AMECON: Abstract Meta-Concept Features for Text-Illustration

Ines Chami[1*] Youssef Tamaazousti[2*] Hervé Le Borgne[2]
[1]Stanford University, Institute for Computational and Mathematical Engineering, California, USA
[2]CEA, LIST, Laboratory of Vision and Content Engineering, F-91191 Gif-sur-Yvette, France
[*]Both authors contributed equally to this work.

ABSTRACT

Cross-media retrieval is a problem of high interest that is at the frontier between computer vision and natural language processing. The state-of-the-art in the domain consists of learning 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, each of its dimension subsuming several semantic concepts ("meta" aspect) but may not correspond to an actual one ("abstract" aspect). In practice, the textual modality is used to generate a multi-label representation, further used to map the visual modality through a simple shallow neural network. 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. The source code is available at http://perso.ecp.fr/~tamaazouy/.

CCS CONCEPTS

•**Computing methodologies → Computer vision representations; Learning latent representations;** Image representations;

KEYWORDS

Multi-Modal Common Space, Text-Illustration, Neural Networks, Abstract Meta Concepts

ACM Reference format:
Ines Chami[1*] Youssef Tamaazousti[2*] Hervé Le Borgne[2] . 2017. AMECON: Abstract Meta-Concept Features for Text-Illustration. In *Proceedings of ICMR'17, Bucharest, Romania, June 06-09, 2017,* 9 pages.
DOI: http://dx.doi.org/10.1145/3078971.3078993

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 [7, 11, 16, 17, 32].

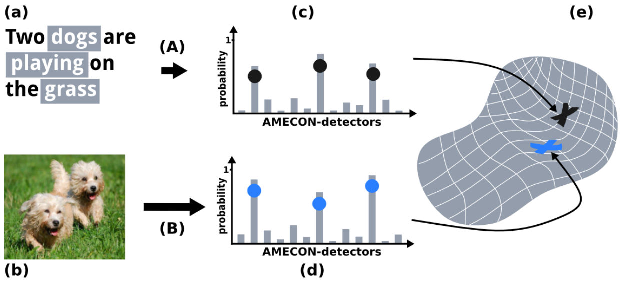

Figure 1: Given an input text (a) and an input image (b), our method computes the AMECON-features for each modality (textual (c) and visual (d)) and matches them in the AMECON Space (e). The key novelty of our approach is that, each dimension of our AMECON representation (each *bar* in (c) and (d)) corresponds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

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 [22] features for textual content and one layer from a pre-trained Convolutional Neural Network (CNN) [3, 25] 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 [11] consisted in maximizing the correlation of the aligned data once projected in the common space. More recently, Frome *et al.* [7] proposed a visual visual-semantic embedding that maps the visual representation to the language model by learning a similarity metric that produces a higher score for aligned data than non-aligned ones. The second family of approaches relies on deep networks to model a full multi-modal embedding. This is the case of [16] who proposed to infer the correspondences between images

ICMR'17, Bucharest, Romania
© 2017 Copyright held by the owner/author(s). Publication rights licensed to ACM. 978-1-4503-4701-3/17/06. . . $15.00
DOI: http://dx.doi.org/10.1145/3078971.3078993

and their sentence description. First, they use a Region Convolutional Neural Network (RCNN) [8] as image representation and a Bidirectional Recurrent Neural Network (BRNN) [24] to textual one. Second, they define a loss that encourages aligned image-sentences pairs to have a higher score than misaligned pairs. A RNN is then learned to generate a description for new images. Another interesting recent work is that of [35] who learn a text-image embedding using a two-view neural network. Two layers of non-linearities are used on top of a CNN-based visual representation and sentences represented by Fisher vectors [19]. They define a loss that tends to force aligned images and sentences to be reciprocal first neighbours within the latent space. Other related works are detailed in Section 2.

Our initial motivation is the same as the work of [7], that aimed at matching the visual and textual representations. However, our model differs from the previous work by proposing very different pipelines to process both modalities. Indeed, most of previous works use two similar pipelines in parallel, 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 [26] that reflects the fact that textual features are closer to human understanding (and language) 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. As explained in the following, the AMECON models a higher-level representation of the human knowledge, that is "meta" since each of its dimension subsumes several semantic concepts and "abstract" because each of them may not correspond to a unique existing semantic concept.

