Semantic Web 0 (0) 1
IOS Press

Studying the Impact of the Full-Network Embedding on Multimodal Pipelines

Armand Vilalta [a,*], Dario Garcia-Gasulla [a], Ferran Parés [a], Eduard Ayguadé [a,b], Jesus Labarta [a,b], E Ulises Moya-Sánchez [a] and Ulises Cortés [a,b]

[a] *Barcelona Supercomputing Center (BSC), Jordi Girona 1-3, 08034 Barcelona, Spain*
E-mails: armand.vilalta@bsc.es, dario.garcia@bsc.es, ferran.pares@bsc.es
[b] *Universitat Politècnica de Catalunya, Jordi Girona 1-3, 08034 Barcelona, Spain*

Editors: Luis Espinosa-Anke, Cardiff University, United Kingdom; Thierry Declerck, German Research Centre for Artificial Intelligence, Germany; Dagmar Gromann, Technical University Dresden, Germany
Solicited reviews: Vered Shwartz, Bar-Ilan University, Israel; Luis Espinosa-Anke, Cardiff University, United Kingdom; Jindrich Helcl, Charles University, Czech Republic

Abstract. The current state of the art for image annotation and image retrieval tasks is obtained through deep neural network multimodal pipelines, which combine an image representation and a text representation into a shared embedding space. In this paper we evaluate the impact of using the Full-Network embedding (FNE) in this setting, replacing the original image representation in four competitive multimodal embedding generation schemes. Unlike the one-layer image embeddings typically used by most approaches, the Full-Network embedding provides a multi-scale discrete representation of images, which results in richer characterisations. Extensive testing is performed on three different datasets comparing the performance of the studied variants and the impact of the FNE on a levelled playground, *i.e.*, under equality of data used, source CNN models and hyper-parameter tuning. The results obtained indicate that the Full-Network embedding is consistently superior to the one-layer embedding. Furthermore, its impact on performance is superior to the improvement stemming from the other variants studied. These results motivate the integration of the Full-Network embedding on any multimodal embedding generation scheme.

Keywords: Multimodal Embedding, Full-Network Embedding, Caption Retrieval, Image Retrieval, Deep Neural Network

1. Introduction

One of the main challenges of the semantic web is vagueness, the difficulty of representing imprecise concepts. An increasing trend in the community is to use vector representations of vague concepts. Vector representations allow for the evaluation of concepts similarity simply by computing a vector distance. Not less important is the possibility of obtaining these vector representations automatically. The use of automated large scale semantic tagging of ambiguous content can bootstrap and accelerate the creation of the semantic web [7].

Deep learning methods are representation learning techniques which can be used to generate such vectors. The models obtained from these methods are composed of multiple processing layers that learn representations of data with multiple levels of abstraction. These methods have dramatically improved the state of the art in speech recognition, visual object recognition, object detection and many other domains [23]. The use of deep learning vector embeddings to represent words has had an a substantial impact in many natural language processing tasks [6] through the use of vector representations. Similarly, deep learning image embeddings have shown great generalisation capabilities, even between distant domains [12]. In this regard,

*Corresponding author. E-mail: armand.vilalta@bsc.es.

we argue that the semantic web can significantly benefit from the use of deep learning based embeddings.

In this paper we focus on multimodal pipelines, which tackle two problems in parallel. First, the problem of obtaining a semantically meaningful embedding of an image representing a scene. Second, the problem of obtaining a visually meaningful embedding of a sentence describing a scene. This is done through the construction of a joint embedding, representing both modalities: an image of a scene, and a caption describing it.

The joint embedding constructed can be used to correlate images with sentences easily. As an example, imagine an e-commerce website where sellers upload images of the product to sell. Some sellers may add a very accurate textual description too, while others' descriptions may be incomplete, inaccurate or nonexistent at all. On the other side, buyers search for the desired product writing a free-text description of it. The use of a multimodal embedding as proposed can help to link the textual information provided by the buyer with the product that best matches it, regardless an accurate description for that individual product was provided or not. In general, this approach can be used to automatically include representations of uncaptioned images in a semantic web.

The proposed methodology can also have an impact on semantic web technologies in disambiguation of vague semantics. Take for instance the concept *sports car*. Certainly, the speed limit of the car is a key factor to define the concept, but there are many high-end cars with very high speed limits which we would not consider a *sports car*. The main difference between them and a *sports car* is that they do not *look* like a *sports car*. In this case, the comparison with a *sports car* visual embedding can be a key element in the definition of the concept.

Information retrieval is a natural way to asses the quality of joint embedding methods [14]. Image annotation (also known as caption retrieval) is the task of automatically associating an input image with a describing text. The complementary task of associating an input text with a fitting image is known as image retrieval or image search.

State-of-the-art image annotation methods are currently based on deep neural network representations, where an image embedding (*e.g.*, obtained from a convolutional neural network or CNN) and a text embedding (*e.g.*, obtained from a recurrent neural network or RNN) are combined into a unique multimodal embedding space. While several techniques for merging both spaces have been proposed in the past [10, 11, 15, 17, 20–22, 26, 29, 34, 37], little effort has been made in finding the most appropriate image embeddings to be used in that process. In fact, most approaches use a straight-forward one-layer CNN embedding [8, 32], and the only method proposed to increase the quality of the image embedding relies on obtaining more data to allow for fine-tuning the CNN in the final stage of training [11].

The main goal of this paper is to explore the impact of using a Full-Network embedding (FNE) [12] to generate the image embedding required by multimodal pipelines, replacing the standard one-layer embedding. We do so by integrating the FNE into the multimodal embedding pipeline defined in *Unifying visual-semantic embeddings with multimodal neural language models* (UVS) [21]. This pipeline is based in the use of a Gated Recurrent Units neural network (GRU) [5] for text encoding and a single-layer CNN embedding for image encoding. Unlike one-layer embeddings, the FNE represents features of varying levels of abstraction by integrating information from different layers of the CNN. This particularity results in a richer visual embedding space, which may be more reliably mapped to a shared visual-textual representation. Furthermore, we hypothesise that the FNE discretization (to 3 values with contextual implications) makes for a more natural mapping to a linguistic representation of concepts than using a regular real-valued embedding.

