

Cross-Modal Retrieval in the Cooking Context: Learning Semantic Text-Image Embeddings

Micael Carvalho*
Sorbonne Université, CNRS, LIP6,
F-75005 Paris, France
micael.carvalho@lip6.fr

Rémi Cadène*
Sorbonne Université, CNRS, LIP6,
F-75005 Paris, France
remi.cadene@lip6.fr

David Picard
Sorbonne Université, CNRS, LIP6,
F-75005 Paris, France
david.picard@lip6.fr

Laure Soulier
Sorbonne Université, CNRS, LIP6,
F-75005 Paris, France
laure.soulier@lip6.fr

Nicolas Thome
CEDRIC - Conservatoire National des
Arts et Métiers, F-75003 Paris, France
nicolas.thome@cnam.fr

Matthieu Cord
Sorbonne Université, CNRS, LIP6,
F-75005 Paris, France
matthieu.cord@lip6.fr

ABSTRACT

Designing powerful tools that support cooking activities has rapidly gained popularity due to the massive amounts of available data, as well as recent advances in machine learning that are capable of analyzing them. In this paper, we propose a cross-modal retrieval model aligning visual and textual data (like pictures of dishes and their recipes) in a shared representation space. We describe an effective learning scheme, capable of tackling large-scale problems, and validate it on the Recipe1M dataset containing nearly 1 million picture-recipe pairs. We show the effectiveness of our approach regarding previous state-of-the-art models and present qualitative results over computational cooking use cases.

CCS CONCEPTS

• **Information systems** → **Multimedia and multimodal retrieval**; • **Computer systems organization** → **Neural networks**;

KEYWORDS

Deep Learning, Cross-modal Retrieval, Semantic Embeddings

ACM Reference Format:

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. In *SIGIR '18: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, July 8–12, 2018, Ann Arbor, MI, USA*. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3209978.3210036>

1 INTRODUCTION

Designing powerful tools that support cooking activities has become an attractive research field in recent years due to the growing

*Equal contribution

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 the author(s) 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.

SIGIR '18, July 8–12, 2018, Ann Arbor, MI, USA

© 2018 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-5657-2/18/07...\$15.00

<https://doi.org/10.1145/3209978.3210036>

interest of users to eat home-made food and share recipes on social platforms [34]. These massive amounts of data shared on devoted sites, such as *All Recipes*¹, allow gathering food-related data including text recipes, images, videos, and/or user preferences. Consequently, novel applications are rising, such as ingredient classification [7], recipe recognition [38] or recipe recommendation [34]. However, solving these tasks is challenging since it requires taking into consideration 1) the heterogeneity of data in terms of format (text, image, video, ...) or structure (e.g., list of items for ingredients, short verbal sentence for instructions, or verbose text for users' reviews); and 2) the cultural factor behind each recipe since the vocabulary, the quantity measurement, and the flavor perception is culturally intrinsic; preventing the homogeneous semantics of recipes.

One recent approach emerging from the deep learning community aims at learning the semantics of objects in a latent space using the *distributional hypothesis* [14] that constrains object with similar meanings to be represented similarly. First used for learning image representations (also called *embeddings*), this approach has been derived to text-based applications, and recently some researchers investigate the potential of representing multi-modal evidence sources (e.g., texts and images) in a shared latent space [20, 30]. This research direction is particularly interesting for grounding language with common sense information extracted from images, or vice-versa. In the context of computer-aided cooking, we believe that this multi-modal representation learning approach would contribute to solving the heterogeneity challenge, since they would promote a better understanding of each domain-specific word/image/video. In practice, a typical approach consists in aligning text and image representations in a shared latent space in which they can be compared [6, 8, 20, 21, 24, 33]. One direct application in the cooking context is to perform cross-modal retrieval where the goal is to retrieve images similar to a text recipe query or conversely text recipes similar to a image query. However, Salvador et al. [33] highlight that this solution based on aligning matching pairs can lead to poor retrieval performances in a large scale framework. Training the latent space by only matching pairs of the exact same dish is not particularly effective at mapping similar dishes close together, which induces a lack of generalization to newer items (recipes or images). To alleviate this problem, [33] proposes to use additional

¹<http://www.allrecipes.com/>

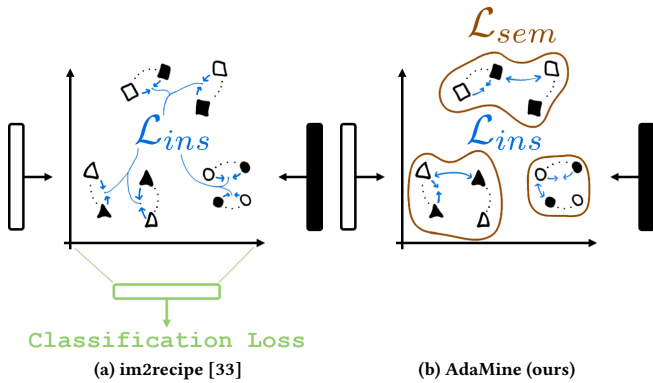


Figure 1: Comparison between (a) the classification augmented latent space learning of [33] and (b) our joint retrieval and semantic latent space learning, which combines instance-based (\mathcal{L}_{ins}) and semantic-based (\mathcal{L}_{sem}) losses.

data (namely, categories of meals) to train a classifier with the aim of regularizing the latent space embeddings (Figure 1a). Their approach involves adding an extra layer to a deep neural network, specialized in the classification task. However, we believe that such classification scheme is under-effective for two main reasons. First, the classifier adds many parameters that are discarded at the end of the training phase, since classification is not a goal of the system. Second, we hypothesize that given its huge number of parameters, the classifier can be trained with high accuracy without changing much of the structure of the underlying latent space, which completely defeats the original purpose of adding classification information.

To solve these issues, we propose a unified learning framework in which we simultaneously leverage retrieval and class-guided features in a shared latent space (Figure 1b). Our contributions are three-fold:

- We formulate a joint objective function with cross-modal retrieval and classification loss to structure the latent space. Our intuition is that directly injecting the class-based evidence sources in the representation learning process is more effective at enforcing a high-level structure to the latent space as shown in Figure 1b.

- We propose a double-triplet scheme to express jointly 1) the retrieval loss (e.g., corresponding picture and recipe of a pizza should be closer in the latent space than any other picture - see blue arrows in Figure 1b) and 2) the class-based one (e.g., any 2 pizzas should be closer in the latent space than a pizza and another item from any other class, like salad). This double-loss is capable of taking into consideration both the fine-grained and the high-level semantic information underlying recipe items. Contrary to the proposal of [33], our class-based loss acts directly on the feature space, instead of adding a classification layer to the model.

- We introduce a new scheme to tune the gradient update in the stochastic gradient descent and back-propagation algorithms used for training our model. More specifically, we improve over the usual gradient averaging update by performing an adaptive mining of the most significant triplets, which leads to better embeddings.

We instantiate these contributions through a dual deep neural network. We thoroughly evaluate our model on Recipe1M [33], the

only English large-scale cooking dataset available, and show its superior performances compared to the state of the art models.

The remaining of this paper is organized as follows. In Section 2, we present previous work related to computer-aided cooking and cross-modal retrieval. Then, we present in Section 3 our joint retrieval and classification model as well as our adaptive triplet mining scheme to train the model. We introduce the experimental protocol in Section 4. In Section 5 we experimentally validate our hypotheses and present quantitative and qualitative results highlighting our model effectiveness. Finally, we conclude and discuss perspectives.