A *semantic concept* can be named by a word from the vocabulary of a given language. In line with [1, 10, 27], a *meta-concept* is defined by a concept subsuming several semantic concepts. Moreover, 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, in the sense they can represent some notion where seemingly unrelated concepts can be mapped together because of some invisible intrinsic quality that they share but which is not obvious to humans. We consider that each modality, namely visual and textual, described in its original feature space is an (imperfect) observation of the AMECON space from a particular point of view. A key particularity of our approach is that the mapping from one modality to this space is specific to said modality. For reasons given above, 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 task (*i.e.*, retrieve best images from a textual query). While being quite easy to implement, our method exhibits performance above the current state-of-the-art on Text-Illustration. Indeed, we conducted extensive experiments (in Sec. 4) on two publicly available benchmarks, namely Flickr-8k and Flickr-30k, on which our method significantly outperforms the previous works. We also conducted (in Sec. 5) 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) [11]. This work has then be extended by [9] 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 [9] 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 [31]. 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.

A major drawback of KCCA-based approaches is their tendency to group the projections with respect to each modality rather than the actual semantic of the content. To compensate this effect, [32] proposed to quantify the common space then code each modality according to the resulting codebook. Their method also include a "completion" of each projected modality. Such a completion compensates the modality separation identified in the common space and can be effective for retrieval even without quantization [33].

Rather than relying on an a priori principle such as maximizing the correlation, other works consider deep neural networks with other type of constraint. Ngiam *et al.* [23] used a deep autoencoder to learn a common space for videos and speech audio data. It allows them to manage the absence of a modality and thus perform cross-modal retrieval. As already cited in the introduction, Frome *et al.* proposed DeViSE [7] 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 closer to ours, in the sense that it tries to directly match the visual representation to the textual one. However, our work differs from them 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 [17] and [16], visual data is also aligned with sentences, thanks to a structured loss that forces aligned sentences and images to be close reciprocal neighbours. In the same vein but with a lighter architecture [35] tends to preserve the local neighbourhood of corresponding text and images by forcing the distances of correct matches in the latent space to be smaller than the wrong ones. 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.

Let note that, a recently released pre-print [6] also proposes an *asymmetric* method that matches text features to CNN-based image representations. The key differences with our work is the mapping as well as the common space. In fact, our common latent space is based on the proposed AMECON principle while they directly use the original visual features as the common space. Also interesting, the asymmetry of both approaches limits the performances on the inverse cross-modal task. Indeed, our method works very well on Text-Illustration but does not work so well on the inverse task (*i.e.*, image-captioning). The same phenomenon seems to appear in [6], since they only evaluate their method on image-captioning. Hence, 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 build a multi-modal common space is named "Abstract Meta-Concept" (**AMECON**). It consists to match texts and images in an AMECON common latent 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. In this paper, we focus on the application of AMECON to the Text-Illustration task.

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

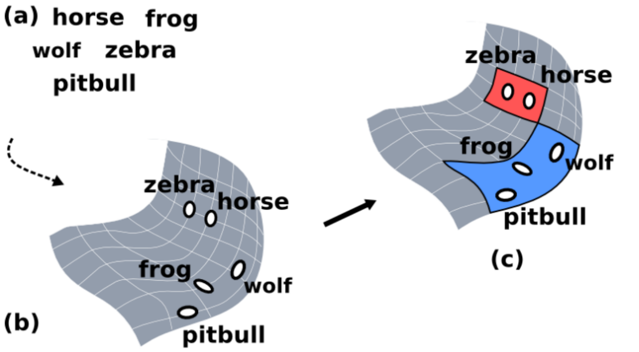

Figure 2: Illustration of the proposed AMECON principle. Given a set of words from a training corpus (a) and their projection in a word embedding space (b), our method clusters the space (c) such that each cluster is an *abstract meta-concept* that corresponds to an *abstract* concept (do not exist in 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. Indeed, the abstraction comes from the fact that inside the group, the words are not semantically related which makes the group *noisy* and thus prevents a human to attribute it a *correct* semantic concept. Best view in color.

Definition 2. An *abstract meta-concept detector* ($\chi_i(\mathbf{x})$) is a visual $\chi_i^V(\mathbf{x})$ or textual $\chi_i^T(\mathbf{x})$ 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.

