients, as well as high intra-class and low inter-class variability between meal classes. In order to tackle these issues, we propose a system for recipe retrieval from images. The recipe information can subsequently be used to estimate the nutrient content of the meal. In this study, we utilize the multi-modal Recipe1M dataset, which contains over 1 million recipes accompanied by over 13 million images. The proposed model can operate as a first step in an automatic pipeline for the estimation of nutrition content by supporting hints related to ingredient and instruction. Through self-attention, our model can directly process raw recipe text, making the upstream instruction sentence embedding process redundant and thus reducing training time, while providing desirable retrieval results. Furthermore, we propose the use of an ingredient attention mechanism, in order to gain insight into which instructions, parts of instructions or single instruction words are of importance for processing a single ingredient within a certain recipe. Attention-based recipe text encoding contributes to solving the issue of high intra-class/low inter-class variability by focusing on preparation steps specific to the meal. The experimental results demonstrate the potential of such a system for recipe retrieval from images. A comparison with respect to two baseline methods is also presented.

CCS CONCEPTS

- Information systems → Specialized information retrieval;
- Computing methodologies → Artificial intelligence; Natural language processing; *Image representations*.

KEYWORDS

Neural Networks, Deep Learning, Cross-modal Retrieval, Natural Language Processing, Self-attention

ACM Reference Format:

Matthias Fontanellaz, Stergios Christodoulidis, and Stavroula Mougiakakou. 2019. Self-Attention and Ingredient-Attention Based Model for Recipe Retrieval from Image Queries. In *5th International Workshop on Multimedia*

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

MADiMa '19, October 21, 2019, Nice, France

© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6916-9/19/10...\$15.00
<https://doi.org/10.1145/3347448.3357163>

Assisted Dietary Management (MADiMa '19), October 21, 2019, Nice, France.
ACM, New York, NY, USA, 7 pages. <https://doi.org/10.1145/3347448.3357163>

1 INTRODUCTION

Social media and designated online cooking platforms have made it possible for large populations to share food culture (diet, recipes) by providing a vast amount of food-related data. Despite the interest in food culture, global eating behavior still contributes heavily to diet-related diseases and deaths, according to *the Lancet* [8]. Nutrition assessment is a demanding, time-consuming and expensive task. Moreover, the conventional approaches for nutrition assessment are cumbersome and prone to errors. A tool that enables users to easily and accurately estimate the nutrition content of a meal, while at the same time minimize the need for tedious work is of great importance for a number of different population groups. Such a tool can be utilized for promoting a healthy lifestyle, as well as to support patients suffering food-related diseases such as diabetes. To this end, a number of computer vision approaches have been developed, in order to extract nutrient information from meal images by using machine learning. Typically, such systems detect the different food items in a picture [1], [27], [21], estimate their volumes [16], [6], [10] and calculate the nutrient content using a food composition database [20]. In some cases however, inferring the nutrient content of a meal from an image can be really challenging - due to unseen ingredients (e.g. sugar, oil) or the structure of the meal (mixed food, soups, etc.).

Humans often use information from diverse sensory modalities (visual, auditory, haptic) to infer logical conclusions. This kind of multi-sensory integration helps us process complex tasks [12]. In this study, we investigate the use of recipe information, in order to better estimate nutrient content of complex meal compositions. With the aim to develop a pipeline for holistic dietary assessment, we present and evaluate a method based on machine learning to retrieve recipe information from images, as a first step towards more accurate nutrient estimation. Such recipe information can then be utilized together with the volume of the food item to enhance an automatic system to estimate the nutrient content of complex meals, such as lasagna, crock pot or stew.

The performance of approaches based on machine learning relies heavily on the quantity and quality of the available data. To this end, a number of efforts have been made to compile informative datasets to be used for machine learning approaches. Most of the early released food databases were assembled only by image data for a special kind of meal. In particular, the first publicly available database was the Pittsburgh Fast-Food Image Dataset (PFID) [5], which contains only fast food images taken under laboratory conditions. After the recent breakthrough in deep learning models,