2 RELATED WORK

2.1 Computational cooking

Cooking is one of the most fundamental human activities connected to various aspects of human life such as food, health, dietary, culinary art, and so on. This is more particularly perceived on social platforms in which people share recipes or their opinions about meals, also known as the *eat and tweet* or *food porn* phenomenon [2]. Users’ needs give rise to smart cooking-oriented tasks that contribute towards the definition of computational cooking as a research field by itself [1]. Indeed, the research community is very active in investigating issues regarding food-related tasks, such as ingredient identification [7], recipe recommendation [10], or recipe popularity prediction [34]. A first line of work consists in leveraging the semantics behind recipe texts and images using deep learning approaches [33, 34]. The objective of such propositions is to generally align different modalities in a shared latent space to perform cross-modal retrieval or recommendation. For instance, Salvador et al. [33] introduce a dual neural network that aligns textual and visual representations under both the distributional hypothesis and classification constraints [33]. A second line of work [10, 26, 37] aims at exploiting additional information (e.g., calories, biological or economical factors) to bridge the gap between computational cooking and healthy issues. For instance, Kusmierczyk et al. [26] extend the Latent Dirichlet Algorithm (LDA) for combining recipe descriptions and nutritional related meta-data to mine latent recipe topics that could be exploited to predict nutritional values of meals.

These researches are boosted by the release of food-related datasets [6, 9, 11, 22]. As a first example, [9] proposes the Pittsburgh fast-food image dataset, containing 4,556 pictures of fast-food plates. In order to solve more complex tasks, other initiatives provide richer sets of images [6, 7, 13, 33, 38]. For example, [6] proposes the Food-101 dataset, containing around 101,000 images of 101 different categories. Additional meta-data information is also provided in the dataset of [5] which involves GPS data or nutritional values. More recently, two very large-scale food-related datasets, respectively in English and Japanese, are released by [33] and [13]. The *Cookpad* dataset [13] gathers more than 1 million of recipes described using structured information, such as recipe description, ingredients, and process steps as well as images. Salvador et al. [33] have collected a very large dataset with nearly 1 million recipes, with about associated 800,000 images. They also add extra information corresponding to recipe classes and show how this new semantic information may be helpful to improve deep cross-modal retrieval systems. To the best of our knowledge, this dataset [33] is

the only large-scale English one including a huge pre-processing step for cleaning and formatting information. The strength of this dataset is that it is composed of structured ingredients and instructions, images, and a large number of classes as well. Accordingly, we focus all our experimental evaluation on this dataset.

2.2 Cross-modal Retrieval

Cross-modal retrieval aims at retrieving relevant items that are of different nature with respect to the query format; for example when querying an image dataset with keywords (image vs. text) [33]. The main challenge is to measure the similarity between different modalities of data. In the Information Retrieval (IR) community, early work have circumvented this issue by annotating images to perceive their underlying semantics [19, 36]. However, these approaches generally require a supervision from users to annotate at least a small sample of images. An unsupervised solution has emerged from the deep learning community which consists in mapping images and texts into a shared latent space \mathcal{F} in which they can be compared [40]. In order to align the text and image manifolds in \mathcal{F} , the most popular strategies are based either on 1) global alignment methods aiming at mapping each modal manifold in \mathcal{F} such that semantically similar regions share the same directions in \mathcal{F} ; 2) local metric learning approaches aiming at mapping each modal manifold such that semantically similar items have a short distances in \mathcal{F} .

In the first category of works dealing with global alignment methods, a well-known state-of-the-art model is provided by the Canonical Correlation Analysis (CCA) [17] which aims at maximizing the correlation in \mathcal{F} between relevant pairs from data of different modalities. CCA and its variations like Kernel-CCA [4, 28] and Deep-CCA [3] have been successfully applied to align text and images [42]. However, global alignment strategies such as CCA do not take into account dissimilar pairs, and thus tend to produce false positives in cross-modal retrieval tasks.

In the second category of work, local metric learning approaches consider cross-modal retrieval as a ranking problem where items are ranked according to their distance to the query in the latent space. A perfect retrieval corresponds to a set of inequalities in which the distances between the query and relevant items are smaller than the distances between the query and irrelevant items [29, 39, 41]. Each modal projection is then learned so as to minimize a loss function that measures the cost of violating each of these inequalities [20, 24]. In particular, [12, 33] consider a loss function that minimizes the distance between pairs of matching cross-modal items while maximizing the distance between non-matching cross-modal pairs. However, ranking inequality constraints are more naturally expressed by considering triplets composed of a query, a relevant item, and an irrelevant item. This strategy is similar to the Large Margin Nearest Neighbor loss [39] in which a penalty is computed only if the distance between the query and the relevant item is larger than the distance between the query and the irrelevant item.

Our contributions differ from previous work according to three main aspects. First, we propose to model the manifold-alignment issue, which is generally based only on the semantic information [4, 17, 28, 42], as a joint learning framework leveraging retrieval

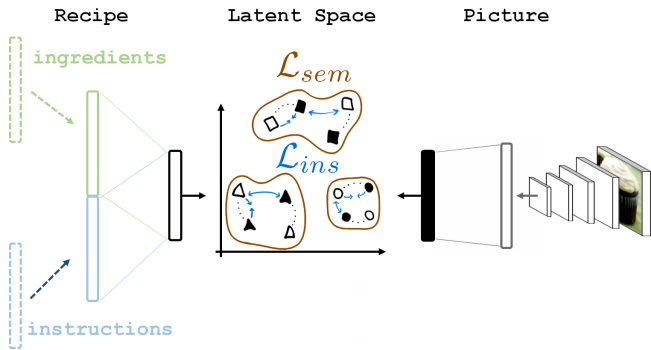


Figure 2: AdaMine overview.

and class-based features. In contrast to [33] which adds an additional classification layer to a manifold-alignment neural model, we directly integrate semantic information in the loss to refine the structure of the latent space while also limiting the number of parameters to be learned. Second, our model relies on a double-triplet (instead of pairwise learning in [12, 33] or a single triplet as in [39]) to fit with the joint learning objectives. Third, we propose a new stochastic gradient descent weighting scheme adapted to such a dual deep embedding architecture, which is computed on minibatches and automatically performs an adaptive mining of informative triplets.

3 ADAMINE DEEP LEARNING MODEL

3.1 Model Overview

The objective of our model **AdaMine** (ADaptive MINing Embedding) is to learn the representation of recipe items (texts and images) through a joint retrieval and classification learning framework based on a double-triplet learning scheme. More particularly, our model relies on the following hypotheses:

- H1: Aligning items according to a retrieval task allows capturing the fine-grained semantics of items, since the obtained embeddings must rank individual items with respect to each other.
- H2: Aligning items according to class meta-data allows capturing the high-level semantic information underlying items since it ensures the identification of item clusters that correspond to class-based meta-data.
- H3: Learning simultaneously retrieval and class-based features allows enforcing a multi-scale structure within the latent space, which covers all aspects of item semantics. In addition, we conjecture that adding a classification layer sequentially to manifold-alignment as in [33] might be under-effective.