The generic pipeline defined by Kiros *et al.*[21] had been outperformed in image annotation and image search tasks by methods specifically targeting either one of those tasks [9, 22]. However, more recent work by Vendrov *et al.*[37] and Faghri *et al.*[11], based on the same generic pipeline, has outperformed previous methods in both tasks, which shows the potential of the approach. This paper extends our previous work [39] by integrating and thoroughly evaluating the improvements proposed by Vendrov *et al.*[37] and Faghri *et al.*[11]. Additionally, some hindrances found on Faghri *et al.*[11] are studied, and a methodology for solving them is proposed which also increases performance.

We report the consequential improvements in our implementation, which increase the performance of the original method [21] as well. Finally, we exhaustively test the main variations on a levelled playground, obtaining insights on the real impact on performance of each of them. Indeed, properly assessing the sources of empirical gains is a key aspect in research that should be further encouraged [25]. Evaluation is done

using three publicly available datasets: Flickr8K [30], Flickr30K [42] and MSCOCO [24].

To sum up, the contributions of this paper are:

- Integration of the FNE into the generic pipeline defined by Kiros *et al.*[21], the Order Embedding by Vendrov *et al.*[37] and the Order++ and VSE++ Embeddings by Faghri *et al.*[11].
- Comparative study of the impact on performance of the main variants introduced by [37] and [11] under equality of the rest of hyper-parameters.
- Exhaustive study of optimal hyper-parameter configuration for the previous methods.
- Novel curriculum learning process to further increase Order++ and VSE++ [11] training stability and performance.

The rest of the paper is structured as follows: In Section 2 the main different approaches existing in the literature for the image/caption retrieval problem are reviewed. This review introduces the basic methodology by Kiros *et al.*[21] and the other approaches studied in this paper. Beyond these, other proposals are considered, grouped according to their similitude with [21] and the possibility to be integrated with the FNE. Afterwards, in Section 3, the FNE and multimodal embedding methods studied here are described in further detail. The last subsection contains the methodology we propose to solve the issues found on the method from Faghri *et al.*[11]. Then, Section 4 presents all the information relative to the experiments conducted. This includes a description of the public datasets used, together with important notes on the choices made here and in the related works. Follows an extensive subsection explaining the details of the implementation which help to improve the results from our previous work [39]. Section 5 contains a discussion of the results obtained. Then, in Section 6, we focus specifically on the experimental difficulties we found when using the methodology of Faghri *et al.*[11]. Finally, Section 7 gathers the most important findings of this work.

2. Related work

This paper builds upon the methodology described by Kiros *et al.*[21], which is in turn based on previous works in the area of Neural Machine Translation[35]. In their work, Kiros *et al.*[21] define a vectorized representation of an input text by using GRU RNNs. In this setting, each word in the text is codified into a vector embedding, vectors which are then fed one by one into the GRUs. Once the last word vector has been processed, the activations of the GRUs at the last time step conveys the representation of the whole input text in the multimodal embedding space. In parallel, images are processed through a CNN pre-trained on ImageNet [31], extracting the activations of the last fully connected layer to be used as a representation of the images. To solve the dimensionality matching between both representations (the output of the GRUs and the last fully-connected layer of the CNN) an affine transformation is applied on the image representation.

Following the same pipeline [21], Vendrov *et al.*[37] proposed an asymmetric order-embedding space. Its main hypothesis is that captions convey more general abstractions than the images, such as the hypernym/hyponym relation. This relation is imposed in the embedding using the order error similarity defined in Eq. (3). Another improvement on the same pipeline was proposed by Faghri *et al.*[11]. This method, instead of taking into account all the contrastive examples, focus only in the hardest of them. This improvement has also been applied to order embeddings successfully [11]. The present work studies the application of the FNE to these methods and variants.

Also, using two different neural networks for image and text, and the ranking loss as methodology keystone, we find the Embedding Network (EN) presented in [41] and the Word2VisualVec (W2VV) model [9]. The first approach (EN) introduces a novel neighbourhood constraint in the form of additional loss penalties *i.e.*, the captions describing the same image should be placed together and far from other captions, and analogously for images. The second approach (W2VV), while restricted to the specific problem of image annotation, also obtain competitive results. This approach uses as a multimodal embedding space the same visual space where images are represented, involving a deeper text processing. These two methods are very similar to the ones presented in this work thus are good candidates to benefit from same improvements (*e.g.*, FNE).

A substantially different group of methods is based on the Canonical Correlation Analysis (CCA). A first successful approach in this direction is the use of Fisher Vectors (FVs) [22]. FVs are computed with respect to the parameters of a Gaussian Mixture Model (GMM) and an Hybrid Gaussian-Laplacian Mixture Model (HGLMM). For both, images and text, FVs are build using deep neural network features: a CNN for images features, and a word2vec [27] for text fea-

tures. A more recent approach based on the same CCA methodology [10], introduces a novel bidirectional neural network architecture. This architecture is based on two channels which share weights: one channel maps images to sentences while the other goes in the opposite direction. Losses are applied in each projection and in a middle layer. The loss in the middle layer seeks to ensure the correlation between both representations at this point. Instead of using the CCA, a more efficient euclidean loss is used. Since both methods rely on a CNN representation of the image, the introduction of the FNE in these pipelines should be straightforward.

Attention-based models is another family of competitive solutions for tackling multimodal tasks. Dual Attention Networks (DANs) [29] currently holds the best results on the Flickr30K dataset. On a general pipeline similar to [21], DANs introduce two additional small neural networks as attention mechanisms for images and captions. This allows DANs to estimate the similarity between images and sentences by focusing on their shared semantics. In a similar fashion, selective multimodal Long Short-Term Memory network (sm-LSTM) [15] includes a multimodal context-modulated attention scheme at each time-step. This mechanism can selectively attend to a pair of instances of image and sentence, by predicting pairwise instance-aware saliency maps for image and sentence. All attention-based methods rely on CNN representations of the images, as the previously described methods did. However, they differ in that the representations are obtained from the last convolutional layer. At this level, information on the features position is available allowing for the use of attention mechanisms. On the contrary, FNE obtains a compact representation of the whole image at the cost of losing the spatial information. Application of the FNE methodology to those techniques would require to modify significantly the FNE schema and is one of our main lines of future work.