Meal Image	Ingredient List	Instruction List
	<ul style="list-style-type: none"> 1. chicken 2. garlic 3. butter 4. lemon 5. rosemary 6. basil 7. thyme 8. salt and pepper. 	<ol style="list-style-type: none"> 1. Preheat oven to 370F. 2. Rub salt and pepper to the chicken and set aside for around 10 mins. 3. rub half of the butter to the chicken. 4. Mince 1 whole garlic and rub on the chicken. 5. Rub chicken with Herbs. 6. Stuff chicken with lemon, butter, 1 whole garlic, salt, pepper and herbs. 7. Put chicken in oven covered with foil for 45 mins. 8. Remove cover and cook for another 45 mins at 400F
	<ul style="list-style-type: none"> 1. ladyfingers 2. cream cheese 3. lemon juice 4. lemon jelly powder 5. boiling water 6. ice cubes 7. cool whip topping 8. fresh raspberries 9. jelly powder 	<ol style="list-style-type: none"> 1. Grease 9 inch (23 cm) springform pan. 2. Place lady fingers around inside rim and set aside 3. Beat cream cheese in large bowl of electric mixer. 4. Add lemon juice and rind, beating on low speed until blended. 5. Dissolve lemon jelly powder in boiling water. 6. Add ice cubes, stirring until slightly thickened. 7. Add lemon jelly slowly to cream cheese mixture while beating. 8. Increase speed and beat just until well blended. 9. etc.
	<ul style="list-style-type: none"> 1. strawberries 2. pineapple 3. non-dairy coffee creamer 4. orange juice 	<ol style="list-style-type: none"> 1. In blender add strawberries, pineapple, non-dairycreamer and orange juice and blend until smooth. 2. Pour into frosted glass and enjoy

Figure 1: Recipe samples from the Recipe1M Dataset.

a number of larger databases were introduced. Bossard et al. [2] introduced the Food-101 dataset, which is composed of 101 food categories represented by 101'000 food images. This was followed by several image-based databases, such as the UEC-100 [18] and its augmented version, the UEC-256 [13] dataset, with 9060 food images referring to 100 Japanese food types and 31651 food images referring to 256 Japanese food types, respectively. Xu et al. [26] developed a specialized dataset by including geolocation and external information about restaurants to simplify the food recognition task. Wang et al. [25] introduced the UPMC Food-101 multi-modal dataset, that shares the same 101 food categories with the popular Food-101 dataset, but contains textual information in addition. A number of studies have been carried out utilizing the aforementioned databases, mainly for the task of food recognition. Salvador et al. [22] published Recipe1M, the largest publicly available multi-modal dataset, that consists of 1 million recipes together with the accompanying images.

The emergence of multi-modal databases has led to novel approaches for meal image analysis. The fusion of visual features learned from images by deep Convolution Neural Networks (CNN) and textual features lead to outstanding results in food recognition applications. An early approach for recipe retrieval was based on jointly learning to predict food category and its ingredients using deep CNN [4]. In a following step, the predicted ingredients are matched against a large corpus of recipes. More recent approach is proposed by [22] and is based on jointly learning recipe-text and image representations in a shared latent space. Recurrent Neural Networks (RNN) and CNN are mainly used to map text and image

into the shared space. To align the text and image embedding vectors between matching recipe-image pairs, cosine similarity loss with margin was applied. Carvalho et al. [3] proposed a similar multi-modal embedding method for aligning text and image representations in a shared latent space. In contrast to Salvador et al. [22], they formulated a joint objective function which incorporates the loss for the cross-modal retrieval task and a classification loss, instead of using the latent space for a multitask learning setup. To address the challenge of encoding long sequences (like recipe instructions), [22] chose to represent single instructions as sentence embedding using the skip-thought technique [15]. These encoded instruction sentences are referred to as skip-instructions and their embedding is not fine tuned when learning the image-text joint embedding.

In this study, we present a method for the joint learning of meal image and recipe embedding, using a multi-path structure that incorporates natural language processing paths, as well as image analysis paths. The main contribution of the proposed method is threefold: i) the direct encoding of the instructions, ingredients and images during training, making the need of skip instruction embedding redundant; ii) the utilization of multiple attention mechanisms (i.e. self-attention and ingredient-attention), and iii) a lightweight architecture.