Based on these hypotheses, we propose to learn the latent space structure (and item embeddings) by integrating both retrieval objective and semantic information in a single cross-modal metric learning problem (see the Latent Space in Figure 2). We take inspiration from the learning-to-rank retrieval framework by building a learning schema based on query/relevant item/irrelevant item triplets noted (x_q, x_p, x_n) . Following hypothesis H3, we propose a double-triplet learning scheme that relies on both instance-based and semantic-based triplets, noted respectively (x_q, x_p, x_n) and (x'_q, x'_p, x'_n) , in order to satisfy the multi-level structure (fine-grained and high-level) underlying semantics. More particularly,

we learn item embeddings by minimizing the following objective function:

$$\mathcal{L}_{total}(\theta) = \mathcal{L}_{ins}(\theta) + \lambda \mathcal{L}_{sem}(\theta) \quad (1)$$

where θ is the network parameter set. \mathcal{L}_{ins} is the loss associated with the retrieval task over instance-based triplets (x_q, x_p, x_n) , and \mathcal{L}_{sem} is the loss coming with the semantic information over semantic-based triplets (x'_q, x'_p, x'_n) . Unlike [33] that expresses this second term \mathcal{L}_{sem} acting as a regularization over the \mathcal{L}_{ins} optimization, in our framework, it is expressed as a joint classification task.

This double-triplet learning framework is a difficult learning problem since the trade-off between \mathcal{L}_{ins} and \mathcal{L}_{sem} is not only influenced by λ but also by the sampling of instance-based and semantic-based triplets and depends on their natural distribution. Furthermore, the sampling of violating triplets can be difficult as the training progresses which usually leads to vanishing gradient problems that are common in triplet-based losses, and are amplified by our double-triplet framework. To alleviate these problems, we propose an adaptive sampling strategy that normalizes each loss allowing to fully control the trade-off with λ alone while also ensuring non-vanishing gradients throughout the learning process.

In the following, we present the network architecture, each component of our learning framework, and then discuss the learning scheme of our model.

3.2 Multi-modal Learning Framework

3.2.1 Network Architecture. Our network architecture is based on the proposal of [33], which consists of two branches based on deep neural networks that map each modality (image or text recipe) into a common representation space, where they can be compared. Our global architecture is depicted in Figure 2.

The image branch (top-right part of Figure 2) is composed of a ResNet-50 model [15]. It contains 50 convolutional layers, totaling more than 25 million parameters. This architecture is further detailed in [15], and was chosen in order to obtain comparable results to [33] by sharing a similar setup. The ResNet-50 is pretrained on the large-scale dataset of the ImageNet Large Scale Visual Recognition Challenge [32], containing 1.2 million images, and is fine-tuned with the whole architecture. This neural network is followed by a fully connected layer, which maps the outputs of the ResNet-50 into the latent space, and is trained from scratch.

In the recipe branch (top-left part of Figure 2), ingredients and instructions are first embedded separately, and their obtained representations are then concatenated as input of a fully connected layer that maps the recipe features into the latent space. For ingredients, we use a bidirectional LSTM [16] on pretrained embeddings obtained with the word2vec algorithm [31]. With the objective to consider the different granularity levels of the instruction text, we use a hierarchical LSTM in which the word-level is pretrained using the skip-thought technique [25] and is not fine-tuned while the sentence-level is learned from scratch.

3.2.2 Retrieval loss. The objective of the retrieval loss \mathcal{L}_{ins} is to learn item embeddings by constraining the latent space according to the following assumptions (Hypothesis H1): 1) ranking items according to a similarity metric so as to gather matching items together and 2) discriminating irrelevant ones. We propose to use a

loss function ℓ_{ins} based on a particular triplet (x_q, x_p, x_n) consisting of a query x_q , its matching counterpart in the other modality x_p and a dissimilar item x_n . The retrieval loss function \mathcal{L}_{ins} is the aggregation of the individual loss ℓ_{ins} over all triplets. The aim of ℓ_{ins} is to provide a fine-grained structure to the latent space where the nearest item from the other modality with respect to the query is optimized to be its matching pair. More formally, the individual retrieval loss $\ell_{ins}(\theta, x_q, x_p, x_n)$ is formalized as follows:

$$\ell_{ins}(\theta, x_q, x_p, x_n) = [d(x_q, x_p) + \alpha - d(x_q, x_n)]_+ \quad (2)$$

where $d(x, y)$ expresses the cosine distance between vectors x and y in the latent space \mathcal{F} .

3.2.3 Semantic loss. \mathcal{L}_{sem} is acting as a regularizer capable of taking advantage of semantic information in the multi-modal alignment, without adding extra parameters to the architecture nor graph dependencies. To leverage class information (Hypotheses H2), we propose to construct triplets that optimize a surrogate of the k-nearest neighbor classification task. Ideally, for a given query x_q , and its corresponding class $c(x_q)$, we want its associated closest sample $x_{\star, q}$ in the feature space to respect $c(x_q) = c(x_{\star, q})$. This enforces a semantic structure on the latent space by making sure that related dishes are closer to each other than to non-related ones. To achieve this, we propose the individual triplet loss ℓ_{sem} :

$$\ell_{sem}(\theta, x'_q, x'_p, x'_n) = [d(x'_q, x'_p) + \alpha - d(x'_q, x'_n)]_+ \quad (3)$$

where x'_p belongs to the set of items with the same semantic class $c(x'_q)$ as the query, and x'_n belongs to the set of items with different semantic classes than the one of the query.

Contrary to the classification machinery adopted by [33], ℓ_{sem} optimizes semantic relations directly in the latent space without changing the architecture of the neural network, as shown on Figure 1. This promotes a smoothing effect on the space by encouraging instances of the same class to stay closer to each other.

3.3 Adaptive Learning Schema

As commonly used in Deep Learning, we use the stochastic gradient descent (SGD) algorithm which approximates the true gradient over mini-batches. The update term is generally computed by aggregation of the gradient using the *average* over all triplets in the mini-batch. However, this *average* strategy tends to produce a vanishing update with triplet losses. This is especially true towards the end of the learning phase, as the few active constraints are averaged with many zeros coming from the many inactive constraints. We believe this problem is amplified as the size of the training set grows. To tackle this issue, our proposed *adaptive* strategy considers an update term δ_{adm} that takes into account informative triplets only (*i.e.*, non-zero loss). More formally, given a mini-batch \mathbb{B} , \mathbb{P}'_q the set of matching items with respect to a query x_q and \mathbb{P}^s_q the set of items with the same class as x_q , the update term δ_{adm} is defined by:

$$\delta_{adm} = \sum_{x_q \in \mathbb{B}} \left(\sum_{x_p \in \mathbb{B} \cap \mathbb{P}'_q} \sum_{x_n \in \mathbb{B} \setminus \mathbb{P}'_q} \frac{\nabla \ell_{ins}(\theta, x_q, x_p, x_n)}{\beta'_r} + \sum_{x_p \in \mathbb{B} \cap \mathbb{P}^s_q} \sum_{x_n \in \mathbb{B} \setminus \mathbb{P}^s_q} \lambda \frac{\nabla \ell_{sem}(\theta, x_q, x_p, x_n)}{\beta'_s} \right) \quad (4)$$

with β'_r and β'_s being the number of triplets contributing to the cost:

$$\begin{aligned}\beta'_r &= \sum_{x_q \in \mathbb{B}} \sum_{x_p \in \mathbb{B} \cap \mathbb{P}_q^r} \sum_{x_n \in \mathbb{B} \setminus \mathbb{P}_q^r} \mathbb{1}_{\ell_{ins} \neq 0} \\ \beta'_s &= \sum_{x_q \in \mathbb{B}} \sum_{x_p \in \mathbb{B} \cap \mathbb{P}_q^s} \sum_{x_n \in \mathbb{B} \setminus \mathbb{P}_q^s} \mathbb{1}_{\ell_{sem} \neq 0}\end{aligned}\quad (5)$$

At the very beginning of the optimization, all triplets contribute to the cost and, as constraints stop being violated, they are dropped. At the end of the training phase, most of the triplets will have no contribution, leaving the hardest negatives to be optimized without vanishing gradient issues. Remark that this corresponds to a curriculum learning starting with the average strategy and ending with the hard negative strategy like in [35], but without the burden of finding the time-step at which to switch between strategies as this is automatically controlled by the weights β_r and β_s .