Formally, the proposed multi-modal representation corresponds to a C-dimensional vector $\chi(\mathbf{x}) = [\chi_1(\mathbf{x}), \ldots, \chi_C(\mathbf{x})]$, where each dimension $\chi_i(\mathbf{x})$ is the output of an abstract meta-concept classifier that can be obtained by one or both modalities. More specifically, $\chi_i(\mathbf{x}) = max(\chi_i^T(\mathbf{x}), \chi_i^V(\mathbf{x}))$ with $\chi_i^T(\mathbf{x})$ and $\chi_i^V(\mathbf{x})$ are respectively the *textual* and *visual* abstract meta-concept detectors. Note that, we have as many visual classifiers as textual ones, thus the visual (χ^V) and textual representation (χ^T) are both of size C.

This scheme has the advantage to consider both modalities for *bi-modal* tasks (bi-modal retrieval and classification) or only one modality for *mono-modal* tasks (cross-modal retrieval and classification). More precisely, in the former case, it is straightforward to compute $\chi_i(\mathbf{x})$ since both modalities are available while in the latter case, where only one modality is available, we set the output values (of the detectors) of the missing modality to zero, resulting to

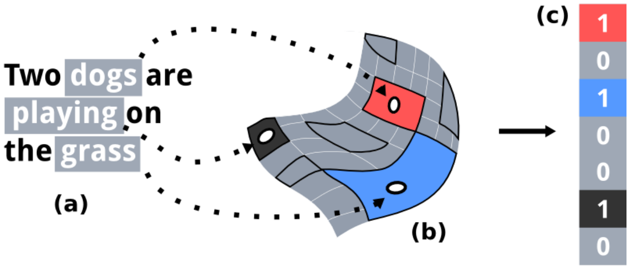

Figure 3: Illustration of the proposed textual AMECON-features. Given an input caption (a), our method first selects the non stop-words (coloured in gray), computes their mid-level features and projects them in the clustered word embedding (b) that corresponds to our AMECON space. After the projection, when a word embedding representation falls in an abstract meta-concept (*e.g*, blue cluster), its associated dimension is activated (*e.g*, 3rd 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.

consider only the available modality. Moreover, for cross-modal retrieval where we need to retrieve the nearest documents from another modality, we 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 AMECONs verifies its two definitional characteristics: (i) it groups similar data into a generic cluster that thus corresponds to a *meta*-concept and (ii) because of 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 [22]). 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. Note that, in [27] and [10], the *meta* concepts are obtained through *manual* annotations. An illustration of our AMECON principle is given 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 (*e.g.*, word2vec) of dimension d. Then, we group the word vectors 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 $(\mathbf{c}_\mathrm{i})_{i=1,\dots,C} \in \mathbb{R}^d$. Hence, within an AMECON cluster, words have *similar* semantic connotations (*i.e.*, similar in the sense of the word representation used). Here, we used the K-means algorithm for clustering but obviously, other unsupervised algorithms (*e.g.*, spectral-clustering, MeanShift, etc.) could be used. 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 [20], originally introduced as locality-constrained linear coding [34], that is nevertheless binarized. Given a caption $\mathcal{T}$ composed of n words, we compute the mid-level representation $\mathbf{x}_j^T \in \mathbb{R}^d$ of each word, resulting into a set of n vectors $\{\mathbf{x}_j^T\}_{j=1,\dots,n}$ in the word embedding space.

The j^{th} word is then encoded according to the codebook in the C-dimensional vector, its k^{th} dimension being:

$$\chi^T_{bin,j}(k) = \begin{cases} 1 & \text{if } k \in NN^m(\mathbf{x}_j^T) \\ 0 & \text{otherwise} \end{cases} \qquad (1)$$

where $NN^m(\mathbf{x}_j^T)$ is the set of indexes of the m nearest AMECON clusters of $\mathbf{x}_j^T$ in the word embedding space. It is thus a "local hard coding" of $\mathbf{x}_j^T$ 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. The representation, in this C-dimensional space, of the i^{th} caption result from the pooling of its word's representation:

$$\chi^T_{bin,i} = \underset{j=1\dots n}{\mathcal{P}} (\chi^T_{bin,j}), \qquad (2)$$

where $\mathcal{P}$ is the pooling operator that can be *max* or *sum* pooling. In the following we use the same notation for the representation of a word j and that of caption i since both of them lie in the same space. 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 first represent images through mid-level features extracted from a pre-trained CNN. Our goal is to project these mid-level features into the learned 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 textual AMECON-features as ground-truth labels. We then, propose (in Sec. 3.3.2) to solve this classification problem using a shallow neural network.