2 MATERIALS AND METHODS

2.1 Database

The proposed method is trained and evaluated on Recipe1M [22], the largest publicly available multi-modal food database. Recipe1M

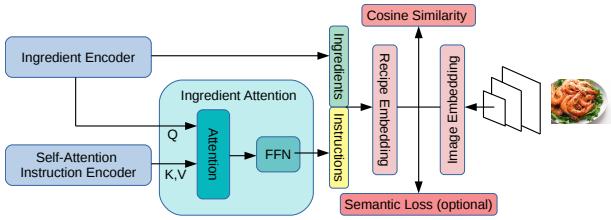


Figure 2: Text-image embedding model with optional semantic classifier for semantic regularization according to [17] and with Ingredient Attention based instruction encoding

provides over 1 million recipes (ingredients and instructions), accompanied by one or more images per recipe, leading to 13 million images. The large corpus is supplemented with semantic information (1048 meal classes) for injecting an additional source of information in potential models. In the table in Figure 1, the structure of recipes belonging to different semantic classes is displayed. Using a slightly adjusted pre-processing than that in [22] (elimination of noisy instruction sentences), the training set, validation set and test set contain 254,238 and 54,565 and 54,885 matching pairs, respectively. In [22], the authors chose the overall amount of instructions per recipe as one criterion for a valid matching pair. But we simply removed instruction sentences that contain only punctuation and gained some extra data for training and validation.

2.2 Model Architecture

The proposed model architecture is based on a multi-path approach for each of the involved input data types namely, instructions, ingredients and images, similarly to [17]. In Figure 2, the overall structure is presented. For the instruction encoder, we utilized a self-attention mechanism [24], which learns which words of the instructions are relevant with a certain ingredient. In order to encode the ingredients, a bidirectional RNN is used, since ingredients are an unordered list of words. All RNNs in the ingredients path were implemented with Long Short-Term Memory (LSTM) cells [11]. We fixed the ingredient representation to have a length of 600, independent of the amount of ingredients. Lastly, the outputs of the self-attention-instruction encoder with ingredient attention and the output of the bidirectional LSTM ingredient-encoder are concatenated and mapped to the joint embedding space. The image analysis path is composed of a ResNet-50 model [9], pretrained on the ImageNet Dataset [7], with a custom top layer for mapping the image features to the joint embedding space. All word embeddings are pretrained with the word2vec algorithm [19] and fine tuned during the joint embedding learning phase. We chose 512-dimensional word embedding for our model with self-attention, whereas [17] and [3] chose a vector length of 300. In the following sections, more details about the aforementioned paths are presented.

2.3 Attention Mechanisms

The instruction encoder follows a transformer based encoder, as suggested by [24]. Since we do not focus on syntactic rules, but mostly on weak sentence semantics or single words, we built a more shallow encoder containing only 2 stacked layers, where each

of this layers contains two sub-layers. The first is the multi-head attention layer, and the second is a position-wise densely connected feed-forward network (FFN). Due to recipes composed of over 600 words as instructions, we decided to trim words per instruction sentence to restrict the overall words per recipe to 300. In order to avoid removing complete instructions at the end of the instruction table, we removed a fraction of words from each instruction, based on this instruction’s length and the overall recipe-instruction length. This strategy reinforces the neglect of syntactic structures in the instruction encoding process. With such a model, we can directly perform the instruction encoding during the learning process for the joint embedding, thus saving training time and reducing disk space consumption. The transformer-like encoder does not make use of any recurrent units, thus providing the opportunity for a more lightweight architecture. By using self-attention [24], the model learns to focus on instructions relevant to recipe-retrieval-relevant, parts of instructions or single instruction-words. Furthermore we gain insight into which instructions are important to distinguish recipes with similar ingredients but different preparation styles.

The instruction encoder transforms the sequence of plain word representations with added positional information to a sequence of similarity-based weighted sum of all word representations. The outputted sequence of the encoder exhibits the same amount of positions as the input to the instruction encoder (in our experiments 300). Each of this positions is represented by a 512-dimensional vector. To obtain a meaningful representation without a vast number of parameters, we reduced the number of word representations before the concatenation with the ingredient representation. For this reduction step, we implemented a recipe-embedding specific attention layer where the ingredient representation is used to construct n queries, where n is the amount of new instruction representation vectors. Each of these new representations is a composition of all previous word representations weighted by the ingredient attention score. Following, the ingredient attention process is formulated mathematically and is visually portrayed in Figure 2.

$$IA(K(inst), V(inst), Q(ing)) = \text{Sofmax} \left(\frac{K(inst) * Q(ing)^T}{\sqrt{d_k}} \right) \quad (1)$$

where $K(inst)$ and $V(inst)$ are linear mappings of the encoded instruction words, and $Q(ing)$ is a linear mapping of the ingredient representation and d_k is the dimensionality of linearly projected position vectors.