Remark also that an added benefit of δ_{adm} is due to the independent normalization of each loss by its number of active triplets. Thus δ_{adm} keeps the trade-off between ℓ_{ins} and ℓ_{sem} unaffected by difference between the number of active triplets in each loss and allows λ to be the only effective control parameter.

4 EVALUATION PROTOCOL

The objective of our evaluation is threefold: 1) Analyzing the impact of our semantic loss that directly integrates semantic information in the latent space; 2) Testing the effectiveness of our model; 3) Exploring the potential of our model and its learned latent space for solving smart cooking tasks. All of our experiments are conducted using PyTorch², with our own implementation³ of the experimental setup (*i.e.* preprocessing, architecture and evaluation procedures) described by [33]. We detail the experimental setup in the following.

4.1 Dataset

We use the Recipe1M dataset [33], the only large-scale dataset including both English cooking recipes (ingredients and instructions), images, and categories. The raw Recipe1M dataset consists of about 1 million image and recipe pairs. It is currently the largest one in English, including twice as many recipes as [27] and eight times as many images as [7]. Furthermore, the availability of semantic information makes it particularly suited to validate our model: around half of the pairs are associated with a class, among 1048 classes parsed from the recipe titles. Using the same preprocessed pairs of recipe-image provided by [33], we end up with 238,399 matching pairs of images and recipes for the training set, while the validation and test sets have 51,119 and 51,303 matching pairs, respectively.

4.2 Evaluation Methodology

We carry out a cross-modal retrieval task following the process described in [33]. Specifically, we first sample 10 unique subsets of 1,000 (1k setup) or 5 unique subsets of 10,000 (10k setup) matching text recipe-image pairs in the test set. Then, we consider each item in a modality as a query (for instance, an image), and we rank items in the other modality (*resp.* text recipes) according to the cosine distance between the query embedding and the candidate embeddings. The objective is to retrieve the associated item in the

other modality at the first rank. The retrieved lists are evaluated using standard metrics in cross-modal retrieval tasks. For each subset (1k and 10k), we estimate the median retrieval rank (MedR), as well as the recall percentage at top K (R@K), over all queries in a modality. The R@K corresponds to the percentage of queries for which the matching item is ranked among the top K closest results.

4.3 Baselines

To test the effectiveness of our model **AdaMine**, we evaluate our multi-modal embeddings with respect to those obtained by state-of-the-art (SOTA) baselines:

- **CCA**, which denotes the Canonical Correlation Analysis method [17]. This baseline allows testing the effectiveness of global alignment methods.
- **PWC**, the pairwise loss with the classification layer from [33]. We report their state-of-the-art results for the 1k and 10k setups when available. This baseline exploits the classification task as a regularization of embedding learning.
- **PWC***, our implementation of the architecture and loss described by [33]. The goal of this baseline is to assess the results of its improved version **PWC++**, described below.
- **PWC++**, the improved version of our implementation **PWC***. More particularly, we add a positive margin to the pairwise loss adopted in [33], as proposed by [18]:

$$\begin{aligned}\ell_{pwc++}(\theta, x_q, x) &= y [d(x_q, x) - \alpha_{pos}]_+ \\ &\quad + (1 - y) [\alpha_{neg} - d(x_q, x)]_+\end{aligned}\quad (6)$$

with $y = 1$ (*resp.* $y = 0$) for pos. (*resp.* neg.) pairs. The positive margin α_{pos} allows matching pairs to have different representations, thus reducing the risk of overfitting. In practice, the positive margin is set to 0.3 and the negative margin to 0.9.

We evaluate the effectiveness of our model **AdaMine**, which includes both the triplet loss and the adaptive learning, in different setups, and having the following objectives:

- Evaluating the impact of the retrieval loss: we run the **AdaMine_ins** scenario which refers to our model with the instance loss \mathcal{L}_{ins} only and the adaptive learning strategy (the semantic loss \mathcal{L}_{sem} is discarded);
- Evaluating the impact of the semantic loss: we run the **AdaMine_sem** scenario which refers to our model with the semantic loss \mathcal{L}_{sem} only and the adaptive learning strategy (the instance loss \mathcal{L}_{ins} is discarded);
- Evaluating the impact of the strategy used to tackle semantic information: we run the **AdaMine_ins+cls** scenario which refers to our **AdaMine** model by replacing the semantic loss by the classification head proposed by [33];
- Measuring the impact of our adaptive learning strategy: we run the **AdaMine_avg**. The architecture and the losses are identical to our proposal, but instead of using the adaptive learning strategy, this one performs the stochastic gradient descent averaging the gradient over all triplets, as is common practice in the literature;
- Evaluating the impact of the text structure: we run our whole model (retrieval and semantic losses + adaptive SGD) by considering either ingredients only (noted **AdaMine_ingr**) or instructions only (noted **AdaMine_instr**).

²<http://pytorch.org>

³<https://github.com/Cadene/recipe1m.bootstrap.pytorch>

Scenarios	Strategies	Image to Textual recipe				Textual recipe to Image			
		MedR	R@1	R@5	R@10	MedR	R@1	R@5	R@10
AdaMine_ins	Retrieval loss	15.4	13.3	32.1	42.6	15.8	12.3	31.1	41.7
AdaMine_ins+cls	Retrieval loss + Classification loss	14.8	13.6	32.7	43.2	15.2	12.9	31.8	42.5
AdaMine	Retrieval loss + Semantic loss	13.2	14.9	35.3	45.2	12.2	14.8	34.6	46.1

Table 1: Impact of the semantic information. MedR means Median Rank (lower is better). R@K means Recall at K (between 0% and 100%, higher is better). The average value over 5 bags of 10,000 pairs each is reported.

4.4 Implementation details

Network learning. As adopted by [33], we use the Adam [23] optimizer with a learning rate of 10^{-4} . Besides, we propose a simpler training scheme: At the beginning of the training phase, we freeze the ResNet-50 weights, optimizing only the text-processing branch, as well as the weights of the mapping of the visual processing branch. After 20 epochs, the weights of the ResNet-50 are unfrozen and the whole architecture is fine-tuned for 60 more epochs. For the final model selection, we evaluate the MedR on the validation set at the end of each training epoch, and we keep the model with the best MedR on validation.