3.3.1 Textual AMECON-Features as Image Labels. At the core of our approach, we associate visual mid-level features 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 (reduced to a single label if the textual document is a unique word). Figure 4 illustrates the pipeline.

Formally, let consider a training 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 $\mathbf{x}_i^V$ and $\mathbf{x}_i^T$. For each *textual* mid-level features we compute the corresponding AMECON-features as depicted in Sec. 3.2. We then use these binary features as ground-truth labels (during training) for the visual mid-level features $\mathbf{x}_i^V$. In the next section, we describe the classification algorithm used to solve this multi-label classification problem.

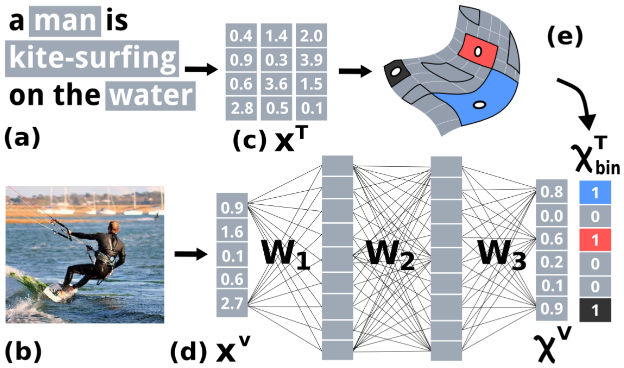

Figure 4: Illustration of the learning of the visual AMECON features. Given an input image (b) and its associated caption (a), we first represent the three selected words and the image through mid-level features (one layer of a CNN (d) for the image and word2vec features (c) for the words). Then, we project the word2vec features 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). A shallow neural-network is finally learned to map from the CNN features to the *binary* textual AMECON-features. Best view in color.

3.3.2 Learning the Visual AMECON-Features. To learn the mapping of visual mid-level features to the AMECON Space, we use a shallow neural-network classification algorithm. Formally, let re-consider the *training* database $\mathcal{D}$ of N text-image pairs $(\mathcal{T}^i, I^i)$. Each image I^i is represented as mid-level features $\mathbf{x}_i^V$ – *e.g* one layer of a pre-trained CNN – and its corresponding caption is represented using textual mid-level features $\mathbf{x}_i^T$ which are mapped to *binary* textual AMECON features $\chi^T_{bin,i}$ that result from the aggregation via Eq. (2). As depicted in the previous section, the mid-level visual features $\mathbf{x}_i^V$ are used as inputs and the textual features $\chi^T_{bin,i}$ are used as ground-truth labels.

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 neural network contain $L-1$ hidden layers. More concretely, by applying an affine transformation on $s(\mathbf{x}^V)$, followed by an element-wise ReLU activation $f(z) = max(0, z)$ we obtain the first hidden layer $h_1(\mathbf{x}^V)$ of the 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:

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

where W_l parametrizes the affine transformation of the l^{th} hidden layer and b_l are the bias terms. 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:

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

where $\chi^T_{bin,i}$ and χ^V_i are the C-dimensional AMECON features for the i^{th} example. The use of a *sigmoid cross-entropy* loss is better adapted to the *multi-label* problem than the commonly used *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, 29] 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:

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

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 features 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" (*i.e* retrieving the best representative images given a text query) through two datasets. Before comparing the results of our method to the 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 commonly used datasets for the task of Text-Illustration, namely Flickr-8K [13] and Flickr-30K [37]. 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. We use the official training, validation and testing splits that consists of 6,000 images in Flickr-8k and $29,783$ in Flickr-8k 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 the most common protocol [16, 17, 36] 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 R@K in the following) as an evaluation metric. We follow the literature and set $\mathrm{K} \in \{1, 5, 10\}$.