$$K(inst) = inst * W_k, \text{ with } inst \in R^{b \times p \times w} \text{ and } W_k \in R^{w \times h}$$

$$V(inst) = inst * W_v, \text{ with } inst \in R^{b \times p \times w} \text{ and } W_v \in R^{w \times w}$$

$$Q(ing) = ing * W_q, \text{ with } inst \in R^{b \times q} \text{ and } W_q \in R^{n \times q \times h}$$

where b is the batch-size, p is the amount of word embeddings, w is the dimensionality of the word embedding, h is the dimensionality of the space to where we project the word embeddings and queries, q is the dimensionality of the ingredient representation and n is the amount of Ingredient Attention-based instruction representations. Ingredient Attention can be performed step-wise, similarly

to the well known dimensionality reduction in convolution neural networks.

2.4 Loss function

To align text and image embeddings of matching recipe-image pairs alongside each other, we maximize the cosine distance between positive pairs and minimize it between negative pairs.

We have trained our model using cosine similarity loss with margin as in [17] and with the triplet loss proposed by [3]. Both objective functions and the semantic regularization by [17] aim at maximizing intra-class correlation and minimizing inter-class correlation.

Let us define the text query embedding as ϕ^q and the embedding of the image query as ϕ^d , then the cosine embedding loss can be defined as follows:

$$L_{cos}(\phi^q, \phi^d, y) = \begin{cases} 1 - \cos(\phi^q, \phi^d), & \text{if } y = 1 \\ \max(0, \cos(\phi^q, \phi^d) - \alpha, & \text{if } y = -1 \end{cases}$$

where $\cos(x, y)$ is the normalized cosine similarity and α is a margin ($-1 \leq \alpha \leq 1$), that determines how similar negative pairs are allowed to be. Positive margins allow negative pairs to share at maximum α similarity, where a maximum margin of zero or negative margins allow no correlation between non matching embedding vectors or force the model to learn antiparallel representations, respectively. ϕ^d is the corresponding image counterpart to ϕ^q if $y = 1$ or a randomly chosen sample $\phi^d \in S \wedge \phi^d \neq \phi^{d(q)}$ if $y = -1$, where $\phi^{d(q)}$ is the true match for ϕ^q and S is the dataset we sample from it. Furthermore, we complement the cosine similarity with cross-entropy classification loss (L_{reg}), leading to the applied objective function.

$$L(\phi^q, \phi^d, c_r, c_v, y) = L_{cos}(\phi^q, \phi^d, y) + \lambda * L_{reg}(\phi^q, \phi^d, c_r, c_v) \quad (2)$$

with c_r and c_v as semantic recipe-class and semantic image-class, respectively, while $c_r = c_v$ if the food image and recipe text are a positive pair.

For the triplet loss, we define ϕ^q as query embedding, ϕ^{d+} as matching image counterpart and ϕ^{d-} as another random sample taken from S . Further $\phi^{d_{sem}+} \in S \wedge \phi^{d_{sem}+} \neq \phi^{d(q)}$ is a sample from S sharing the same semantic class as ϕ^q and $\phi^{d_{sem}-}$ is a sample from any other class. The triplet loss is formulated as follows:

$$L_{sample}(\phi^q, \phi^{d+}, \phi^{d-}) = [\alpha - \cos(\phi^q, \phi^{d+}) + \cos(\phi^q, \phi^{d-})]$$

$$L_{sem}(\phi^q, \phi^{d_{sem}+}, \phi^{d_{sem}-}) = [\alpha - \cos(\phi^q, \phi^{d_{sem}+}) + \cos(\phi^q, \phi^{d_{sem}-})]$$

$$\begin{aligned} L_{triplet}(\phi^q, \phi^{d+}, \phi^{d-}, \phi^{d_{sem}+}, \phi^{d_{sem}-}) = \\ \beta * L_{sample}(\phi^q, \phi^{d+}, \phi^{d-})^2 + \\ (1 - \beta) * (L_{sample}(\phi^q, \phi^{d+}, \phi^{d-})) + \\ \gamma * (\beta * L_{sem}(\phi^q, \phi^{d_{sem}+}, \phi^{d_{sem}-})^2 + \\ (1 - \beta) * (L_{sem}(\phi^q, \phi^{d_{sem}+}, \phi^{d_{sem}-}))) \end{aligned} \quad (3)$$

where $\beta \in [0, 1]$ weights between quadratic and linear loss, $\alpha \in [0, 2]$ is the margin and $\gamma \in [0, 1]$ weights between semantic-and sample-loss. The triplet loss encourages the embedding vectors of a matching pair to be larger by a margin above its non-matching counterpart. Further, the semantic loss encourages the model to form clusters of dishes, sharing the same class. We chose β to be 0.1, α to be 0.3 and γ to be 0.3.