It is worth mentioning that in order to learn our model, a single NVidia Titan X Pascal is used, and the training phase lasts for 30 hours. We also improved the efficiency of the PWC baseline, initially implemented in Torch and requiring 3 days of learning using four NVidia Titan X Pascal to 30 hours on a single NVidia Titan X Pascal. We will release codes for both our model and the PWC* model.

Parameter choices. Our model **AdaMine** is a combination of the adaptive bidirectional instance and semantic triplet losses. Its margin α and the weight λ for the semantic cost \mathcal{L}_{sem} are determined using a cross-validation with values varying between 0.1 and 1, and step of 0.1. We finally retained 0.3 for both α and λ . The parameter λ further analyzed in Section 5.1 and in Figure 4.

Triplet sampling. As is common with triplet based losses in deep learning, we adopt a per-batch sampling strategy for estimating \mathcal{L}_{ins} and \mathcal{L}_{sem} (see subsection 3.3). The set of multi-modal (image-recipe) matching pairs in the train (resp. validation) set are split in 2383 (resp. 513) mini-batches of 100 pairs. Following the dataset structure in which half of the pairs are not labeled by class meta-data, those 100 pairs are split into: 1) 50 randomly selected pairs among those not associated with class information; 2) 50 labeled pairs for which we respect the distribution over all classes in the training set (resp. validation set).

Within each mini-batch, we then build the set of double-triplets fitting with our joint retrieval and semantic loss functions. Each item in the 100 pairs is iteratively seen as the query. The main issue is to build positive and negative sets with respect to this query. For the retrieval losses, the item in the other modality associated to the query is assigned to the positive set while the remaining items in the other modality (namely, 99 items) are assigned to the negative instance set. For the semantic loss, we randomly select, as the positive set, one item in the other modality that does not belong to the matching pair while sharing the query class. For the negative set, we consider the remaining items in the other modality that do not belong to the query class. For fair comparison between queries over the mini-batch, we limit the size of the negative sets over each query to the smallest negative ensemble size inside the batch.

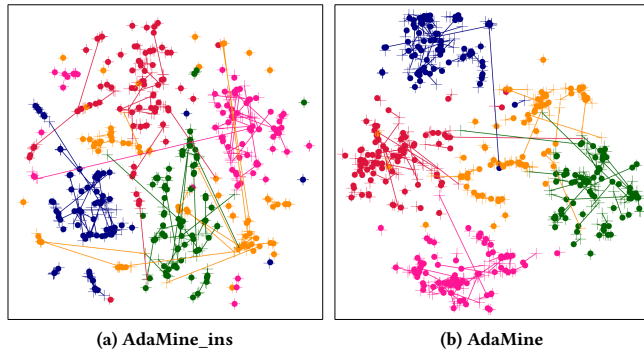


Figure 3: t-SNE visualization. Image (resp. Recipe) points are denoted with the + (resp. •) symbol. Matching pairs are connected with a trace. Blue points are associated to the cupcake class, orange to hamburger, pink to green beans, green to pork chops, and red to pizza.

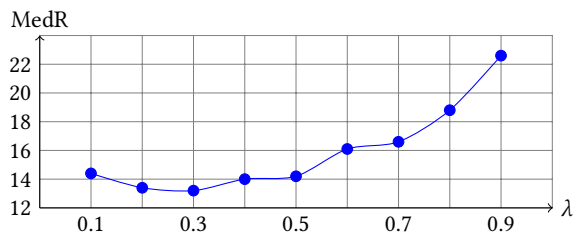


Figure 4: MedR scores for different values of λ , responsible for weighting the semantic regularization cost \mathcal{L}_{sem} of AdaMine, calculated over 5 bags of 10,000 validation samples.

5 EXPERIMENTS

5.1 Analysis of the semantic contribution

We analyze our main hypotheses related to the importance of semantic information for learning multi-modal embeddings (see Hypothesis H2 in 3.1). Specifically, in this part we test whether semantic information can help to better structure the latent space, taking into account class information and imposing structural coherence. Compared with [33] which adds an additional classification layer, we believe that directly injecting this semantic information with a global loss $\mathcal{L}(\theta)$ (Equation 1) comes as a more natural approach to integrating class-based meta-data (see Hypothesis H3 in 3.1).

To test this intuition, we start by quantifying, in Table 1, the impacts of the semantic information in the learning process. To do so, we evaluate the effectiveness of different scenarios of our model **AdaMine** with respect to the multi-modal retrieval task (image-to-text and text-to-image) in terms of *MedR* and Recall at ranks 1, 5, and 10. Compared with a retrieval loss alone (**AdaMine_ins**),





















Ingredient query	Cooking instruction query	Top 5 retrieved images					
Yogurt, cucumber, salt, garlic clove, fresh mint.	Stir yogurt until smooth. Add cucumber, salt, and garlic. Garnish with mint. Normally eaten with pita bread. Enjoy!						AM
Olive oil, balsamic vinegar, thyme, lemons, chicken drumsticks with bones and skin, garlic, potatoes, parsley.	Whisk together oil, mustard, vinegar, and herbs. Season to taste with a bit of salt and pepper and a large pinch or two of brown sugar. Place chicken in a non-metal dish and pour marinade on top to coat. [...]						AM
Pizza dough, hummus, arugula, cherry or grape tomatoes, pitted greek olives, feta cheese.	Cut the dough into two 8-ounce sized pieces. Roll the ends under to create round balls. Then using a well-floured rolling pin, roll the dough out into 12-inch circles. [...]						AM
Unsalted butter, eggs, condensed milk, sugar, vanilla extract, chopped pecans, chocolate chips, butterscotch chips, [...]	Preheat the oven to 375 degrees F. In a large bowl, whisk together the melted butter and eggs until combined. Whisk in the sweetened condensed milk, sugar, vanilla, pecans, chocolate chips, butterscotch chips, [...]						AM

Table 2: Recipe-to-images visualization. For each recipe, we have the top row, indicating the top 5 images retrieved by our AdaMine model for a given recipe query, and the bottom row, indicating the top 5 images by the triplet loss for the same recipe. In green, the matching image. In blue, images belonging to the same class than the recipe. In red, images belonging to a different class. AM indicates AdaMine, and AM_ins AdaMine_ins.

