CS224N Final Project
Abstract Meta-Concept Features for Text-Illustration

Ines Chami[*]
Institute for Computational and Mathematical Engineering
Stanford University
Stanford, CA 94305
chami@stanford.edu

Abstract

Cross language-image retrieval is a problem of high interest that is at the frontier between computer vision and natural language processing. State-of-the-art methods learn a common space with regard to some constraints of correlation or similarity from two textual and visual modalities that are processed in parallel and possibly jointly. This paper proposes a different approach that considers the cross-modal problem as a supervised mapping of visual modalities to textual ones. Each modality is thus seen as a particular projection of an abstract meta-concept. In practice, this space is learned through an asymmetric process, where the textual modality is used to generate a multi-label representation, further used to map the visual modality through a simple multi-layer perceptron. While being quite easy to implement, the experiments show that our approach significantly outperforms the state-of-the-art on FlickR-8K and FlickR-30K datasets for the text-illustration task.

1 Introduction

Many works deal with multi-modal tasks, either to retrieve an image given a text query (text illustration) or to linguistically describe an image (image captioning) or to classify bi-modal documents. Most of these approaches aim at learning a joint embedding for both modalities into a common latent space, in which vectors from the two different modalities are directly comparable [9, 13, 12, 6, 23].

Two families of approaches emerge when reviewing the literature about the design of such a common latent space. The first, specifically focuses on learning the latent space from *existing* textual and visual features. These last, typically result from an embedding representation, such as the word2vec [17] features for textual content and one layer from a pre-trained Convolutional Neural Network (CNN) [3, 19] for the visual modality. Then, the latent space is learned according to a certain principle from aligned textual and visual data described with these features. By “aligned data”, one must understand that an image is for example aligned with its caption, in the sense that their respective contents are supposed to match. Regarding the principle used to learn the latent space, the seminal work of Hardoon [9] consisted in maximizing the correlation of the aligned data once projected in the common space.

Alternative approaches rely on deep networks to model a full multi-modal embedding. This is the case of [12] who proposed to infer the correspondences between images and their sentence description. First, they use a Region Convolutional Neural Network (RCNN) [7] as image representation and a Bidirectional Recurrent Neural Network (BRNN) [18] to textual one. Second, they define a loss that encourages aligned image-sentences pairs to have a higher score than misaligned pairs.

[*] This research has been conducted jointly with Youssef Tamaazousti and supervised by Hervé Le Borgne

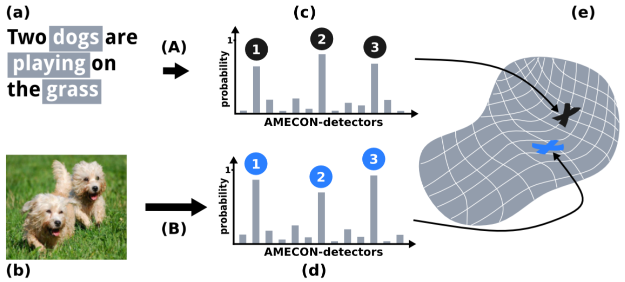

Figure 1: Given input text (a) and an input image (b), our method computes the AMECON-features for each modality (textual (c) and visual (d)) and matches them in the AMECON Space (e). The key novelty of our approach is that, each dimension of our AMECON representation (each *bar* in (c) and (d)) corresponds to the output probability of an *abstract meta-concept* detector applied on the input data. For instance here, the two input modalities are close together in the AMECON space (e) since they both have the *same* three abstract meta-concepts that are highly activated. The arrows (A) and (B) highlight the *asymmetry* of our approach. In fact, it shows that more computations are needed to project the visual features than the textual ones. Best view in color.

Our initial motivation is thus the same as previous work, in particular that of [6] that aimed at matching the visual representation to the textual one. However, our model differs from the previous work by being much more *asymmetric*. Indeed, most of previous works use two parallel pipelines, one for each modality that only differ by the features considered, then determine a method to design a common space. We adopt a different point of view, considering that the visual and the textual modalities should not (yet) be processed symmetrically. It refers to the well-known semantic gap [20] that reflects the fact that textual features are closer to human understanding than the pixel-based features. Despite the progresses due to deep learning in visual recognition, we argue that gap is still relevant to consider.

We thus propose to consider the Abstract MEta-CONcept (AMECON) principle for a multi-modal (texts and images) alignment. A *semantic concept* can be named by a word from the vocabulary of a given language. In line with [1], a *meta-concept* is defined by a concept subsuming several semantic concepts. Last, a concept can be qualified as *abstract* when it does not reflect a notion that is explicit in a given language. For example, it is sometimes handy to use some words from a foreign language when it does not really exist in ours. However, in the case of AMECONs, they can be even more abstract. A key particularity of our approach is that, the mapping from one modality to this space is asymmetric, *i.e* specific to each modality. Simply said, the proposed AMECON space is much closer to the textual embedding space than to the visual one. In practice, on the one hand, the *textual modality* is mapped through vector quantization of the textual features, because we consider they are close enough from the human conceptual space. On the other hand, we learn a mapping from the *visual modality* to the AMECON space with a multi-layer perceptron. It takes the visual features as input and the target (labels) are derived from the textual features by local hard coding. An overview of our approach is illustrated in Figure 1.