3. Methods

The multimodal embedding pipeline of Kiros *et al.*[21] represents images and textual captions within the same space. The pipeline is composed of two main elements, one which generates image embeddings and another one which generates text embeddings. In this work we replace the original image embedding generator by the FNE, resulting in the architecture shown in Figure 1. In subsection 3.1 the main characteristics and methods of the FNE are described. Subsection 3.2 explains the generic multimodal embedding pipeline by Kiros *et al.*[21] alongside with the main modifications proposed, including the integration of the FNE. Following subsections 3.3 and 3.4 explain the variations introduced by Vendrov *et al.*[37] and Faghri *et al.*[11] respectively. Finally, 3.5 explains the methodology developed to overcome the hindrances found in maximum loss methods.

3.1. Full-network Embedding

The FNE [12] generates a vector representation of an input image by processing it through a pre-trained CNN, extracting the neural activations of all convolutional and fully-connected layers. After the initial feature extraction process, the FNE performs a dimensionality reduction step for convolutional activations, by applying a spatial average pooling on each convolutional filter. After the spatial pooling, every feature (from both convolutional and fully-connected layers) is standardized through the z-values, which are computed over the whole image train set. This standardization process puts the value of each feature in the context of the dataset. At this point, the meaning of a single feature value in an image is the degree with which the feature value is atypically high (if positive) or atypically low (if negative) for that image in the context of the dataset. Zero marks the typical behavior.

The last step of the FNE is a feature discretization process. The previously standardized embedding is usually of large dimensionality (*e.g.*, 12,416 features for VGG16 [33]) which entails problems related with the curse of dimensionality. A common approach to address this issue would be to apply some dimensionality reduction methods (*e.g.*, PCA) [1, 28]. Instead, the FNE reduces expressiveness through the discretization of features, while keeping the dimensionality. Specifically, the FNE discretization maps the feature values to the $\{-1, 0, 1\}$ domain, where -1 indicates an unusually low value (*i.e.*, the feature is significant by its absence for an image in the context of the dataset), 0 indicates that the feature has an average value (*i.e.*, the feature is not significant) and 1 indicates an uncommonly high activation (*i.e.*, the feature is significant by its presence for an image in the context of the dataset). The mapping of standardized values into these three categories is done through the definition of two constant thresholds. The optimal values of these thresholds can be found empirically for a

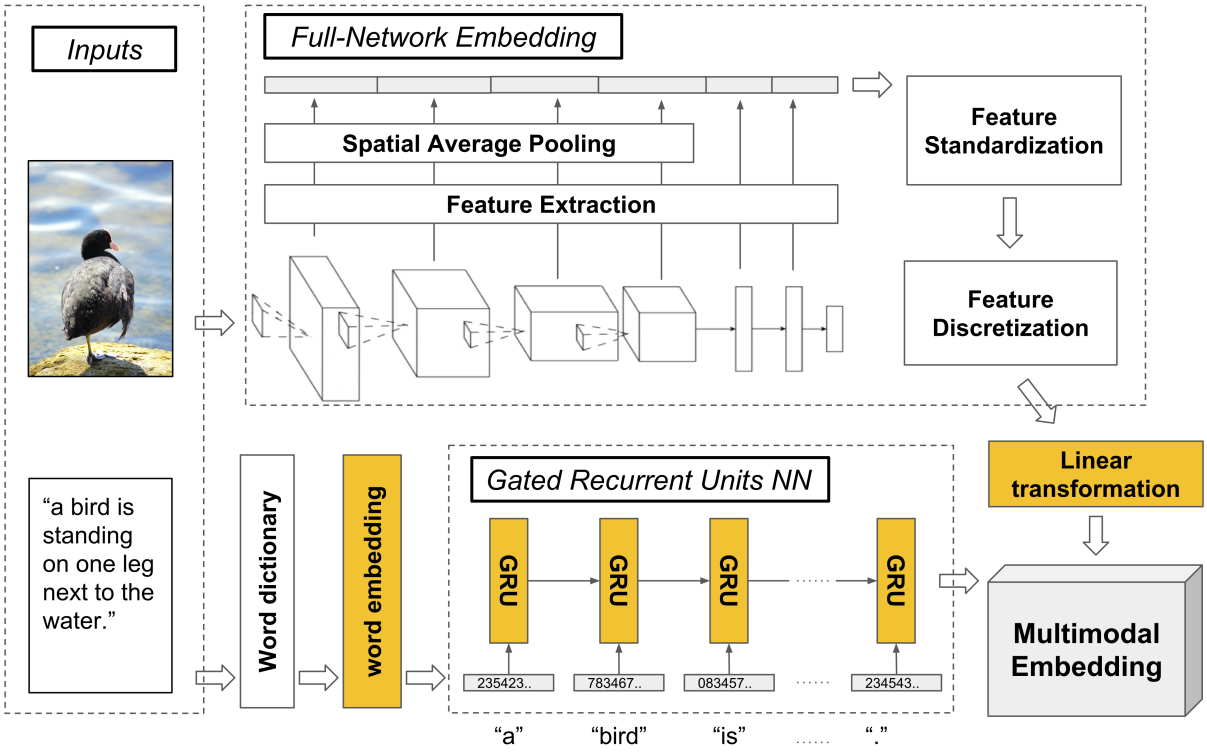

Fig. 1. Overview of the proposed multimodal embedding generation pipeline with the integrated full-network embedding. Elements colored in orange are components modified during the neural network training phase. During testing, only one of the inputs is provided.

labeled dataset [13]. However, we use certain threshold values shown to perform consistently across several domains [12].

3.2. Multimodal embedding

In our approach, we integrate the FNE with the multimodal embedding pipeline of Kiros *et al.*[21]. To do so we obtain the FNE image representation instead of the output of the last layer of a CNN, as the original model does. The encoder architecture processing the text is used as in the original pipeline, using a GRUs recurrent neural network to encode the sentences. Each word in the sentence is first encoded in a one-hot vector using a dictionary containing all the words in the train and validation sets. Next, it is encoded through a trainable linear embedding into a word embedding of lower dimensionality. Finally, the embeddings are fed to a GRU and the final state of the GRU's hidden units is normalised to obtain the sentence embedding. To combine both embeddings, Kiros *et al.*[21] use an affine transformation on the image representation (in our case, the FNE) analogous to a fully connected neural network layer with identity activation function. We simplified it by removing the bias term, resulting in a linear transformation as in [37]. This simplification is also motivated by the good results of W2VV [9], where the transformation is completely removed. The output of the linear transformation is normalised to obtain the embedding. This linear transformation is trained simultaneously with the GRUs and the word embedding. The elements of the multimodal pipeline that are tuned during the training phase of the model are shown in orange in Figure 1 (notice the image embedding is not fitted to the data).