		Image to Textual recipe				Textual recipe to Image				
		MedR	R@1	R@5	R@10	MedR	R@1	R@5	R@10	
1k items	SOTA	Random	499	0.0	0.0	0.0	499	0.0	0.0	0.0
		CCA [33]	15.7	14.0	32.0	43.0	24.8	9.0	24.0	35.0
		PWC [33]	5.2	24.0	51.0	65.0	5.1	25.0	52.0	65.0
		PWC [33]*	5.0 ± 0.4	22.8 ± 1.4	47.7 ± 1.4	60.1 ± 1.4	5.3 ± 0.4	21.2 ± 1.2	48.0 ± 1.1	60.4 ± 1.4
		PWC++	3.3 ± 0.4	25.8 ± 1.6	54.5 ± 1.3	67.1 ± 1.4	3.5 ± 0.5	24.8 ± 1.1	55.0 ± 1.8	67.1 ± 1.2
	Model scenarios	AdaMine_sem	21.1 ± 2.0	8.7 ± 0.7	25.5 ± 0.9	36.5 ± 0.9	21.1 ± 1.9	8.2 ± 0.9	25.5 ± 1.0	36.2 ± 0.9
		AdaMine_ins	1.5 ± 0.5	37.5 ± 1.1	67.0 ± 1.3	76.8 ± 1.5	1.6 ± 0.5	36.1 ± 1.6	66.6 ± 1.3	76.8 ± 1.5
		AdaMine_ins+cls	1.1 ± 0.3	38.3 ± 1.6	67.5 ± 1.2	78.0 ± 0.9	1.2 ± 0.4	37.5 ± 1.4	67.7 ± 1.2	77.3 ± 1.0
		AdaMine_avg	2.3 ± 0.5	30.6 ± 1.1	60.3 ± 1.2	71.4 ± 1.3	2.2 ± 0.3	30.6 ± 1.8	60.6 ± 1.1	71.9 ± 1.1
		AdaMine_ingr	4.9 ± 0.5	22.6 ± 1.4	48.5 ± 1.6	59.8 ± 1.3	5.0 ± 0.6	21.5 ± 1.4	47.7 ± 2.1	59.8 ± 1.8
AdaMine_instr	3.9 ± 0.5	24.4 ± 1.6	52.6 ± 2.0	65.4 ± 1.6	3.7 ± 0.5	23.6 ± 1.7	52.7 ± 1.6	65.5 ± 1.5		
AdaMine	1.0 ± 0.1	39.8 ± 1.8	69.0 ± 1.8	77.4 ± 1.1	1.0 ± 0.1	40.2 ± 1.6	68.1 ± 1.2	78.7 ± 1.3		
10k items	PWC++ (best SOTA)	34.6 ± 1.0	7.6 ± 0.2	19.8 ± 0.1	30.3 ± 0.4	35.0 ± 0.9	6.8 ± 0.2	21.5 ± 0.2	28.8 ± 0.3	
	AdaMine_sem	207.3 ± 3.9	1.4 ± 0.3	5.7 ± 0.3	9.6 ± 0.3	205.4 ± 3.2	1.4 ± 0.1	5.4 ± 0.2	9.1 ± 0.4	
	AdaMine_ins	15.4 ± 0.5	13.3 ± 0.2	32.1 ± 0.7	42.6 ± 0.8	15.8 ± 0.7	12.3 ± 0.3	31.1 ± 0.5	41.7 ± 0.6	
	AdaMine_ins+cls	14.8 ± 0.4	13.6 ± 0.2	32.7 ± 0.4	43.2 ± 0.3	15.2 ± 0.4	12.9 ± 0.3	31.8 ± 0.3	42.5 ± 0.2	
	AdaMine_avg	24.6 ± 0.8	10.0 ± 0.2	25.9 ± 0.4	35.7 ± 0.5	24.0 ± 0.6	9.2 ± 0.4	25.4 ± 0.5	35.3 ± 0.4	
	AdaMine_ingr	52.8 ± 1.2	6.5 ± 0.2	17.9 ± 0.2	25.8 ± 0.3	53.8 ± 0.7	5.8 ± 0.3	17.3 ± 0.2	25.0 ± 0.2	
	AdaMine_instr	39.0 ± 0.9	6.4 ± 0.1	18.9 ± 0.4	27.6 ± 0.5	39.2 ± 0.7	5.7 ± 0.4	17.9 ± 0.6	26.6 ± 0.5	
AdaMine	13.2 ± 0.4	14.9 ± 0.3	35.3 ± 0.2	45.2 ± 0.2	12.2 ± 0.4	14.8 ± 0.3	34.6 ± 0.3	46.1 ± 0.3		

Table 3: State-of-the-art comparison. MedR means Median Rank (lower is better). R@K means Recall at K (between 0% and 100%, higher is better). The mean and std values over 10 (resp. 5) bags of 1k (resp. 10k) pairs each are reported for the top (resp. bottom) table. Items marked with a star (*) are our reimplementations of the cited methods.

we point out that adding semantic information with a classification cost **AdaMine_ins+cls** or a semantic loss **AdaMine** improves the results. When evaluating with 10,000 pairs (10k setting), while **AdaMine_ins** obtains MedRs 15.4 and 15.8, the semantic models (**AdaMine_ins+cls** and **AdaMine**) lower these values to 14.8 and 15.2, and 13.2 and 12.2, respectively (lower is better) for both retrieval tasks (image-to-text and text-to-image).

The importance of semantic information becomes clearer when we directly compare the impact of adding the semantic loss to the base model (**AdaMine** vs **AdaMine_ins**), since the former obtains the best results for every metric. To better understand this phenomenon, we depict in Figure 3 item embeddings obtained by the **AdaMine_ins** and **AdaMine** models using a t-SNE visualization. This figure is generated by selecting 400 matching recipe-image pairs (800 data points), which are randomly selected from, and equally distributed among 5 of the most occurring classes of the Recipe1M dataset. Each item is colored according to its category (e.g., blue points for the cupcake class), and items of the same instance are connected with a trace. Therefore, Figure 3 allows drawing two conclusions: 1) our model—on the right side of the figure—is able to structure the latent space while keeping items of the same class close to each other (see color clusters); 2) our model reduces the sum of distances between pairs of instances (in the figure, connected with traces), thus reducing the MedR and increasing the recall. We also illustrate this comparison through qualitative examples. In Table 2, **AdaMine** (top row) and **AdaMine_ins** (bottom row) are compared on four queries, for which both models are able to rank the correct match in the top-5 among 10,000 candidates. For the first and second queries (cucumber salad and roasted chicken, respectively), both models are able to retrieve the matching image in the first position. However, the rest of the top images retrieved by our model are semantically related to the query, by sharing critical ingredients (cucumber, chicken) of the recipe. In the third and fourth queries (pizza and chocolate chip, respectively), our model is able to rank both the matching image and semantically connected samples in a more coherent way, due to a better alignment of the retrieval space produced by the semantic modeling. These results reinforce our intuition that it is necessary to integrate semantic information in addition to item pairwise anchors while learning multi-modal embeddings.

Second, we evaluate our intuition that classification is under-effective for integrating the semantics within the latent space (see Hypothesis H3 in 3.1). Table 1 shows that our semantic loss **AdaMine**, proposed in subsection 3.2.3, outperforms our model scenario **AdaMine_ins+cls** which relies on a classification head as proposed in [33]. For instance, we obtain an improvement of +9.57% in terms of $R@1$ with respect to the classification loss setting **AdaMine_ins+cls**. This result suggests that our semantic loss is more appropriate to organize the latent space so as to retrieve text-image matching pairs. It becomes important, then, to understand the impacts of the weighting factor λ between the two losses \mathcal{L}_{ins} and \mathcal{L}_{sem} (Equation 1). In Figure 4, we observe a fair level of robustness for lower values of λ , but any value over 0.5 has a hindering effect on the retrieval task, since the semantic grouping starts to be of considerable importance. These experiments confirm the

importance of additional semantic clues: despite having one million less parameters than [33]’s proposal, our approach still achieves better scores, when compared to the addition of the classification head.