Our proposal is thus a new method to build a multi-modal common latent space. It particularly, matches visual content to sentences and thus aims to perform cross-modal or bi-modal tasks. In this paper, we focus on the text-illustration (*i.e*, retrieve an image from a textual query). Due to the asymmetry of our approach, we consider the inverse problem (*i.e*, retrieve a text from a visual query) as outside of the scope of this paper. While being quite easy to implement, our method exhibits performance above the current state-of-the-art on text-illustration. We conducted experiments (in Sec. 4) on two publicly available benchmarks, namely FlickR-8k and FlickR-30k, on which our method significantly outperforms the previous works, using a comparable protocol of performance measure. We also conducted (in Appendix A) an in-depth analysis of the proposed model to highlight its insights, including an ablation study that shows the relative importance of each component.

2 Related Work

In its seminal work on the design of common space to visual and textual data, Hardoon proposed to maximize the correlation between the projections of both modalities using the Canonical Correlation Analysis (CCA) and its kernelized version (KCCA) [9]. This work has then be extended by [8] who added a third view that reflects the "semantic classes" derived from the ground-truth or the keywords used to download the images. This work also proposed to derive this third view from unsupervised clustering of the tags to avoid the use of ground truth. While being very different from our approach since it relies on a symmetric projection of both modalities through KCCA, such a clustering of tags relates to the process we use to define the projection of the textual features on the AMECON space. However, while [8] uses the clusters to define a third view that is further projected on the KCCA space, our approach uses it as a codebook to directly encode the textual projection.

In the vein of reflecting semantics, [4] proposed to build semantic features into the common space, that is to say to create a signature where each dimension is a given semantic concept that is estimated by a learned binary classifier [22]. Contrary to ours, these concepts are neither meta nor abstract. However, one could image to apply the approach of [1] to get meta-concepts in the common space. Still, a major difference with our work is that each concept is obtained by supervised classification, while in our case, the *abstract* concepts deeply result from an unsupervised approach.

Rather than relying on an priori principle such as maximizing the correlation, other works consider deep neural networks with other type of constraint. As already cited in the introduction, Frome *et al.* proposed DeViSE [6] that learns a similarity metric between the top layer of a visual network and a skip-gram text model (*word2vec*), optimizing an objective function that forces the similarity of a given image to the relevant label to be higher than that to other randomly chosen text terms. This is probably the work that is is closer to ours, in the sense that it tries to directly match the visual representation to the textual one. However, our work differs on several points. Indeed, our approach transforms explicitly the textual information into labels to use a supervised classification scheme to map the visual representation. Thereby, the advantage of our approach is to design a non-linear mapping between both modalities while DeViSE only proposes a linear transformation between the original features. In [13] and [12], visual data is also aligned with sentences, thanks to a structured loss that forces aligned sentences and images to be close reciprocal neighbours. The main difference between our approach and these deep learning-based approaches is that they rely on a quite *symmetric* scheme where both modalities are processed similarly. While our *asymmetric* approach seems more straightforward, it remains conceptually simpler and has much better performances. Also interesting, the asymmetry of our approach limits the performances on the inverse cross-modal task thus we only evaluate it on text-illustration. In fact, for these *asymmetrical* approaches, getting good performances on one direction of cross-modal task when building the common latent space on the inverse direction, remains an open problem.

3 Proposed Approach

Our approach to Text-Image matching is named "Abstract Meta-Concept" (**AMECON**). It consists of matching texts and images in a common latent space (the AMECON Space, described in Sec. 3.1) where the cues (visual, textual or both) contribute to activate the different *abstract meta-concept* detectors. In Sec. 3.2, we describe how to learn the abstract meta-concepts and how to generate AMECON features for the text modality. Sec. 3.3 details the learning of AMECON features for the visual modality.

3.1 AMECON Space

Let first recall that a *semantic-concept* is *any* word (associated to a particular *notion*) from the real-world vocabulary used by humans (*e.g*, bicycle plant, bird, etc.).

Definition 1. An *abstract meta-concept* is both, an *abstract* concept and a *meta*-concept. An abstract concept describes a concept that is not associated to a semantic connotation (that does not exist in the real-world vocabulary used by humans) and a meta-concept is a concept that subsumes

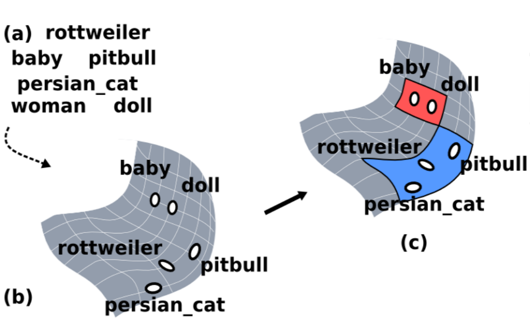

Figure 2: Illustration of the proposed AMECON principle. Given a set of words from a training corpus (a) and their projections in a words embedding space (b), our method clusters the space (c) such that each cluster is an *abstract meta-concept* that corresponds to an *abstract* concept (do not exist in the real-world) and a *general* concept (group of concepts). For instance, the blue cluster in (c) is *general* since it subsumes the vectors of many words and is *abstract* since no semantic connotation can be attributed to it. Best view in color.