4.2 Implementation Details

Representations: For all experiments, the mid-level features used to represent words and images are respectively the word2vec [22] representation (300-dimensional vector) and the penultimate fully-connected layer (4096-dimensional vector) extracted from a pre-trained CNN. Once the mid-level features are computed for each modality, they are projected in the AMECON Space (Sec. 3.1). More precisely, each textual caption is represented through the binary textual AMECON feature, as depicted in Sec. 3.2.2 (with max-pooling for $\mathcal{P}$) and each image is represented through the visual AMECON features with respect to the method described in Sec. 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 $(\mathcal{T}_1^i, \ldots, \mathcal{T}_5^i)$. Thus, we use them as five *different* training examples that result in the following set of text-image pairs $\{(I^i, \mathcal{T}_1^i), \ldots, (I^i, \mathcal{T}_5^i)\}$. Regarding, the CNN features used to represent images, we used the pre-trained VGG network of [30] that has been trained on a diversified set of ImageNet [5], which gives slightly better results than the standard VGG [25]. Note that, our method could also benefit from other best representations such as [12, 14, 28]. 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 [29] and extract the full image as region $\mathcal{R}_0$ and choose the following $\mathcal{R}_{i>0}$ according to a regular grid at a smaller scale ($2/3$ of the image size). We use the *max*-pooling operator in Eq. (7) and the euclidean distance to compute the similarity in the AMECON space.

Neural Networks: We used the Caffe framework [15] 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 Sec. 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: $\{2048, 3072, 4096\}$. 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. We conducted an in-depth analysis about the network architecture in Sec. 5.4. Regarding the input and output layers, they respectively corresponds to the visual CNN features (4096 units) and to the *binary* textual AMECON feature (C units since the AMECON space has C dimensions). The C parameter has also been set by cross-validation and we conducted an analysis on its impact in Sec. 5.2. 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.* [32]. 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. Regarding the parameters (network architecture, C and m) of our method, they have been set by cross-validation on the validation set of each dataset. For instance, on Flickr-8k, the best performances on the validation set are obtained with $C = 1,000$, $m = 3$ and an architecture of 2 layers with 4096 units each. The results on the two datasets are presented in Table 1.

The best results of the literature on the two datasets were achieved by the method of [16] (BRNN). However, it is important to consider that above their interesting vision-language integration method, they use a costly representation on the visual side. More precisely, from each image they extract $2,000$ salient regions through a RCNN then pool the local features of the top 19 detections, which is *unfair* with respect to other methods. On our side, we only extract 5 regions per image, and attain much better results (*e.g.*, an absolute improvement of 4.1 points of R@1 on FlickR-8k and 3.1 points of R@1 on Flickr-30k). Moreover, our proposal outperforms all other methods on the two datasets. Specifically, it outperforms the best method (except BRNN) with an improvement of, 4.1 to 5.8 absolute points of R@K on FlickR-8k and, 3.0 to 5.7 absolute points of R@K on Flickr-30k.

As said in Sec. 2, here we evaluate our method on one direction of cross-modal retrieval, namely *text-illustration*. By construction, 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 said above, getting good performances on one direction of cross-modal task when building the common latent space on the inverse direction, remains an open problem.

5 ANALYSIS

5.1 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,

Method	Denotation	Flickr-8k			Flickr-30k		
		R@1	R@5	R@10	R@1	R@5	R@10
Karpathy *et al.* [17]	DeFrag	9.7	29.6	42.5	10.3	31.4	44.5
Kiros *et al.* [18]	MNLM	10.4	31.0	43.7	11.8	34.0	46.3
Mao *et al.* [21]	m-RNN	11.5	31.0	42.4	12.6	31.2	41.5
Karpathy *et al.* [16]	BRNN*	11.8	32.1	44.7	15.2	37.7	50.5
Yan *et al.* [36]	DCCA	12.7	31.2	44.1	12.6	31.0	43.0
Tran *et al.* [32]	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 the 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. The Method marked with * was achieved using a costly visual representation that consists to pool the top-19 features after an extraction of 2,000 regions per images.

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 features and a computation of *locality* (projection *each* word in the AMECON space), 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 $\mathbf{T_{loc}}$+$\mathbf{C_{hard}}$+$\mathbf{V_{loc}}$) to three baseline methods, that are described in Sec. 5.1. The evaluation is carried in a Text-Illustration task through the FlickR-8K dataset, with $C = 700$ and $m = 3$.

5.2 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 features (Sec. 3.3.1) is 700 and the mapping for the visual side is from a 4096-dimensional CNN features to a 700-dimensional textual AMECON features.

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, from 700 to $1,300$ the results are quite similar, meaning that our method is quite robust to the number of selected clusters (AMECONs).