In simple terms, the pipeline training procedure consists of the optimisation of the pairwise ranking loss between the correct image-caption pair and a random pair. Assuming that a correct pair of elements should be closer in the multimodal space than a random pair, the loss $\mathcal{L}_S$ can be formally defined as follows:

$$\mathcal{L}_S = \sum_{\mathbf{I}} \sum_{k} \max(0, \alpha - \mathcal{S}(\mathbf{i}, \mathbf{c}) + \mathcal{S}(\mathbf{i}, \mathbf{c}_k)) \quad (1)$$
$$+ \sum_{\mathbf{C}} \sum_{k} \max(0, \alpha - \mathcal{S}(\mathbf{i}, \mathbf{c}) + \mathcal{S}(\mathbf{c}, \mathbf{i}_k))$$

Where $\mathbf{i}$ is an image vector, $\mathbf{c}$ is its correct caption vector, and $\mathbf{i}_k$ and $\mathbf{c}_k$ are sets of random images and captions respectively. $\mathbf{I}$ and $\mathbf{C}$ are, respectively, the sets of images and captions in the train subset. The operator $\mathcal{S}(\bullet,\bullet)$ stands for a similarity metric. This formulation is a Hinge Loss as it includes a margin term α to avoid pulling the image and caption closer once their distance is smaller than the margin. This makes the optimisation focus on pulling together distant pairs instead of improving the ones that are already close.

The similarity metric proposed in [21] is the cosine similarity $\mathcal{S}_{COS}$ defined in Eq. (2). In our case, since all embeddings $(\mathbf{c},\mathbf{i})$ are already normalised to have unit norm, it is equivalent to the dot product of the vectors.

$$\mathcal{S}_{COS}(\mathbf{c},\mathbf{i}) = \frac{\mathbf{c}\cdot\mathbf{i}}{\parallel \mathbf{c} \parallel \cdot \parallel \mathbf{i} \parallel} \tag{2}$$

3.3. Multimodal Order Embedding

Using the same general schema, Vendrov *et al.*[37] proposed an asymmetric order embedding space. Their main hypothesis is that captions are abstractions of the images, including information such as the hypernym/hyponym relation. In the resulting shared embedding space, an image corresponds to a caption if the value of all components of the image embedding have higher values than the components of the caption embedding ($i_k > c_k \forall i_k \in \mathbf{i}, c_k \in \mathbf{c}$). This relation is imposed during training, using the order error similarity $\mathcal{S}_{OE}$ defined in Eq. (3) instead of the cosine similarity in the same contrastive loss formulation defined in Eq. (1).

$$\mathcal{S}_{OE}(\mathbf{c},\mathbf{i}) = -\parallel \max(0, \mathbf{c}-\mathbf{i}) \parallel^2 \tag{3}$$

Notice that since image and caption embeddings are normalised to have unit L2-norm, both lay on an hypersphere centred on its coordinate origin, thus a perfect order-embedding will not be achieved unless they are the same vector, which is extremely unlikely to happen.

3.4. Maximum error loss

A recent contribution to the field [11] proposes to compute the loss focusing only on the worst contrasting example (*i.e.*, the closest mistake) instead of taking into account all the examples. To achieve it, Eq. (1) is modified substituting the sum over all contrasting examples for the maximum contrasting example, as shown in Eq. (4).

$$\begin{aligned}\mathcal{L}_M = &\sum_{\mathbf{I}} \max_k \{\max(0, \alpha - \mathcal{S}(\mathbf{i},\mathbf{c}) + \mathcal{S}(\mathbf{i},\mathbf{c}_k))\} \\ &+ \sum_{\mathbf{C}} \max_k \{\max(0, \alpha - \mathcal{S}(\mathbf{i},\mathbf{c}) + \mathcal{S}(\mathbf{c},\mathbf{i}_k))\}\end{aligned} \tag{4}$$

3.5. Curriculum learning

Faghri *et al.*[11] reported problems in training when using their proposed Maximum of Hinge Loss (MH). They indicate that a rough form of curriculum learning [2] could be applied, but do not develop or experiment it further as in their preliminary experiments it obtained worse performance than the proposed method. Our experiments replicated their training problems, as well as an unstable behaviour with respect to hyperparameter selection. As a result, on several occasions, the model is unable to start learning within a reasonable number of epochs.

To fix that, we define a sort of curriculum learning approach to combine the benefits of the sum loss $\mathcal{L}_S$ and the max loss $\mathcal{L}_M$. The basic idea is to train using one method until there is no improvement in the validation set. Then, take this pre-trained model and train it again using a different method. Several of those training steps can be concatenated.

We propose to train the model using the sum of errors loss $\mathcal{L}_S$, to obtain the best performing model and, in a second step, to re-train it using the maximum error loss $\mathcal{L}_M$. Notice that different hyper-parameters may be used in each training phase as long as the dimensionalities of the embeddings are not changed. The motivation to define this process is the intuition that using the sum loss $\mathcal{L}_S$ help to a achieve first a generally correct embedding which is refined using the max loss $\mathcal{L}_M$ that focus on improving single misplaced examples.

We performed preliminary experiments using this methodology to apply a learning rate reduction, which resulted in small performance gains for some algorithms. We kept these results out of the paper as we do not consider them to be conclusive enough, and to avoid shadowing more relevant contributions.

4. Experiments

In this section, we evaluate the impact of using the FNE in a multimodal pipeline for both image annota-

tion and image retrieval tasks. We extend our previous work [39] introducing the FNE in different multimodal pipelines. To properly measure the relevance of the FNE, we compare the results obtained with those of the original multimodal pipelines (*i.e.*, without the FNE). Given the discrepancies in the experimental setup of the different contributions, we define baselines by keeping as much of the original setup as possible while leveling the playground (*i.e.*, using the same training and test sets, the same text preprocessing, the same source CNN, the same data augmentation, *etc.*).