2.5 Training configuration

We used Adam [14] optimizer with an initial learning rate of 10^{-4} . At the beginning of the training session, we freeze the pretrained ResNet-50 weights and optimize only the text-processing branch until we do no longer make progress. Then, we alternate train image and text branch until we switched modality for 10 times. Lastly, we fine-tune the overall model by releasing all trainable parameters in the model. Our optimization strategy differs from [17] in that we use an aggressive learning rate decay, namely exponential decay, so that the learning rate is halved all 20 epochs. Since the timing of freezing layers proved not to be of importance unless the recipe path is trained first, we used the same strategy under the cosine distance objective [17] and for the triplet loss [3].

3 EXPERIMENTAL SETUP AND RESULTS

Recipe1M is already distributed in three parts, the training, validation and testing sets. We did not make any changes to these partitions. Except with our more sensitive preprocessing algorithm, we accept more recipes from the raw corpus. [17] used 238,399 samples for their effective training set and for the validation and testing set 51,119 and 51,303 samples, respectively. By filtering out noisy instructions sentences (e.g. instructions containing only punctuation) we increased the effective dataset size to 254,238 samples for the training set and 54,565 and 54,885 for the validation and testing sets, respectively.

Similarly to [17] and [3], we evaluated our model on 10 subsets of 1000 samples each. One sample of these subsets is composed of text embedding and image embedding in the shared latent space. Since our interest lies in the recipe retrieval task, we optimized and evaluated our model by using each image embedding in the subsets as query against all text embeddings. By ranking the query and the candidate embeddings according to their cosine distance, we estimate the median rank. The model's performance is best, if the matching text embedding is found at the first rank. Further, we estimate the recall percentage at the top K percent over all queries. The recall percentage describes the quantity of queries ranked amid the top K closest results. In Table 1 the results are presented, in comparison to baseline methods.

Table 1: Comparison between our method, our Joint Neural Embedding (JNE)[17] and AdaMine [3] re-implementation. For all models we were using selected matching pairs generated by reducing noisy instruction sentences as described above. Recall rates are averaged over the evaluation batches.

		Image to Recipe			
		MedR	R@1	R@5	R@10
1k samples	Random [17]	500.0	0.001	0.005	0.01
	JNE [17]	5.0 ± 0.1	25.9	52.6	64.1
	AdaMine [3]	3.0 ± 0.1	33.1	64.3	75.2
	IA	2.9 ± 0.3	34.6	66.0	76.6

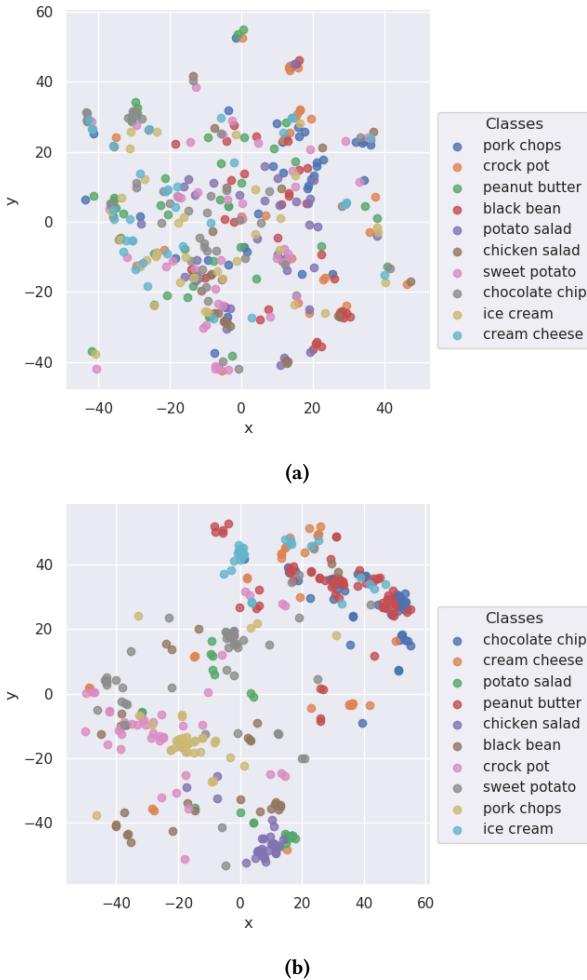


Figure 3: (a) Visualization of the joint embedding space under the cosine distance with semantic regularization objective. (b) organization of the joint embedding space under the triplet