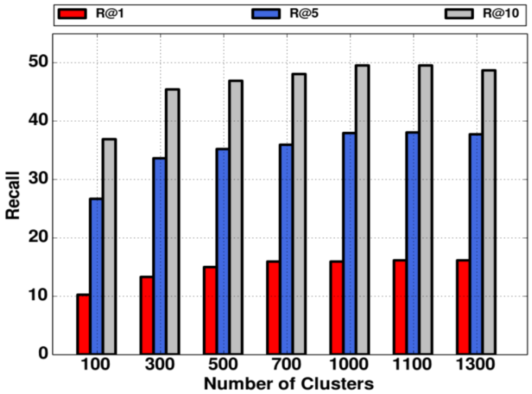

Figure 5: Evaluation of the impact of the number of selected clusters on our method in Text-Illustration through the FlickR-8k dataset. The graph presents the recall (R@1, R@5 and R@10) according to the number of clusters. Best view in color.

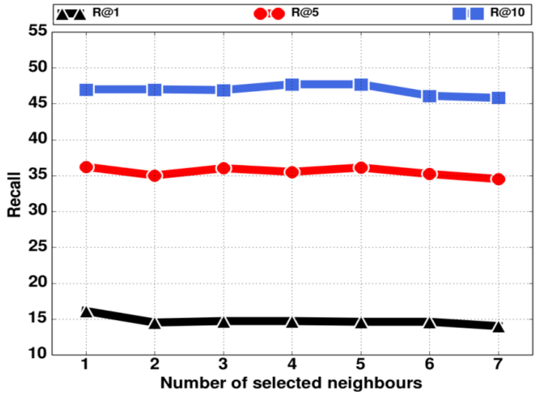

Figure 6: Evaluation of the impact of the number of selected neighbours (m parameter of Eq. (1)) on our method in Text-Illustration through the FlickR-8k dataset. The graph presents the R@1, R@5 and R@10 according to the number of selected neighbours. Here we report the scores with $C = 700$ and a 2-layer architecture with 4096 units each. Best view in color.

5.3 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.

L	Architecture	R@1	R@5	R@10
2	2048-2048	14.6	35.4	46.8
2	3072-2048	15.2	**36.6**	47.8
2	3072-3072	15.0	36.3	47.8
2	4096-3072	**16.2**	**36.6**	**48.0**
2	4096-4096	15.9	35.9	**48.0**
3	2048-2048-2048	13.0	33.6	45.0
3	3072-3072-3072	14.5	35.5	47.0
3	4096-4096-4096	15.8	36.4	47.7

Table 3: Text-illustration results on the FlickR-8K dataset with different network architectures. The first column indicates the number of hidden-layers in the architecture and the second indicates the number of units per hidden-layer. Here we report the scores with $C = 700$ meaning that we have 700 units in the output layer and $m = 3$.

5.4 Neural-Network Architecture

In this section, we evaluate the results of our method for different neural-network architectures. We conducted the experiment on the Flickr-8k dataset with the C and m parameters respectively fixed to 700 and 3. Since the C parameter is fixed to 700, the dimensionality of the textual AMECON features and thus the number of units in the output layer of the neural-network are 700. Regarding the hidden-layers of the neural-network, we set the L parameter of Equation (4) to two values (2 and 3). The size of each hidden-layer was set to one of the following values $\{2048, 3072, 4096\}$.

The results are given in Table 3. We observe that the best results are given with only 2 hidden-layers, which is desirable since no high computational complexity is needed to achieve great performance. Regarding the number of units in each hidden-layer, we can roughly say that increasing the number of units leads to better performance.

6 CONCLUSION

We introduced the Abstract Meta-Concept principle to build a multimodal common space and we demonstrated its ability on a Text-Illustration task. Contrary to most of recent work on this topic, we consider an *asymmetric* scheme to process both modalities and 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 demonstrate its success on the inverse retrieval task, namely image-captioning (*i.e.*, retrieving the best captions given an image query).

REFERENCES

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

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

[3] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2015. Return of the devil in the details: Delving deep into convolutional nets. In *BMVC*.

[4] Jose Costa Pereira, Emanuele Coviello, Gabriel Doyle, Nikhil Rasiwasia, Gert Lanckriet, Roger Levy, and Nuno Vasconcelos. 2014. On the role of Correlation and Abstraction in Cross-Modal Multimedia Retrieval. *PAMI* (2014).

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

[6] Jianfeng Dong, Xirong Li, and Cees G. M. Snoek. 2016. Word2VisualVec: Image and Video to Sentence Matching by Visual Feature Prediction. In *ArXiv 1604.06838*.

[7] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Tomas Mikolov, and others. 2013. Devise: A deep visual-semantic embedding model. In *NIPS*.

[8] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2014. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *CVPR*.