We identify the different combinations of embedding and multimodal pipeline with a notation in the form of EMB-PIPE. EMB denotes the embedding being either FNE (for the full network embedding) or FC7 (for the baselines using the last CNN layer, `fc7`). PIPE denotes the multimodal pipeline used, one of SH, MH, SOE, MOE, PH, POE. The details of each pipeline and the hyper-parameters used in the experiments can be found in Section 4.2.

4.1. Datasets

In our experiments we use three different and publicly available datasets:

The **Flickr8K** dataset [30] contains 8,000 hand-selected images from Flickr, depicting actions and events. Five correct captions are provided for each image. Following the provided splits, 6,000 images are used for train, 1,000 are used for validation and 1,000 are kept for testing.

The **Flickr30K** dataset [42] is an extension of Flickr8K. It contains 31,783 photographs of everyday activities, events and scenes. Five correct captions are provided for each image. In our experiments 29,000 images are used for training, 1,014 conform the validation set and 1,000 are kept for test. These splits are the same ones used in [16, 21].

The **MSCOCO** dataset [24] includes images of everyday scenes containing common objects in their natural context. For captioning, 82,783 images and 413,915 textual descriptions are available for training, while 40,504 images and 202,520 captions are available for validation. Captions from the test set are not publicly available. Previous contributions consider using a subset of the validation set for validation and the rest for test. In most cases, such subsets are composed by either 1,000 or 5,000 images per set, with their corresponding 5 captions per image. In our experiments we only consider the 1K test set to simplify results presentation. Some previous work extend the training set by adding the images and captions in the original validation set that are not used for validation or test [11, 37]. This split raises the number of training images to 113,287, consequently increasing the performance of algorithms [11]. We did not consider using this extended training set since the effect of the quantity of training data is already seen on the performance obtained for the 3 different datasets (which have different sizes).

4.2. Experimental Setup

We investigate the impact of the FNE on the methods proposed in [11, 21, 37], and on the curriculum learning methodology proposed in Section 3.5. The methods are named following the convention of [11]. Notice all losses are actually based on a Hinge Loss:

- Sum of Hinge Loss (**SH**). Uses the sum loss $\mathcal{L}_S$ with cosine similarity $\mathcal{S}_{COS}$. Analogous to UVS [21]
- Maximum of Hinge Loss (**MH**). Uses the max loss $\mathcal{L}_M$ with cosine similarity $\mathcal{S}_{COS}$. Analogous to VSE++ [11]
- Sum of Order Embedding Loss (**SOE**). Uses the sum loss $\mathcal{L}_S$ with order embedding similarity $\mathcal{S}_{OE}$. Analogous to Order [37]
- Maximum of Order Embedding Loss (**MOE**). Uses the max loss $\mathcal{L}_M$ with order embedding similarity $\mathcal{S}_{OE}$. Analogous to Order++ [11]
- Pre-trained Hinge Loss (**PH**). Use curriculum learning. Pre-train using the sum loss $\mathcal{L}_S$ and fine-tune using the max loss $\mathcal{L}_M$ using always cosine similarity $\mathcal{S}_{COS}$.
- Pre-trained Order Embedding Loss (**POE**). Use curriculum learning. Pre-train using the sum loss $\mathcal{L}_S$ and fine-tune using the max loss $\mathcal{L}_M$ using always order embedding similarity $\mathcal{S}_{OE}$.

The details of the hyper-parameters used in the experiments for each method can be found in Table 1.

4.3. Implementation Details

The devil is in the details. To facilitate the reproducibility and interpretability of our work, we provide in this section all the details regarding our implementation. The Theano [36] based implementation we used is available at [38].

Table 1
Hyper-parameter configuration for the experiments

Model		SH	SH-bl	MH	MH-bl	SOE	SOE-bl	MOE	MOE-bl	PH	POE
Loss		sum	sum	max	max	sum	sum	max	max	sum-max[b]	sum-max[b]
Similarity		cos	cos	cos	cos	order	order	order	order	cos	order
Embed. dim.	f8k	1536	2048	1024	1024	1024	1024	1536	1024	1536	1024
	f30k	1536	2048	1536	1024	1024	1024	1536	1024	1536	1024
	coco	1536	2048	2048	1024	1536	1024	2048	1024	1536	2048
Word embed. dim.		1024	1000[a]	1024	300	1024	300	1536	300	1024	1024
Learning rate		0.0002	0.0002[a]	0.0002	0.0002	0.001	0.001	0.001	0.0002	0.0002	0.001 - 0.0001[b]
Margin		0.2	0.2	0.2	0.2	0.05	0.05	0.05	0.2	0.2	0.05
Absolute value embed.		✗	✗	✗	✗	✓	✓	✓	✗	✗	✓

[a] For MSCOCO Word embedding dimensionality is 2000 and Learning rate is 0.00025. [b] First training - second training parameters

4.3.1. Training

During a training epoch, all images are presented with one caption chosen randomly from the five captions available. This approach differs from the usual of presenting all five captions per image each epoch [21, 39]. If all five image-caption pairs are included in the dataset, it may be the case that more than one correct image-caption pairs can be included in the same random batch. Since the method uses all image-caption combinations in the batch as contrastive examples, a correct pair could be wrongly used as an incorrect pair during the loss computation, leading to noise during the training. By using only one correct caption, we remove this possibility. On the other hand it is now possible (although highly unlikely depending on the number of training epochs) that a correct caption is never used during training. In fact, the probability that a correct caption is never used during training is in the order of 10^{-8} for our setups. Practically, this approach implies that to achieve similar training it requires five times the number of epochs (but with the same computational cost). On the other side, it reduces the memory requirements to almost 1/5.

The models are trained until a maximum number of epochs is reached, and the best performing model on the validation set is chosen. Notice that the result of this process is very similar to what could be obtained through an early stopping policy. In the case of baseline experiments, the maximum number of epochs is set to 200 for all our executions. In MH experiments on Flicker8k and Flicker30k, we raise the maximum number of epochs to 400 as we observed results kept improving after 200 epochs.