Both [17] and [3] use time-consuming instruction text preprocessing over the skip-thought technique [15]. This process doubles the overall training time from three days to six days using two

Nvidia Titan X GPU's. By using online-instruction encoding with the self-attention encoder, we were able train the model for its main task in under 30 hours. Furthermore, the proposed approach offers more flexibility for dataset alterations.

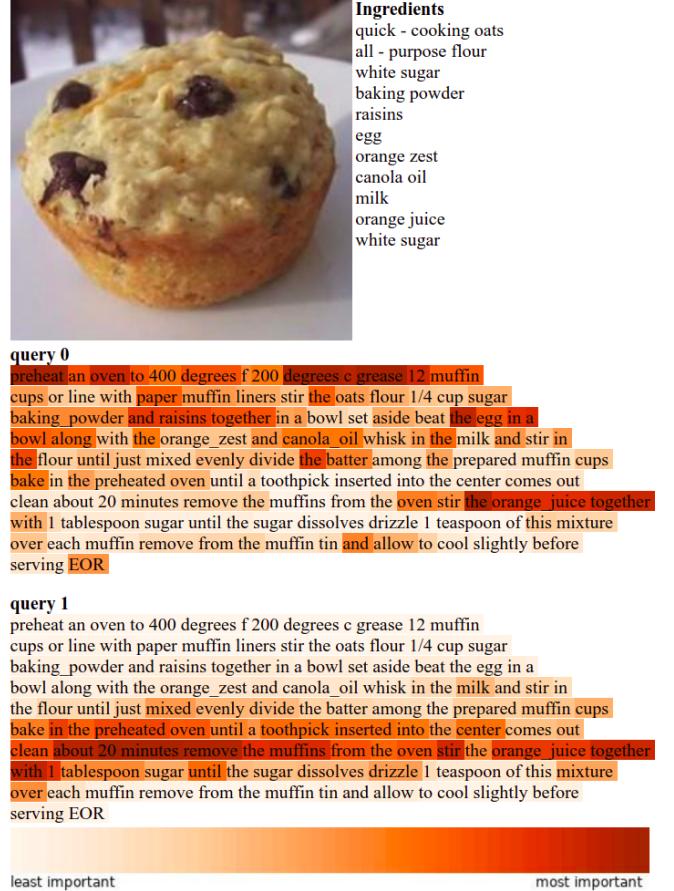


Figure 4: Ingredient-Attention based focus on instruction sentences. We use two different mapping matrices for the two ingredient based queries.

Qualitative results such as recipe retrieval, quality of the cluster formation in the joint embedding space and heat maps of instruction words are more important than the previously mentioned benchmarking scores. Depending on meal type, all baseline implementations as well as our Ingredient Attention based model exhibit a broad range of retrieval accuracy. In Figure 5 we present a few typical results on the intended recipe retrieval task.

AdaMine [3] creates more distinct class clusters than in [17]. In Figure 3, we demonstrate the difference in cluster formation using the aforementioned Methods for our Ingredient Attention. We visualize the top ten most common recipe classes in Recipe1M using t-SNE [23]. Since chocolate chip, peanut butter, cream cheese and/or ice cream are used as ingredients in desserts, due to semantic regularization inside the triplet loss, clusters of sweet meals are close together (Figure 3b top right corner).