5.2 Testing the effectiveness of the model

In the following, we evaluate the effectiveness of our model, compared to different baseline models. Results are presented in Table 3 for both image-to-text and text-to-image retrieval tasks. We report results on the 1K setup and test the robustness of our model on the 10k setup by reporting only the best state-of-the-art (SOTA) baseline for comparison. From a general point of view, we observe that our model **AdaMine** overpasses the different baselines and model scenarios. Small values of standard deviation outlines the low variability of experimented models, and accordingly the robustness of obtained results. For instance, our model reaches a value equal to 1 for the Median Rank metric (MedR) for the 1k setting and both retrieval tasks while the well-known SOTA models CCA and PWC++ obtain respectively 15.7 and 3.3. Contrary to PWC, all of our model scenarios, denoted **AdaMine_***, adopt the triplet loss. Ablation tests on our proposals show their effectiveness. This trend is noticed over all retrieval tasks and all metrics. The comparison of the results obtained over 1k and 10k settings outlines the same statement with larger improvements (with similar standard deviation) for our model **AdaMine** with respect to SOTA models and **AdaMine**-based scenarios. More particularly, we first begin our discussion with the comparison with respect to SOTA models and outline the following statements:

- Global alignment models (baseline **CCA**) are less effective than advanced models (**PWC**, **PWC++**, and **AdaMine**). Indeed, the **CCA** model obtains a MedR value of 15.7 for the image-to-text retrieval task (1k setting) while the metric range of advanced models is between 1 and 5.2. This suggests the effectiveness of taking into account dissimilar pairs during the learning process.
- We observe that our triplet based model **AdaMine** consistently outperforms pairwise methods (**PWC** and **PWC++**). For instance, our model obtains a significant decrease of -61.84% in terms of MedR with respect to **PWC++** for the 10k setting and the image-to-text retrieval task. This suggests that relative cosine distances are better at structuring the latent space than absolute cosine distances.
- Our model **AdaMine** surpasses the current state-of-the-art results by a large margin. For the 1k setup, it reduces the medR score by a factor of 5—from 5.2 and 5.1 to 1.0 and 1.0—, and by a factor bigger than 3 for the 10k setup. One strength of our model is that it has fewer parameters than **PWC++** and **PWC**, since the feature space is directly optimized with a semantic loss, without the addition a parameter-heavy head to the model.

Second, the comparison according to different versions of our model outlines three main statements:

- The analysis of **AdaMine_ins**, **AdaMine_ins+cls**, and **AdaMine** corroborates the results observed in Section 5.1 dealing with the impact of the semantic loss on the performance of the model. In the 1k setting, the instance-based approach (**AdaMine_ins**) achieves a MedRs value equal of 1.5 and 1.6 for both tasks (lower is better), while the addition of a classification head (**AdaMine_ins+cls**), proposed by [33], improves these results



Table 4: Ingredient-to-Image. Examples of images in the top 20 results when searching for an ingredient within the class *Pizza*.

Ingredients and Instructions Query	Top 4 retrieved images				
<p><i>Oregano, Zucchini, Tofu, Bell pepper, Onions, Broccoli, Olive Oil</i></p> <ol style="list-style-type: none"> 1. Cut all ingredients into small pieces. 2. Put broccoli in hot water for 10 min 3. Heat olive oil in pan and put oregano in it. 4. Put cottage cheese and saute for 1 minute. 5. Put onion, bell pepper, broccoli, zucchini. 6. Put burnt chilli garlic dressing with salt. 7. Saute for 1 minutes. 					with broccoli
					without broccoli

Table 5: Removing ingredient task. Top 4 retrieved images with (top row) and without (bottom row) broccoli in the ingredient and instruction lists w.r.t the original recipe (Tofu Sauté).

to 1.1 and 1.2. Removing the classification head and adding a semantic loss (**AdaMine**) further improves the results to 1 for both retrieval tasks which further validates Hypothesis H3 in 3.1.

- The adaptive sampling strategy described in subsection 3.3 strongly contributes to the good results of **AdaMine**. With **AdaMine_avg**, we test the same setup of **AdaMine**, replacing the adaptive strategy with the average one. The importance of removing triplets that are not contributing to the loss becomes evident when the scores for both strategies are compared: 24.6 and 24.0 of MedR (lower is better) for **AdaMine_avg**, and 13.2 and 12.2 for **AdaMine**, an improvement of roughly 46.34% and 49.17%.

- AdaMine** combines the information coming from the image and all the parts of the recipe (instructions and ingredients), attaining high scores. When compared to the degraded models **AdaMine_ingr** and **AdaMine_instr**, we conclude that both textual information are complementary and necessary for correctly identifying the recipe of a plate. While **AdaMine** achieves MedRs of 13.2 and 12.2 (lower is better), the scenarios without instructions or without ingredients achieve 52.8 and 53.8, and 39.0 and 39.2, respectively.

5.3 Qualitative studies on downstream tasks

In what follows, we discuss the potential of our model for promising cooking-related application tasks. We particularly focus on downstream tasks in which the current setting might be applied. We provide illustrative examples issued from the testing set of our

evaluation process. For better readability, we always show the results as images, even for text recipes for which we display their corresponding original picture.

Ingredient To Image. An interesting ability of our model is to map ingredients into the latent space. One example of task is to retrieve recipes containing specific ingredients that could be visually identified. This is particularly useful when one would like to know what they can cook using aliments available in their fridge. To demonstrate this process, we create each recipe query as follows: 1) for the ingredients part, we use a single word which corresponds to the ingredient we want to retrieve; 2) for the instructions part, we use the average of the instruction embeddings over all the training set. Then, we project our query into the multi-modal space and retrieve the nearest neighbors among 10,000 images randomly picked from the testing set. We show on Table 4 examples of retrieved images when searching for different ingredients while constraining the results to the class *pizza*. Searching for *pineapple* or *olives* results in different types of pizzas. An interesting remark is that searching for *strawberries* inside the class *pizza* yields images of *fruit pizza* containing strawberries, *i.e.*, images that are visually similar to pizzas while containing the required ingredient. This shows the fine-grain structure of the latent space in which recipes and images are organized by visual or semantic similarity inside the different classes.

Removing ingredients. The capacity of finely model the presence or absence of specific ingredients may be interesting for generating

menus, specially for users with dietary restrictions (for instance, peanut or lactose intolerance, or vegetarians and vegans). To do so, we randomly select a recipe having *broccoli* in its ingredients list (Table 5, first column) and retrieve the top 4 closest images in the embedding space from 1000 recipe images (Table 5, top row). Then we remove the *broccoli* in the ingredients and remove the instructions having the *broccoli* word. Finally, we retrieve once again the top 4 images associated to this "modified" recipe (Table 5, bottom row). The retrieved images using the original recipe have broccoli, whereas the retrieved images using the modified recipe do not have broccoli. This reinforces our previous statement, highlighting the ability of our latent space to correctly discriminate items with respect to ingredients.

6 CONCLUSION

In this paper, we introduce the AdaMine approach for learning crossmodal embeddings in the context of a large-scale cooking oriented retrieval task (image to recipe, and *vice versa*). Our main contribution relies on a joint retrieval and classification learning framework in which semantic information is directly injected in the cross-modal metric learning. This allows refining the multi-modal latent space by limiting the number of parameters to be learned. For learning our double-triplet learning scheme, we propose an adaptive strategy for informative triplet mining. AdaMine is evaluated on the very large scale and challenging Recipe1M crossmodal dataset, outperforming the state-of-the-art models. We also outline the benefit of incorporating semantic information and show the quality of the learned latent space with respect to downstream tasks. We are convinced that such very large scale multimodal deep embeddings frameworks offer new opportunities to explore joint combinations of Vision and Language understanding. Indeed, we plan in future work to extend our model by considering hierarchical levels within object semantics to better refine the structure of the latent space.