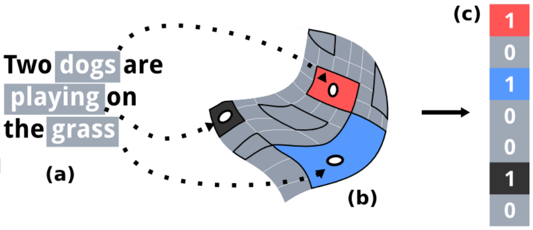

Figure 3: Illustration of the proposed textual AMECON-features. Given an input caption (a), our method first selects the non stop-words (coloured in gray), computes their mid-level features and projects them in the clustered word embedding (b) that corresponds to our AMECON space. After the projection, when a word embedding representation falls in an abstract meta-concept (*e.g*, blue cluster), its associated dimension is activated (*e.g*, 3^{rd} dimension). All other dimensions are filled with a zero-value. Applying this process on all the selected words and pooling their *binary* textual AMECON features together, results in a *binary* multi-label representation (c). Best view in color.

others (at least one). Note that, the subsumed concepts can be either semantic or abstract.

Definition 2. An *abstract meta-concept detector* ($\phi_i(\mathbf{x})$) is a visual or textual classifier that takes as input a mid-level representation (visual $\mathbf{x}^V$ or textual $\mathbf{x}^T$) of an input data and an AMECON-model (that has been learned with positive and negative samples of that abstract meta-concept) and returns the probability of presence of that abstract meta-concept given the input data.

Let us consider a visual representation of an image I noted χ^V and a linguistic representation of a text $\mathcal{T}$ noted χ^T, such that each dimension χ_i^V or χ_i^T reflects the *same* abstract meta-concept. Their integration into a unique multi-modal description χ results from a scheme where each representation (visual or textual) is an imperfect representation of the corresponding abstract meta-concept. Therefore, we name "AMECON Space" the space containing these abstract meta-concept and illustrate the principle in Figure 1. For cross-modal retrieval where we need to retrieve the nearest documents from another modality, we can simply use the k-Nearest Neighbours (k-NN) algorithm since both modalities are represented in the *same* AMECON space.

3.2 Textual AMECON-Features

3.2.1 Learning the AMECONs

We propose to learn the abstract meta-concepts (AMECONs) using unsupervised clustering. Hence, the AMECON principle verifies its two definitional characteristics: (i) it groups similar data into a generic cluster that thus corresponds to a *meta*-concept and (ii) thanks to the unsupervised aspect, the resulting clusters do not have any explicit semantic connotation (*i.e* do not exist in the real-world vocabulary of humans) making them *abstract*-concepts. More generally, AMECONs are obtained through unsupervised clustering of textual mid-level features (*e.g.*, word2vec [17]). In that sense, our method adopts a "bottom-up" approach, generating high-level knowledge from low-level data in the same vein as [1] for the *meta* aspect. We illustrate the AMECON principle in Figure 2.

To learn the AMECONs in practice, we collect all the words of a training corpus and represent them in an embedding space of dimension d. Then, we group the word vector representations using a clustering algorithm (*e.g* K-means) that results into C clusters (C being chosen arbitrarily or obtained through cross-validation). Each cluster is an AMECON that is represented by the corresponding cluster center $(\mathbf{c_i})_{i=1,\dots,C} \in \mathbb{R}^d$. Within an AMECON cluster, words have *similar* semantic connotations. Note that, the number of AMECONs (C) directly corresponds to the dimensionality of the AMECON Space presented in the previous section.

3.2.2 Learning the Textual AMECON-Features

The set of C abstract meta-concepts is now seen as a codebook that we use to encode any piece of information. In the case of textual information, we adopt a coding scheme similar to local soft coding [15], originally introduced as locality-constrained linear coding [24], that is nevertheless binarized. Given a caption $\mathcal{T}$ composed of n words, we compute the mid-level representation $\mathbf{x}^{T,j} \in \mathbb{R}^d$ of each word, resulting into a set of n vectors $\{\mathbf{x}^{T,j}\}_{j=1,\dots,n}$ in the word embedding space. The j^{th} word is then encoded according to the codebook in the C-dimensional vector:

$$\chi_{bin}^{T,j} = \sum_{i=1}^{C} \mathbb{1}_{NN^m(\mathbf{x}^{T,j})}(\mathbf{c_i})\mathbf{e_i}, \tag{1}$$