On all our experiments (for both the FC7 and the FNE variants) the batch size is of 128 image-caption pairs. Within the same batch, every possible alternative image-caption pair is used as contrasting example (*i.e.*, we sum over 127 contrasting examples or we choose the worst example out of 127, depending on the loss used). In the GRUs we use gradient clipping with a threshold of 2. We use ADAM [18] as optimisation algorithm.

4.3.2. Caption processing

The caption sentences are word-tokenized using the Natural Language Toolkit (NLTK) for Python [3]. We did not remove punctuation marks as in [11, 39], and in contrast to [37]. Also, unlike some previous works [21, 39] we do not remove long sentences from the training split. We did not observe a significant impact on performance with this reduction of the text preprocessing. These observations are aligned with conclusions from [4], where simple tokenization works equally or better than more complex text preprocessing systems in general domain datasets. We hypothesise that the short nature of the texts combined with the availability of multiple text instances for each image helps the system to overcome sparsity issues.

The choice of the word embedding size and the number of GRUs has been analyzed to obtain a range of suitable parameters to test in the validation set. Previous contributions [11, 21, 37] set the word embedding dimensionality to 300. In our preliminary experiments, we tested word embedding dimensionalities of 300, 600, 1,024, 1,536, 2,048 and 3,072, finding that a higher dimensionality helps to obtain better results. We also found that very different dimensionalities between the word embedding and the multimodal embedding (*i.e.*, 300 - 2048) slow down the convergence speed during training. It seems reasonable that it may also affect the final performance. This could explain why higher word embedding dimensionalities help to obtain better results in this methodology. For word em-

beddings, a dimensionality between 1,024 and 1,536 performs competitively on all methods.

Similarly, we explored different multimodal embedding dimensionalities (*i.e.*, number of GRU units) of 300, 400, 500, 800, 1,024, 1,536, 2,048, 2,560, 3,072, 5,000 and 10,000 finding that dimensionalities between 1,024 and 2,048 give good results for all methods considered. In all experiments we tested at least 3 different dimensionalities presenting the results of the best performing one on the validation set. Previous methods usually adopt 1,024 as the dimensionality of the multimodal embedding space [11, 37], while others consider a much smaller dimensionality of 300 [21].

4.3.3. Image processing

For generating the image embedding we use the classical VGG16 CNN architecture [33] pretrained for ImageNet [31] as source model. This architecture is composed by 16 convolutional layers combined with pooling layers, followed by two fully connected layers and a final softmax output layer. Using only the activations of the last fully connected layer before the softmax (`fc7`), the dimensionality of the image embedding is 4,096. When using the FNE, features from different layers are combined in an image embedding space of 12,416 dimensions.

To obtain a better representation of the image, the full network embedding resizes the image to 256x256 pixels and extracts 5 crops of 224x224 pixels (one from each corner and the center). Mirroring these 5 crops horizontally we obtain a total of 10 crops which are processed through the CNN independently. The activations collected from each of these 10 crops are averaged to obtain a single representation of the image before further processing. For the baseline we use the same process before L2-normalization. Although a similar process is common for data augmentation, notice that we are not actually doing data augmentation since the number of training samples does not increase.

4.4. Evaluation metrics

To evaluate the image annotation and image retrieval tasks we use the following metrics:

- **Recall@K** (R@K) is the fraction of images for which a correct caption is ranked within the top-K retrieved results (and vice-versa for sentences). Results are provided for R@1, R@5 and R@10.
- **Median rank** (Med *r*) of the highest ranked ground truth result.

To obtain a comparable performance metric per model, we use the sum of the recalls on both tasks. This has been done before in [39] and in [11], the latter using only R@1 and R@10. We only use the score obtained on the validation set to select the best performing model for early stopping and hyper-parameter selection.

5. Results

Table 2 shows the results of the proposed full network embedding on the Flickr8K dataset, for both image annotation and image retrieval tasks. The top part of the table includes the current state-of-the-art (SotA) results as published. The second part summarises the results published by the original contributions this work is based on. Following parts contain the results produced by us for each of the models defined in Section 4.2. Each of these blocks comprises two pairs of results, the first pair corresponds to the results while using a configuration of hyper-parameters as close as possible to the original (*i.e.*, baseline or -bl), while the second pair corresponds to the results while using the best configuration we found for the FNE. Within each pair, the first experiment uses the FC7 embedding and the second uses the FNE, keeping all hyper-parameters unchanged. Best results for each pair are underlined. Tables 3 and 4 are analogous for the Flickr30K and MSCOCO datasets. Additional results of the UVS model [21] were made publicly available later on by the original authors [19]. We include these for the MSCOCO dataset, which was not evaluated in the original paper.

First, let us consider the effect of all modifications in the pipeline (detailed in sections 3 and 4.3) compared to our previous work [39]. In the first block of experiments, we can compare the results from [39] (FC7-SH-bl and FNE-SH-bl) with the ones obtained in this work for the same model (FC7-SH and FNE-SH). Notice that in FC7-SH-bl and FNE-SH-bl hyper-parameters were already optimized for FNE. We can see a substantial improvement in results obtained using both the FC7 and the FNE image embeddings. With an average increase in recall of 4.75% on MSCOCO, these results validate globally the improvements made in the pipeline and the exhaustive hyper-parameter fine-tuning.

Results obtained in this work for the original pipeline from Kiros *et al.*(FC7-SH) are now very close to the ones obtained by other studied methods (FC7-MH,

Table 2

Results obtained for the Flickr8K dataset. R@K is Recall@K (high is good). Med *r* is Median rank (low is good). Best results for each FC7 - FNE comparison are shown in underline. Best results for SotA and our experiments are shown in **bold**