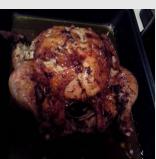
Sample 1	Sample 2	Sample 3
 1. marinade 2. whole chicken 3. salt 4. garlic powder 5. onion powder	 1. cooked chicken breasts 2. grapes 3. feta cheese 4. celery 5. dried onion flakes 6. cashews 7. etc.	 1. water 2. crabmeat 3. parmesan cheese 4. lemon juice 5. mayonnaise 6. black pepper
 1. whole chicken 2. ground black pepper 3. onion powder 4. cayenne pepper 5. garlic powder 6. garlic cloves 7. dried thyme 8. butter	 1. cooked chicken breasts 2. celery 3. mayonnaise 4. sour cream 5. fresh rosemary	 1. 2 can Cream of Chicken 2. 2 can Cream of mushroom 3. Cooked chicken 4. egg noodles 5. onion 6. ream cheese 7. etc.
 1. chicken drumsticks 2. sea salt 3. ground black pepper 4. garlic powder 5. onion powder 6. dried oregano 7. dried chipotle powder 8. ground cumin 9. coconut sugar crystals	 1. tuna 2. celery 3. sweet pickle 4. mayonnaise	 1. fettuccine 2. cooked chicken breasts 3. cream of chicken soup 4. sour cream 5. onions 6. dried parsley 7. white wine 8. cheddar cheese 9. salt
 1. whole chicken 2. lime 3. paprika 4. cumin 5. salt and pepper	 1. 1 can Canned tuna 2. celery 3. dill 4. lemon juice 5. mayonnaise 8. salt and pepper	 1. chicken 2. noodles 3. mayonnaise 4. onions 5. salt 6. cream of mushroom soup 7. milk 8. cheddar cheese

Figure 5: The retrieval performance of our model depends heavily on the meal type. We marked matching retrieved ingredients or those of the same family in green. The Ingredient Attention model performed well on Sample 1, and acceptably on Sample 2. On Sample 3, the model missed the main ingredient in all top three retrievals.

We use heat maps on instruction words as tool to visualize words relevant to ingredient-lists in plain instruction text. In Figure 4, we demonstrate how easily we can achieve insight into the models decision making.

4 CONCLUSIONS

In this paper, we have introduced self-attention for instruction encoding in the context of the recipe retrieval task and ingredient attention for disclosing ingredient dependent meal preparation steps. Our main contribution is the aforementioned ingredient attention, empowering our model to solve the recipe retrieval without any upstream skip instruction embedding, as well as the light-weight architecture provided by the transformer-like instruction encoder. On the recipe retrieval task, our method performs similarly to our baseline implementation of [3]. Regarding training time on the other hand, we increased the efficiency significantly for cross-modal based retrieval methods. There is no need for a maximum number of instructions for a recipe to be considered as valid for training or testing; only for total words, making more samples of the large Recipe1M corpus usable for training. Through ingredient

attention, we are able to unveil internal focus in the text processing path by observing attention weights. Incorporation of new samples in the train set can be done by retraining just one model. Overall, an accurate and flexible method for recipe retrieval from meal images could provide downstream models (e.g. automatic nutrient content estimation) with decisive information and significantly improve their results.

REFERENCES

- [1] Eduardo Aguilar, Beatriz Remeseiro, Marc Bolaños, and Petia Radeva. 2017. Grab, Pay and Eat: Semantic Food Detection for Smart Restaurants. *CoRR* abs/1711.05128 (2017). arXiv:1711.05128 <http://arxiv.org/abs/1711.05128>
- [2] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. 2014. Food-101 – Mining Discriminative Components with Random Forests. In *Computer Vision – ECCV 2014*, David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars (Eds.), Springer International Publishing, Cham, 446–461.
- [3] Micael Carvalho, Rémi Cadène, David Picard, Laure Soulier, Nicolas Thome, and Matthieu Cord. 2018. Cross-Modal Retrieval in the Cooking Context: Learning Semantic Text-Image Embeddings. *CoRR* abs/1804.11146 (2018). arXiv:1804.11146 <http://arxiv.org/abs/1804.11146>
- [4] Jingjing Chen and Chong-Wah Ngo. 2016. Deep-based Ingredient Recognition for Cooking Recipe Retrieval. 32–41. <https://doi.org/10.1145/2964284.2964315>
- [5] Mei Chen, Kapil Dhingra, Wen Wu, Lei Yang, Rahul Sukthankar, and Jie Yang. [n.d.] PFID: PITTSBURGH FAST-FOOD IMAGE DATASET.