where $(\mathbf{e_1}, \dots, \mathbf{e_C})$ is the standard canonical basis of $\mathbb{R}^C$, $NN^m(\mathbf{x}^{T,j})$ is the set of the m nearest AMECON clusters of $\mathbf{x}^{T,j}$ in the word embedding space and $\mathbb{1}_S$ is the indicator function for the set S. It is thus a "local hard coding" of $\chi^{T,j}$ according to the codebook. We add the index notation $\cdot_{bin}$ to highlight it is a binary vector. The parameter m can be set arbitrarily or determined by cross-validation. To compute a unique caption's representation χ^T in this C-dimensional space, we pool all the word's representations. Formally, we obtain χ^T through:

$$\chi_{bin}^{T} = \underset{j=1\dots n}{\mathcal{P}} (\chi_{bin}^{T,j}), \tag{2}$$

where $\mathcal{P}$ is the pooling operator that can be *max* or *sum* pooling. Our proposal to compute the textual AMECON-features for an input caption is illustrated in Figure 3.

3.3 Visual AMECON-Features

In this section, we describe the proposed method to learn and compute the AMECON-features for the image modality. More precisely, we represent images through mid-level features extracted from pre-trained CNNs. Our goal is to project these mid-level features into the AMECON Space. To do this, we propose (in Sec. 3.3.1) to approach the projection problem as a *classification problem* with CNN features as inputs and the corresponding AMECON-features as ground-truth labels. We then, propose (in Sec. 3.3.2) to solve this classification problem using a neural network algorithm.

3.3.1 Textual AMECON-Features as Image Labels

At the core of our approach, we associate visual mid-level feature to *binary* textual AMECON features. It is posed as a classification problem with CNN features as input data and AMECON features as ground-truth labels. Indeed, the textual AMECON features being binary, they can be used as ground-truth labels for a multi-label supervised classification problem. Figure 4 illustrates the pipeline.

Formally, let consider a database $\mathcal{D}$ containing N pairs of text-image $(\mathcal{T}^i, I^i)$. From each image I^i and caption $\mathcal{T}^i$, we respectively extract their mid-level features $\mathbf{x}^{V,i}$ and $\mathbf{x}^{T,i}$. For each *textual* mid-level feature we compute the corresponding AMECON-feature as depicted in Sec. 3.3.1. We then use these binary features as ground-truth labels (during training) for the visual mid-level features $\mathbf{x}^{V,i}$. In the next section, we describe the classification algorithm used to solve this multi-label classification problem.

3.3.2 Learning the Visual AMECON-Features

To solve the above classification problem, we use an L-layer perceptron. The input layer is the visual mid-level features $\mathbf{x}^V$, the output layer is the *predicted visual* AMECON representation χ^V, and the ground truth label is the *binary* textual AMECON feature χ_{bin}^T. More concretely, by applying an affine transformation on $\mathbf{x}^V$, followed by an element-wise ReLU activation $f(z) = max(0, z)$ we

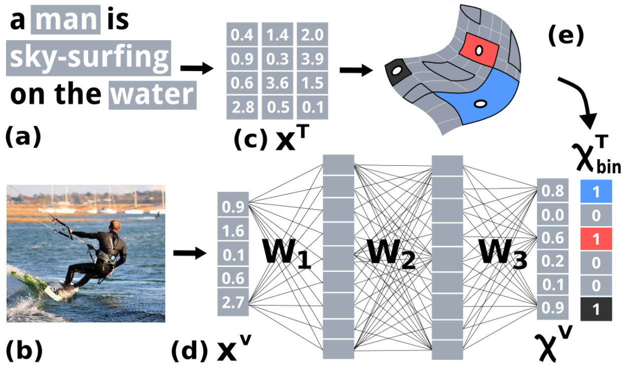

Figure 4: Illustration of the proposed method that consists to learn the visual AMECON-features through a multi-layer neural network using *binary* textual AMECON-features as ground-truth labels. Given an input image (b) and its associated caption (a), we first represent the three selected words and the image through mid-level features (one layer of a CNN (d) for the image and the word2vec features (c) for the words). Then, we project them in the AMECON space (e) and compute the *binary* textual AMECON-features χ^T_{bin} of the caption. This latter, is then used as output layer (ground-truth label for the input image vector). The shallow neural-network is finally learned to map the CNN features to the *binary* textual AMECON-features. Best view in color.

obtain the first hidden layer $h_1(\mathbf{x}^V)$ of a L-layer neural-network through:

$$h_1(\mathbf{x}^V) = f(W_1\mathbf{x}^V + b_1). \tag{3}$$

The following hidden layers are expressed by:

$$h_l(h_{l-1}) = f(W_l h_{l-1} + b_l), \forall l \in [2, \dots, L-2], \tag{4}$$

where W_l parametrizes the affine transformation of the l^{th} hidden layer and b_l is a bias term. In the same vein, we compute the output layer χ^V by:

$$\chi^V(h_{L-1}) = \sigma(W_L h_{L-1} + b_L), \tag{5}$$

where $\sigma(z) = \frac{1}{1+e^{-z}}$ is the sigmoid function that maps the raw scores to the predicted probabilities. We then implement the sigmoid cross-entropy loss function $\mathcal{L}$ that is computed for N samples through:

$$\mathcal{L} = \frac{1}{N}\sum_{k=1}^{N} \chi^T_{bin,k}\log(\chi^V_k) + (1-\chi^T_{bin,k})\log(1-\chi^V_k), \tag{6}$$

where $\chi^T_{bin,k}$ and χ^V_k are C-dimensional AMECON features for the k^{th} training example. The use of a *sigmoid cross-entropy* loss is better adapted to the *multi-label* problem than a *softmax* loss, since it leads to model the marginal probabilities while *softmax* leads to model the joint probability of the prediction. The cost function $\mathcal{L}$ is then minimized through asynchronized stochastic gradient descent.

Note that the training dataset $\mathcal{D}$ is composed of real-world images and texts that may contain very rich information. For instance, sentences may contain many entities and relations between them while images may contain very localized entities. Thus, it is important to consider this complex information in our model. For the text modality, our textual AMECON feature directly models this rich information by considering each word (that corresponds to *local* information) separately before pooling them together. Regarding the image modality, we follow the local schemes of [3] which models the rich information through the pooling of features extracted from local regions. Practically, we extract a set of R regions $\{\mathcal{R}_i, i \in [1, R]\}$ that have been identified into an image I. From each region, we extract a visual mid-level feature $\mathbf{x}^{V,\mathcal{R}_i}$. Then, all these local features are pooled into a global representation of the image through

$$\mathbf{x}^V = \underset{i=1\dots R}{\mathcal{P}}(\mathbf{x}^{V,\mathcal{R}_i}), \tag{7}$$

Method	Denotation	FlickR-8k			FlickR-30k		
		R@1	**R@5**	**R@10**	**R@1**	**R@5**	**R@10**
Karpathy *et al.* [13]	DeFrag	9.7	29.6	42.5	10.3	31.4	44.5
Kiros *et al.* [14]	MNLM	10.4	31.0	43.7	11.8	34.0	46.3
Mao *et al.* [16]	m-RNN	11.5	31.0	42.4	12.6	31.2	41.5
Karpathy *et al.* [12]	BRNN	11.8	32.1	44.7	15.2	37.7	50.5
Yan *et al.* [25]	DCCA	12.7	31.2	44.1	12.6	31.0	43.0
Tran *et al.* [23]	MACC†	10.2	29.3	41.4	12.4	33.5	46.1
Our Approach	AMECON	**15.9**	**37.9**	**49.5**	**18.3**	**41.3**	**53.5**

Table 1: Comparison of our approach with state-of-the-art methods on text-illustration task through the FlickR-8k and FlickR-30k datasets. The second columns states the denotation of the different methods. Each method is evaluated on its R@1, R@5 and R@10. All scores are those released in the original papers, except those marked with † that were re-implemented by ourselves for fair comparisons.

where $\mathcal{P}$ is the pooling operator (max or sum). The resulting mid-level visual features $\mathbf{x}^V$ that models the local information of images are thus used as inputs of the neural-network.

During the test phase, given an input image, we extract its mid-level feature according to Eq. (7), then compute its projection into the AMECON space through a forward pass on the learned network, which results in the *predicted* visual AMECON feature χ^V. In this space, features that are projection from visual and textual data are directly comparable which allows us to perform multi-modal tasks.

4 Experiments

In this section, we evaluate the performance of our approach in a cross-modal retrieval task namely text-illustration through two datasets. Before comparing the results of our method to state-of-the-art in Sec. 4.3, we describe (in Sec. 4.1) the different datasets that we use and the implementation details (in Sec. 4.2).

4.1 Datasets

We evaluate our system on two datasets commonly used for the task of text-illustration, namely FlickR-8K [10] and FlickR-30K [26]. Both of them contain images from FlickR groups, but they differ by their size. In fact, the former (FlickR-8k) contains 8,000 images while the latter consists of 31,783 images. Moreover, each image is associated to five captions (sentences) thus, they also differ by their number of texts, *i.e.*, 40,000 captions for FlickR-8k and 158,915 for FlickR-30k. The two datasets have an official training, validation and testing split that consists of 6,000 images in FlickR-8k and 29,783 in FlickR-30k for training, and 1,000 images for validation and test sets in both datasets. In each subset, the images are associated to their five captions. Since, even the test images are associated to *five* captions and not *one*, different evaluation protocols have been used in the literature. Thus, we used a popular protocol [13, 12, 25] where each caption is treated individually, *i.e.* each of the 5,000 captions has to be illustrated by one image from the whole test set of $1,000$ images. For both datasets, we adopt recall at top K retrieved results (denoted Recall@K or R@K in the following) as an evaluation metric. We follow the literature and set K $\in \{1, 5, 10\}$.