ACKNOWLEDGMENT

This work was partially supported by CNPq – Brazilian’s National Council for Scientific and Technological Development – and by Labex SMART, supported by French state funds managed by the ANR within the Investissements d’Avenir program under reference ANR-11-LABX-65.

REFERENCES

- [1] 2017. *CEA2017: Proceedings of the 9th Workshop on Multimedia for Cooking and Eating Activities in Conjunction with The 2017 International Joint Conference on Artificial Intelligence*.
- [2] Giuseppe Amato, Paolo Bolettieri, Vinicius Monteiro de Lira, Cristina Ioana Muntean, Raffaele Perego, and Chiara Renso. 2017. Social Media Image Recognition for Food Trend Analysis. In *SIGIR*. 1333–1336.
- [3] Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. 2013. Deep canonical correlation analysis. In *ICML*. 1247–1255.
- [4] Francis R Bach and Michael I Jordan. 2002. Kernel independent component analysis. *Journal of machine learning research* 3, Jul (2002), 1–48.
- [5] O. Beijbom, N. Joshi, D. Morris, S. Saponas, and S. Khullar. 2015. Menu-Match: Restaurant-Specific Food Logging from Images. In *2015 IEEE Winter Conference on Applications of Computer Vision*. 844–851.
- [6] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. 2014. Food-101 – Mining Discriminative Components with Random Forests. In *ECCV*.
- [7] Jingjing Chen and Chong-Wah Ngo. 2016. Deep-based ingredient recognition for cooking recipe retrieval. In *Proceedings of the 2016 ACM on Multimedia Conference*.
- [8] Jingjing Chen, Lei Pang, and Chong-Wah Ngo. 2017. Cross-Modal Recipe Retrieval: How to Cook this Dish?. In *MultiMedia Modeling*. 588–600.
- [9] M. Chen, K. Dhingra, W. Wu, L. Yang, R. Sukthankar, and J. Yang. 2009. PFID: Pittsburgh fast-food image dataset. In *ICIP*. 289–292.
- [10] David Elsweiler, Christoph Trattner, and Morgan Harvey. 2017. Exploiting Food Choice Biases for Healthier Recipe Recommendation. In *SIGIR*. 575–584.
- [11] Giovanni Maria Farinella, Dario Allegra, and Filippo Stanco. 2015. *A Benchmark Dataset to Study the Representation of Food Images*. 584–599.
- [12] R. Hadsell, S. Chopra, and Y. LeCun. 2006. Dimensionality Reduction by Learning an Invariant Mapping. In *CVP*. 1735–1742.
- [13] Jun Harashima, Yuichiro Someya, and Yohei Kikuta. 2017. Cookpad Image Dataset: An Image Collection As Infrastructure for Food Research. In *SIGIR*. 1229–1232.
- [14] Zellig Harris. 1954. Distributional structure. *Word* 10, 23 (1954), 146–162.
- [15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. *arXiv arXiv:1512.03385* (2015).
- [16] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Comput.* 9, 8 (1997), 1735–1780.
- [17] Harold Hotelling. 1936. Relations between two sets of variates. *Biometrika* 28, 3/4 (1936), 321–377.
- [18] J. Hu, J. Lu, and Y. P. Tan. 2014. Discriminative Deep Metric Learning for Face Verification in the Wild. In *CVPR*. 1875–1882.
- [19] J. Jeon, V. Lavrenko, and R. Manmatha. 2003. Automatic Image Annotation and Retrieval Using Cross-media Relevance Models. In *SIGIR*. 119–126.
- [20] Andrej Karpathy and Li Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *CVPR*. 3128–3137.
- [21] Yoshiyuki Kawano and Keiji Yanai. 2014. Food image recognition with deep convolutional features. In *UbiComp '14*. 589–593.
- [22] Yoshiyuki Kawano and Keiji Yanai. 2014. FoodCam: A Real-Time Mobile Food Recognition System Employing Fisher Vector. In *MMML*. 369–373.
- [23] Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv arXiv:1412.6980* (2014).
- [24] Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. 2015. Unifying visual-semantic embeddings with multimodal neural language models. *TACL* (2015).
- [25] Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-Thought Vectors. In *NIPS*. 3294–3302.
- [26] Tomasz Kusmierczyk and Kjetil Nørvåg. 2016. Online Food Recipe Title Semantics: Combining Nutrient Facts and Topics. In *CIKM*. 2013–2016.
- [27] Tomasz Kusmierczyk, Christoph Trattner, and Kjetil Nørvåg. 2016. Understanding and predicting online food recipe production patterns. In *HT*. 243–248.
- [28] Pei Ling Lai and Colin Fyfe. 2000. Kernel and nonlinear canonical correlation analysis. *International Journal of Neural Systems* 10, 05 (2000), 365–377.
- [29] Marc T Law, Nicolas Thome, and Matthieu Cord. 2013. Quadruplet-wise image similarity learning. In *ICCV*. 249–256.
- [30] Angeliki Lazaridou, Nghia The Pham, and Marco Baroni. 2015. Combining Language and Vision with a Multimodal Skip-gram Model. In *NAACL HLT*. 153–163.
- [31] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*. 3111–3119.
- [32] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. ImageNet Large Scale Visual Recognition Challenge. *IJCV* 115, 3 (2015), 211–252.
- [33] Amaia Salvador, Nicholas Hynes, Yusuf Aytar, Javier Marin, Ferda Ofli, Ingmar Weber, and Antonio Torralba. 2017. Learning Cross-modal Embeddings for Cooking Recipes and Food Images. In *CVPR*.
- [34] Satoshi Sanjo and Marie Katsurai. 2017. Recipe Popularity Prediction with Deep Visual-Semantic Fusion. In *CIKM*. 2279–2282.
- [35] Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 815–823.
- [36] Aixun Sun, Sourav S. Bhowmick, Khanh Tran Nam Nguyen, and Ge Bai. 2011. Tag-based Social Image Retrieval: An Empirical Evaluation. *J. Am. Soc. Inf. Sci. Technol.* 62, 12 (2011), 2364–2381.
- [37] Christoph Trattner and David Elsweiler. 2017. Investigating the Healthiness of Internet-Sourced Recipes: Implications for Meal Planning and Recommender Systems. In *WWW*. 489–498.
- [38] Xin Wang, D. Kumar, N. Thome, M. Cord, and F. Precioso. 2015. Recipe recognition with large multimodal food dataset. In *ICMEW*. 1–6.
- [39] Kilian Q. Weinberger and Lawrence K. Saul. 2009. Distance Metric Learning for Large Margin Nearest Neighbor Classification. *J. Mach. Learn. Res.* 10 (2009), 207–244.
- [40] Jianlong Wu, Zhouchen Lin, and Hongbin Zha. 2017. Joint Latent Subspace Learning and Regression for Cross-Modal Retrieval. In *SIGIR*. 917–920.
- [41] Eric P. Xing, Michael I. Jordan, Stuart J Russell, and Andrew Y. Ng. 2003. Distance Metric Learning with Application to Clustering with Side-Information. In *NIPS*. 521–528.
- [42] Fei Yan and Krystian Mikolajczyk. 2015. Deep correlation for matching images and text. In *CVPR*. 3441–3450.