Model		Image Annotation				Image Retrieval			
		R@1	R@5	R@10	Med *r*	R@1	R@5	R@10	Med *r*
FV	[22]	21.2	50.0	64.8	5	**31.0**	**59.3**	**73.7**	**4**
m-CNN	[26]	24.8	53.7	67.1	5	20.3	47.6	61.7	5
Bi-LSTM	[40]	29.3	58.2	69.6	**3**	19.7	47.0	60.6	5
W2VV	[9]	33.6	62.0	**75.3**	**3**	-	-	-	-
2WayNet	[10]	**43.4**	**63.2**	-	-	29.3	49.7	-	-
UVS	[21]	18.0	40.9	55.0	8	12.5	37.0	51.5	10
FC7-SH-bl[a]		21.0	45.7	60.4	7	14.0	35.8	48.6	11
FNE-SH-bl[a]		23.3	50.8	66.8	5	15.0	38.2	51.6	10
FC7-SH		22.4	49.8	62.9	6	16.6	41.2	54.3	8
FNE-SH		25.0	50.8	64.3	5	18.6	44.9	58.0	7
FC7-MH-bl [b]		22.6	48.6	61.9	6	17.7	42.7	54.9	8
FNE-MH-bl [b]		24.2	52.0	65.2	5	19.4	44.3	57.3	7
FC7-MH		23.0	49.0	63.3	6	18.5	43.2	56.1	8
FNE-MH		**27.3**	**56.8**	**69.3**	**4**	**21.2**	**47.1**	**59.7**	**6**
FC7-SOE-bl		20.6	45.4	58.0	7	15.4	38.8	52.7	9
FNE-SOE-bl		21.5	48.5	60.7	6	16.2	40.7	53.8	9
FC7-SOE		21.2	48.1	61.7	6	17.8	43.6	56.5	8
FNE-SOE		24.0	52.4	63.9	5	18.7	44.2	57.7	7
FC7-MOE-bl		22.6	48.2	62.3	6	16.9	41.5	54.2	9
FNE-MOE-bl[b]		0.1	0.3	0.4	2,476	0.1	0.5	1.0	501
FC7-MOE		21.5	46.1	60.0	7	15.6	39.0	51.9	9
FNE-MOE		25.5	55.5	67.8	**4**	18.7	44.4	58.4	7
FC7-PH		22.9	48.8	62.5	6	17.1	41.7	54.6	8
FNE-PH		26.3	55.7	68.5	**4**	20.5	45.8	58.1	7
FC7-POE		21.0	48.3	62.0	6	16.9	41.7	55.3	8
FNE-POE		26.2	53.6	65.8	5	19.7	45.6	58.4	7

[a] Results from [39]. [b] Trained for 400 epochs.

FC7-SOE and FC7-MOE) dimming the benefits of the proposed variants. In Table 4, we can easily compare the results claimed in the original papers [11, 21, 37] with the ones obtained under equal conditions (notice that not all methods were tested on Flickr datasets in original works). The most explicit differences are in recall@1 for both image annotation and image retrieval. For instance, VSE++ [11] obtains 21.2% and 21.0% increments over UVS [21], while the increments of our analogous versions (FC7-MH and FC7-SH) are now of 0.6% and 0.5% respectively. We hypothesise that most of the previously reported increment was due to different dataset sizes, CNN architectures and hyperparameter fine tuning; factors that we set equal for all methods.

These results highlight the difficulty to perform a consistent comparison between different multimodal approaches since different authors make different choices in the settings of their experiments (and sometimes fail to detail them thoroughly). Notably, important differences arise depending on the data used for training and testing, specially when experimenting with the MSCOCO dataset as we have seen in Section 4.1. Similarly, data augmentation techniques, a standard approach in most SotA methods, can give a boost to performance. In our experiments, we did our best to avoid such differences or to specify them entirely when they are unavoidable. In this context, the results we provide are as comparable as possible. It is essential to keep in mind all these considerations, when comparing the results we report with the ones from other publications.

Comparing the results of the family of methods based on [21] with the state of the art we see that their relative performance increases with dataset size (larger datasets lead to more competitive performances

Table 3

Results obtained for the Flickr30K dataset. R@K is Recall@K (high is good). Med *r* is Median rank (low is good). Best results for each FC7 - FNE comparison are shown in underline. Best results for SotA and our experiments are shown in **bold**

Model		Image Annotation				Image Retrieval			
		R@1	R@5	R@10	Med *r*	R@1	R@5	R@10	Med *r*
FV	[22]	25.0	52.7	66.0	5	35.0	62.0	73.8	3
m-CNN	[26]	33.6	64.1	74.9	3	26.2	56.3	69.6	4
Bi-LSTM	[40]	28.1	53.1	64.2	4	19.6	43.8	55.8	7
W2VV	[9]	39.7	67.0	76.7	2	-	-	-	-
sm-LSTM[a]	[15]	42.4	67.5	79.9	2	28.2	57.0	68.4	4
2WayNet	[10]	49.8	67.5	-	-	36.0	55.6	-	-
DAN (VGG)	[29]	41.4	73.5	82.5	2	31.8	61.7	72.5	3
DAN (ResNet)	[29]	**55.0**	**81.8**	**89.0**	**1**	39.4	**69.2**	79.1	**2**
EN	[41]	43.2	71.6	79.8	-	31.7	61.3	72.4	-
UVS	[21]	23.0	50.7	62.9	5	16.8	42.0	56.5	8
VSE++(1C)	[11]	31.9	-	68.0	4	23.1	-	60.7	6
VSE++(ResNet)[b]	[11]	52.9	-	87.2	1	**39.6**	-	**79.5**	**2**
FC7-SH-bl[c]		30.4	58.0	69.5	4	18.9	44.6	57.0	7
FNE-SH-bl[c]		30.4	61.8	73.2	3	22.1	47.6	59.8	6
FC7-SH		32.4	60.9	72.6	3	24.1	51.1	64.1	5
FNE-SH		36.4	64.6	75.7	3	25.5	53.8	65.7	5
FC7-MH-bl [d]		29.5	59.9	70.8	4	23.0	48.9	60.4	6
FNE-MH-bl [d]		34.7	63.1	75.6	3	25.1	52.3	64.7	5
FC7-MH		33.6	59.4	69.3	3	23.6	50.0	61.8	5
FNE-MH		**37.7**	66.6	**78.6**	**2**	27.8	56.0	67.1	**4**
FC7-SOE-bl		31.6	60.0	72.4	3	24.0	52.1	64.1	5
FNE-SOE-bl		33.7	63.8	75.3	3	26.0	55.1	67.7	**4**
FC7-SOE		30.2	59.4	70.4	4	23.8	50.5	62.7	5
FNE-SOE		35.5	63.4	75.3	3	26.8	56.1	67.5	**4**
FC7-MOE-bl		31.1	56.2	67.8	4	20.8	47.1	58.2	7
FNE-MOE-bl		0.1	0.4	0.4	2,461	0.1	0.5	0.9	498
FC7-MOE		31.9	61.3	72.7	3	23.8	50.2	61.5	5
FNE-MOE		35.3	65.0	77.1	3	27.3	55.2	68.0	**4**
FC7-PH		31.8	60.1	73.6	3	24.0	51.8	63.3	5
FNE-PH		36.6	63.9	75.0	3	25.9	54.3	66.2	4
FC7-POE		31.4	60.9	72.3	3	24.5	51.3	63.7	5
FNE-POE		37.2	**67.1**	77.9	**2**	**28.1**	**57.8**	**69.1**	**4**