[9] Yunchao Gong, Qifa Ke, Michael Isard, and Svetlana Lazebnik. 2014. A Multi-View Embedding Space for Modeling Internet Images, Tags, and Their Semantics. *IJCV* (2014).

[10] Amirhossein Habibian, Thomas Mensink, and Cees G.M. Snoek. 2015. Discovering Semantic Vocabularies for Cross-Media Retrieval. In *ICMR*.

[11] David R. Hardoon, Sandor R. Szedmak, and John R. Shawe-Taylor. 2004. Canonical correlation analysis: An overview with application to learning methods. *Neural Computation* (2004).

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*.

[13] Micah Hodosh, Peter Young, and Julia Hockenmaier. 2013. Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research* (2013).

[14] Gao Huang, Zhuang Liu, Kilian Q Weinberger, and Laurens van der Maaten. 2016. Densely connected convolutional networks. In *ArXiv 1608.06993*.

[15] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. 2014. Caffe: Convolutional architecture for fast feature embedding. In *ACM Multimedia*.

[16] Andrej Karpathy and Li Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *CVPR*.

[17] Andrej Karpathy, Armand Joulin, and Li Fei Fei. 2014. Deep fragment embeddings for bidirectional image sentence mapping. In *NIPS*.

[18] Ryan Kiros, Ruslan Salakhutdinov, and Richard S. Zemel. 2014. Unifying Visual-Semantic Embeddings with Multimodal Neural Language Models. In *ArXiv 1411.2539*.

[19] B. Klein, G. Lev, G. Sadeh, and L. Wolf. 2015. Fisher vectors derived from hybrid gaussian-laplacian mixture models for image annotation. In *CVPR*.

[20] Lingqiao Liu, Lei Wang, and Xinwang Liu. 2011. In Defense of Soft-assignment Coding. In *ICCV*.

[21] Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, and Alan L. Yuille. 2014. Explain Images with Multimodal Recurrent Neural Networks. In *ArXiv 1410.1090*.

[22] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*.

[23] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y Ng. 2011. Multimodal deep learning. In *ICML*.

[24] M. Schuster and K. K. Paliwal. 1997. Bidirectional recurrent neural networks. *Signal Processing* (1997).

[25] Karen Simonyan and Andrew Zisserman. 2015. Very deep convolutional networks for large-scale image recognition. In *ICLR*.

[26] Arnold W. M. Smeulders, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh Jain. 2000. Content-Based Image Retrieval at the End of the Early Years. *PAMI* (2000).

[27] Youssef Tamaazousti, Hervé Le Borgne, and Céline Hudelot. 2016. Diverse Concept-Level Features for Multi-Object Classification. In *ICMR*.

[28] Youssef Tamaazousti, Hervé Le Borgne, and Céline Hudelot. 2017. MuCaLe-Net: Multi Categorical-Level Networks to Generate More Discriminating Features. In *CVPR*.

[29] Youssef Tamaazousti, Hervé Le Borgne, and Adrian Popescu. 2016. Constrained Local Enhancement of Semantic Features by Content-Based Sparsity. In *ICMR*.

[30] Youssef Tamaazousti, Hervé Le Borgne, Adrian Popescu, Etienne Gadeski, Alexandru Ginsca, and Céline Hudelot. 2017. Vision-Language Integration using Constrained Local Semantic Features. *CVIU* (2017).

[31] Lorenzo Torresani, Martin Szummer, and Andrew Fitzgibbon. 2010. Efficient Object Category Recognition Using Classemes. In *ECCV*.

[32] Thi Quynh Nhi Tran, Hervé Le Borgne, and Michel Crucianu. 2016. Aggregating Image and Text Quantized Correlated Components. In *CVPR*.

[33] Thi Quynh Nhi Tran, Hervé Le Borgne, and Michel Crucianu. 2016. Cross-modal Classification by Completing Unimodal Representations. In *ACM Multimedia Workshop*.

[34] Jinjun Wang, Jianchao Yang, Kai Yu, Fengjun Lv, Thomas Huang, and Yihong Gong. 2010. Locality-constrained linear coding for image classification. In *CVPR*.

[35] Liwei Wang, Yin Li, and Svetlana Lazebnik. 2016. Learning Deep Structure-Preserving Image-Text Embeddings. In *CVPR*.

[36] Fei Yan and Krystian Mikolajczyk. 2015. Deep correlation for matching images and text. In *CVPR*.

[37] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *ACL* (2014).