- [6] J. Dehais, M. Anthimopoulos, S. Shevchik, and S. Mougiakakou. 2017. Two-View 3D Reconstruction for Food Volume Estimation. *IEEE Transactions on Multimedia* 19, 5 (May 2017), 1090–1099. <https://doi.org/10.1109/TMM.2016.2642792>
- [7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. 2009. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*.
- [8] Nita G Forouhi and Nigel Unwin. 2019. Global diet and health: old questions, fresh evidence, and new horizons. *The Lancet* 393, 10184 (2019), 1916 – 1918. [https://doi.org/10.1016/S0140-6736\(19\)30500-8](https://doi.org/10.1016/S0140-6736(19)30500-8)
- [9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. *CoRR* abs/1512.03385 (2015). arXiv:1512.03385 <http://arxiv.org/abs/1512.03385>
- [10] Y. He, C. Xu, N. Khanna, C. J. Boushey, and E. J. Delp. 2013. Food image analysis: Segmentation, identification and weight estimation. In *2013 IEEE International Conference on Multimedia and Expo (ICME)*. 1–6. <https://doi.org/10.1109/ICME.2013.6607548>
- [11] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Comput.* 9, 8 (Nov. 1997), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- [12] N. P. Holmes and C. Spence. 2005. Multisensory integration: Space, time and superadditivity. *Current Biology* 15 (Spt 2005), R762–R764.
- [13] Y. Kawano and K. Yanai. 2014. Automatic Expansion of a Food Image Dataset Leveraging Existing Categories with Domain Adaptation. In *Proc. of ECCV Workshop on Transferring and Adapting Source Knowledge in Computer Vision (TASK-CV)*.
- [14] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. <http://arxiv.org/abs/1412.6980> cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.
- [15] Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2015. Skip-Thought Vectors. *CoRR* abs/1506.06726 (2015). arXiv:1506.06726 <http://arxiv.org/abs/1506.06726>
- [16] Ya Lu, Dario Allegra, Marios Anthimopoulos, Filippo Stanco, Giovanni Maria Farinella, and Stavroula G. Mougiakakou. 2018. A Multi-Task Learning Approach for Meal Assessment. *CoRR* abs/1806.10343 (2018). arXiv:1806.10343 <http://arxiv.org/abs/1806.10343>
- [17] Javier Marin, Aritro Biswas, Ferda Ofli, Nicholas Hynes, Amaia Salvador, Yusuf Aytar, Ingmar Weber, and Antonio Torralba. 2018. RecipeIM: A Dataset for Learning Cross-Modal Embeddings for Cooking Recipes and Food Images. *CoRR* abs/1810.06553 (2018). arXiv:1810.06553 <http://arxiv.org/abs/1810.06553>
- [18] Y. Matsuda, H. Hoashi, and K. Yanai. 2012. Recognition of Multiple-Food Images by Detecting Candidate Regions. In *Proc. of IEEE International Conference on Multimedia and Expo (ICME)*.
- [19] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and Their Compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2 (NIPS'13)*. Curran Associates Inc., USA, 3111–3119. <http://dl.acm.org/citation.cfm?id=2999792.2999959>
- [20] U.S. Department of Agriculture. 2019. Agricultural Research Service. FoodData Central.
- [21] L. Pan, S. Pouyanfar, H. Chen, J. Qin, and S. Chen. 2017. DeepFood: Automatic Multi-Class Classification of Food Ingredients Using Deep Learning. In *2017 IEEE 3rd International Conference on Collaboration and Internet Computing (CIC)*. 181–189. <https://doi.org/10.1109/CIC.2017.00033>
- [22] A. Salvador, N. Hynes, Y. Aytar, J. Marin, F. Ofli, I. Weber, and A. Torralba. 2017. Learning Cross-Modal Embeddings for Cooking Recipes and Food Images. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 3068–3076. <https://doi.org/10.1109/CVPR.2017.327>
- [23] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE.
- [24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. *CoRR* abs/1706.03762 (2017). arXiv:1706.03762 <http://arxiv.org/abs/1706.03762>
- [25] Xin Wang, D. Kumar, N. Thome, M. Cord, and F. Precioso. 2015. Recipe recognition with large multimodal food dataset. In *2015 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*. 1–6. <https://doi.org/10.1109/ICMEW.2015.7169757>
- [26] R. Xu, L. Herranz, S. Jiang, S. Wang, X. Song, and R. Jain. 2015. Geolocalized Modeling for Dish Recognition. *IEEE Transactions on Multimedia* 17, 8 (Aug 2015), 1187–1199. <https://doi.org/10.1109/TMM.2015.2438717>
- [27] K. Yanai and Y. Kawano. 2015. Food image recognition using deep convolutional network with pre-training and fine-tuning. In *2015 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*. 1–6. <https://doi.org/10.1109/ICMEW.2015.7169816>