4.2 Implementation Details

Representations: For all experiments, the mid-level feature used to represent words and images are respectively represented using the word2vec [17] representation (300-dimensional vector) and the penultimate fully-connected layer (4096-dimensional vector) extracted from a pre-trained

CNN [3, 19]. Once the mid-level features are computed for each modality, they are projected in the AMECON Space (Section 3.1). More precisely, each textual caption is represented through the binary textual AMECON feature, as depicted in Section 3.3.1 and each image is represented through the visual AMECON features with respect to the method described in Section 3.3.2. Note that, for the captions, we apply pre-processing that aims to remove *stop-words* following the pipeline provided by [2]. It is also worth noting that during training, each image I_i is associated to five captions ($\mathcal{T}_1, \dots, \mathcal{T}_5$). Thus, we use them as five *different* training examples that result in the following set of text-image pairs $\{(I_i, \mathcal{T}_1)$, ..., $(I_i, \mathcal{T}_5)\}$. Regarding, the CNN features used to represent images, we used the popular VGG [19] network, slightly modified one to be pre-trained on a diversified set of ImageNet [5] images. As depicted in Sec. 3.3.2, each image is represented by the pooling of a global representation (from the whole image) and local features (from local regions). Regarding the exact regions extracted from each image, we follow [21] and extract the full image as region $\mathcal{R}_0$ and choose following $\mathcal{R}_{i>0}$ according to regular grid at a smaller scale ($2/3$ of the image size).

Neural Networks: We used the Caffe framework [11] to train the networks using standard parameters (*e.g.*, learning rate: 10^{-4}, momentum: 0.9, weight decay: $5 \cdot 10^{-4}$, batch size: 512). The networks were trained with full back propagation *from scratch*, *i.e.*, using a random initialization (with respect to a Gaussian law) of the weights. Regarding the architecture of the neural network in Section 3.3.2, we used a standard multi-layer perceptron and tested different architectures through cross-validation on each dataset. More precisely, we tested with one to three hidden layers (the L parameter of Eq. 4 is set to 3, 4 or 5) and for each layer, we set a number of hidden units to one of the following values: $\{1024, 2048, 3072, 4096\}$. Note that, the number of hidden-units is set according to each hidden layer, *i.e*, one layer can be of size 4096 and the other of size 1024. Regarding the input and output layers, they respectively corresponds to the visual CNN feature (4096 units) and to the *binary* textual AMECON feature (K units since the AMECON space has K dimensions). The K parameter has also been set by cross-validation and we conducted an analysis on its impact in A.2. Also important, each layer of the multi-layer perceptron is followed by a ReLU and a dropout function.

4.3 Text-Illustration Results

In this section, we evaluate our method for the text-illustration task on the two datasets presented above. We compare our method to the methods of the literature that reports the best results for text-illustration. All scores of the comparison methods are those released in the original papers, except those of Tran *et al.* [23]. Indeed, this very recent paper achieves great results on multi-modal tasks but uses another evaluation protocol different from ours. Thus, we re-implemented their method and evaluated it with our protocol for a fair comparison. The results on the FlickR-8k and FlickR-30k datasets are presented in Table 1.

The best results of the literature on the two datasets were achieved by the method of [12] (BRNN). Our approach gives an absolute improvement of 4.1 points of R@1 on FlickR-8k and 3.1 points of R@1 on FlickR-30k.

As said in Sec. 2, here we evaluate our method on one direction of cross-modal retrieval, namely *text-illustration*. By definition, our method could also technically deal with the inverse cross-modal retrieval task that consist to retrieve texts from image queries, which is also well known as *image-captioning*. The performances of our proposal on that task are still below those of the state-of-the-art, certainly due to the *asymmetrical* property of our approach. As mentioned above, achieving good performances on one direction of a cross-modal task when building the common latent space on the inverse direction, remains an open problem.

5 Conclusion

We introduced the Abstract Meta-Concept principle to perform text-illustration. Contrary to most of recent work on this topic, we consider an *asymmetric* scheme to process both modalities. The unifying common space contains concepts that are *abstract* and that *subsumes* several semantic-concepts.

We evaluated our method on a text-illustration task and obtained significantly better results than recent methods on publicly available benchmarks, namely FlickR-8k and FlickR-30k. We also conducted an in-depth analysis of the parameters of our method, including an ablation study that shows the relative importance of each component of the proposed pipeline.

Above the formal definition of the AMECON and the experiments that demonstrate its efficiency on a particular application task, the proposed principle would be confirmed if one could perform the inverse projection to that proposed here, namely from the AMECON space to the each original feature space, or original modality.

A Appendix: Model Parameters

A.1 Impact of Each Component

The goal of this section is to compare our proposal to baseline methods in order to demonstrate the utility of each component. Roughly, on the textual side, our method represents each word of an input caption with a word embedding vector (word2vec) and projects them in the AMECON space. We then use a *hard-coding* process to compute the textual AMECON features. Thus, in this section, we denote our method by $\mathbf{T_{loc}}$+$\mathbf{C_{hard}}$+$\mathbf{V_{loc}}$, with $\mathbf{T_{loc}}$ meaning a *local* textual representation (extracted from *each word* w.r.t Eq. (1)) in the sentence, $\mathbf{C_{hard}}$, a *hard*-coding process, and $\mathbf{V_{loc}}$ a *local* visual representation (extracted from each *local region* w.r.t Eq.(7)). We thus compare our method to three baseline methods that differ from ours by one or two component which are replaced by baseline components. The following items describe the baseline methods:

- $\mathbf{T_{loc}}$+$\mathbf{C_{hard}}$+$\mathbf{V_{glob}}$: In this baseline method, we use a *global* visual representation instead of a local one. More specifically, the visual CNN features are extracted only from the *global* image;
- $\mathbf{T_{loc}}$+$\mathbf{C_{soft}}$+$\mathbf{V_{loc}}$: Here, we use a *soft*-coding process instead of a *hard*-coding one. Indeed, for each dimension of the textual AMECON features, we compute the euclidean distance between the word embedding vector and the vector representing the corresponding AMECON cluster;
- $\mathbf{T_{glob}}$+$\mathbf{C_{soft}}$+$\mathbf{V_{loc}}$: In this baseline approach, we use a *global* textual representation instead of a local one. Practically, for an input caption, we compute the word features for all words and then average them in a vector that corresponds to a *global* representation of that caption. This latter is then projected to the AMECON space through Eq (1) and coded with *soft*-coding.

The results are presented in Table 2. Our method clearly outperforms the three baselines. More precisely, $\mathbf{T_{loc}}$+$\mathbf{C_{hard}}$+$\mathbf{V_{loc}}$ is better than $\mathbf{T_{loc}}$+$\mathbf{C_{soft}}$+$\mathbf{V_{loc}}$ which proves the utility of the *hard*-coding process. Our method also outperforms $\mathbf{T_{loc}}$+$\mathbf{C_{hard}}$+ $\mathbf{V_{glob}}$ and $\mathbf{T_{glob}}$+$\mathbf{C_{soft}}$+$\mathbf{V_{loc}}$ which demonstrates the utility of the modelization of the local visual and textual information in our scheme. Moreover, the results of the baseline $\mathbf{T_{glob}}$+$\mathbf{C_{soft}}$+$\mathbf{V_{loc}}$ are very low, which confirms the clear need of binary outputs in the textual AMECON feature and a computation of locality (at least on the textual modality).

Method	**R@1**	**R@5**	**R@10**
$\mathbf{T_{loc}}$+$\mathbf{C_{hard}}$+$\mathbf{V_{glob}}$	12.8	32.5	43.0
$\mathbf{T_{glob}}$+$\mathbf{C_{soft}}$+$\mathbf{V_{loc}}$	1.5	3.5	5.1
$\mathbf{T_{loc}}$+$\mathbf{C_{soft}}$+$\mathbf{V_{loc}}$	13.1	30.6	41.5
$\mathbf{T_{loc}}$+$\mathbf{C_{hard}}$+$\mathbf{V_{loc}}$	**15.9**	**35.9**	**48.0**

Table 2: Comparison of our method (denoted $\mathbf{P_{all}}$+$\mathbf{C_{hard}}$) to three baseline methods ($\mathbf{P_{avg}}$+$\mathbf{C_{soft}}$, $\mathbf{P_{all}}$+$\mathbf{C_{soft}}$ and $\mathbf{P_{avg}}$+$\mathbf{C_{hard}}$) that are described in Sec. A.1. The evaluation is carried in a text-illustration task through the FlickR-8K dataset, with $\mathbf{C = 700}$ and $\mathbf{m = 3}$.

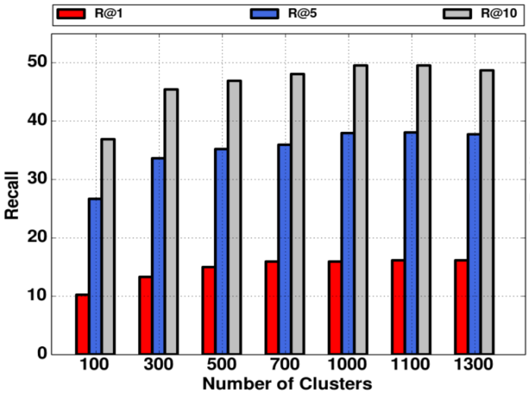

Figure 5: Evaluation of the impact of the number of selected clusters on our method in text-illustration through the FlickR-8k dataset. The graph presents the recall (R@1, R@5 and R@10) according to the number of clusters. Best view in color.

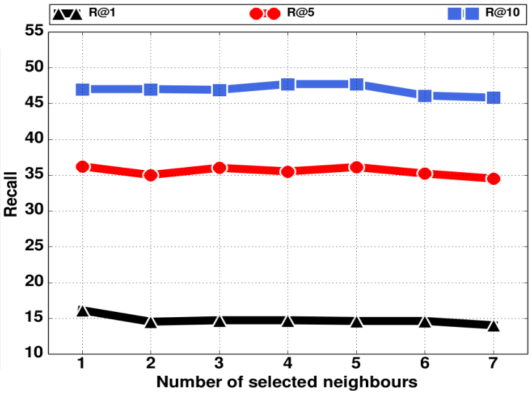

Figure 6: Evaluation of the impact of the number of selected neighbours ($\mathbf{m}$ parameter) on our method in text-illustration through the FlickR-8k dataset. The graph presents the recall (R@1, R@5 and R@10) according to the number of selected neighbours. Best view in color.

A.2 Impact of the Number of AMECONs

In this section, we study the impact of the parameter C in Equations (1) and (2), that corresponds to the number of abstract meta-concepts (clusters) and thus to the dimensionality of the AMECON Space. To evaluate its impact on our method, we set it to the seven values of the following set: $\{100, 300, 500, 700, 1000, 1100, 1300\}$. For instance, $C = 700$ means that the clustering algorithm (Sec. 3.2.1) was set to output 700 clusters that directly correspond to the AMECONs. Therefore, the dimensionality of our textual AMECON feature (Sec. 3.3.1) is 700 and the mapping for the visual side is from a 4096-dimensional CNN feature to a 700-dimensional textual AMECON feature.

The results of our method for the different values of the C parameter evaluated on the FlickR-8k dataset are presented in Figure 5. We clearly observe that increasing the C parameter significantly improves the retrieval results. It is also important to note that, *globally* (from 300 to $1,100$), the results are very close to one another, meaning that our method is quite robust to the number of selected clusters (AMECONs).

A.3 Impact of the Number of Neighbours

In this section, we evaluate the impact of the m parameter of Equation (1) that corresponds to the number of selected neighbours when performing the hard-coding process for each word. To evaluate its impact on our method, we set it in the seven values of the following set: $\{1, 2, 3, 4, 5, 6, 7\}$. For instance, $m = 3$ means that three dimensions are activated in the textual AMECON features computed by Equation (2).

The results of our method for the different values of the m parameter evaluated on the FlickR-8k dataset are presented in Figure 6. We clearly observe that the three curves (R@1, R@5 and R@10) are quite flat. This latter means that our method is desirably highly robust to the m parameter.

References

[1] A. Bergamo and L. Torresani. Meta-class features for large-scale object categorization on a budget. In *CVPR*, 2012.

[2] S. Bird, E. Klein, and E. Loper. *Natural language processing with Python*. " O'Reilly Media, Inc.", 2009.

[3] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. *arXiv preprint arXiv:1405.3531*, 2014.

[4] J. Costa Pereira, E. Coviello, G. Doyle, N. Rasiwasia, G. Lanckriet, R. Levy, and N. Vasconcelos. On the role of correlation and abstraction in cross-modal multimedia retrieval. *PAMI*, pages 521–535, 2014.

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. CVPR, 2009.

[6] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, T. Mikolov, et al. Devise: A deep visual-semantic embedding model. NIPS, 2013.

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, 2014.

[8] Y. Gong, Q. Ke, M. Isard, and S. Lazebnik. A multi-view embedding space for modeling internet images, tags, and their semantics. *IJCV*, 106(2):210–233, Jan. 2014.

[9] D. R. Hardoon, S. R. Szedmak, and J. R. Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural Computation*, 16(12):2639–2664, 2004.

[10] M. Hodosh, P. Young, and J. Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research*, 47:853–899, 2013.

[11] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACM International Conference on Multimedia*, ACM, 2014.

[12] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. CVPR, 2015.

[13] A. Karpathy, A. Joulin, and L. Fei Fei. Deep fragment embeddings for bidirectional image sentence mapping. NIPS, pages 1889–1897, 2014.

[14] R. Kiros, R. Salakhutdinov, and R. S. Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *CoRR*, abs/1411.2539, 2014.

[15] L. Liu, L. Wang, and X. Liu. In defense of soft-assignment coding. In *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, pages 2486–2493, Washington, DC, USA, 2011. IEEE Computer Society.

[16] J. Mao, W. Xu, Y. Yang, J. Wang, and A. L. Yuille. Explain images with multimodal recurrent neural networks. *CoRR*, abs/1410.1090, 2014.

[17] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. NIPS, 2013.

[18] M. Schuster and K. K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, Nov 1997.

[19] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[20] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE T PAMI*, 22:1349–1380, 2000.

[21] Y. Tamaazousti, H. Le Borgne, and A. Popescu. Constrained local enhancement of semantic features by content-based sparsity. In *ICMR*, 2016.

[22] L. Torresani, M. Szummer, and A. Fitzgibbon. Efficient object category recognition using classemes. ECCV, 2010.

[23] T. Q. N. Tran, H. Le Borgne, and M. Crucianu. Aggregating image and text quantized correlated components. In *Computer Vision and Pattern Recognition*, CVPR, 2016.

[24] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *Computer Vision and Pattern Recognition*, CVPR, 2010.

[25] F. Yan and K. Mikolajczyk. Deep correlation for matching images and text. In *Computer Vision and Pattern Recognition*, CVPR, 2015.

[26] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014.