[a] Single model. [b] CNN fine-tuned. [c] Results from [39]. [d] Trained for 400 epochs.

of these methods). Since the methods tested are more data-driven (*i.e.*, fewer assumptions are made apriori), it is to be expected that they can benefit more from the increase of available data. These results are congruent with the ones in [11] where the experiments using more data obtain state-of-the-art results.

Now, let us focus on the differences between a model and the same model using the FNE image embedding. This is the most significant contribution of this paper, as it incorporates the FNE on several multimodal embedding pipelines. We can see through the tables of results that every method on every dataset obtains better results when using the FNE embedding when compared to the FC7. Moreover, even with the original hyper-parameter configuration (sub-optimal for FNE) the FNE obtains better results on all tests. The only exception is FNE-MOE-bl where training problems occur with the original configuration (in Section 6 we analyze this issue). Even in this case, results using an appropriate hyper-parameter selection are superior to those of the baseline (FC7-MOE-bl). Considering all the experiments on MSCOCO dataset (including baselines), the average increase in recall using the FNE embedding is 3.7%.

Table 4

Results obtained for the MSCOCO dataset. R@K is Recall@K (high is good). Med *r* is Median rank (low is good). Best results for each FC7 - FNE comparison are shown in underline. Best results for SotA and our experiments are shown in **bold**

Model		Image Annotation				Image Retrieval			
		R@1	R@5	R@10	Med *r*	R@1	R@5	R@10	Med *r*
FV	[22]	25.1	59.8	76.6	4	39.4	67.9	80.9	2
m-CNN	[26]	42.8	73.1	84.1	2	32.6	68.6	82.8	3
sm-LSTM [a]	[15]	52.4	81.7	90.8	1	38.6	73.4	84.6	2
2WayNet	[10]	55.8	75.2	-	-	39.7	63.3	-	-
EN	[41]	54.9	**84.0**	92.2	-	43.3	**76.4**	87.5	-
UVS [b]	[21]	43.4	75.7	85.8	2	31.0	66.7	79.9	3
Order [c]	[37]	46.7	-	88.9	2	37.9	-	85.9	2
VSE++(1C)	[11]	43.6	-	85.8	2	33.7	-	81.0	3
VSE++(ResNet)[c,d]	[11]	**64.6**	-	**95.7**	**1**	**52.0**	-	**92.0**	**1**
Order++[c]	[11]	53.0	-	91.9	1	42.3	-	88.1	2
FC7-SH-bl[e]		41.2	72.8	85.1	2	26.2	58.6	73.9	4
FNE-SH-bl[e]		47.3	76.8	85.8	2	31.4	65.4	78.7	3
FC7-SH		44.0	77.0	86.0	2	33.6	68.8	81.1	3
FNE-SH		**50.6**	80.0	88.4	**1**	36.7	71.3	82.7	**2**
FC7-MH-bl		43.8	74.7	84.5	2	32.8	67.5	80.5	3
FNE-MH-bl		49.6	78.9	89.5	2	37.5	72.1	83.6	**2**
FC7-MH		44.6	75.8	85.7	2	34.1	68.2	80.7	3
FNE-MH		50.2	80.5	**90.5**	**1**	37.2	71.9	83.0	2
FC7-SOE-bl		41.5	74.4	86.0	2	33.8	69.0	82.6	3
FNE-SOE-bl		47.1	78.5	89.6	2	36.8	71.6	84.2	**2**
FC7-SOE		44.3	74.8	84.4	2	34.9	69.2	81.9	3
FNE-SOE		46.7	79.8	88.9	2	36.4	72.8	84.7	**2**
FC7-MOE-bl		40.7	75.3	85.9	2	32.2	66.4	78.3	3
FNE-MOE-bl		0.1	0.3	0.4	2,472	0.1	0.5	0.9	499
FC7-MOE		43.9	75.4	84.9	2	34.2	68.0	81.2	3
FNE-MOE		47.1	79.6	88.3	2	36.6	71.7	83.3	**2**
FC7-PH		45.3	75.0	85.5	2	33.8	68.4	81.0	3
FNE-PH		**50.6**	80.0	88.4	**1**	36.7	71.3	82.7	**2**
FC7-POE		45.6	75.9	86.6	2	35.2	69.7	83.1	**2**
FNE-POE		48.2	**81.5**	89.7	2	**38.8**	**73.5**	**85.0**	**2**

[a] Single model. [b] Results provided on [19]. [c] Extra training data from validation set.
[d] CNN fine-tuned. [e] Results from [39].

Considering the methods tested in our consistent experimental setup, we see that FNE-MH tend to obtain the best results on image annotation while FNE-POE is usually superior in image retrieval tasks. With these results, we can not consider one method preferable to the other except in the smallest Flicker8K dataset, where FNE-MH is superior. In any case, the performance differences between the best versions of each method remain lower than the impact of the FNE. For instance, in the experiments on MSCOCO, the recall gap between the best and the worst method (for each task separately) is on average 2.1%.

Finally, we observe that the proposed methodology of curriculum learning increases the already good performance of the original FC7-MOE [11] and the FNE-MOE 1.7% on average at MSCOCO. On the other hand, on methods based on the cosine similarity $\mathcal{S}_{COS}$, the second training step (using max loss $\mathcal{L}_M$) adds minimal improvement on the sum loss $\mathcal{L}_S$ results. Final results of FC7-PH and FNE-PH are in general inferior to those achieved by single training using max loss $\mathcal{L}_M$ (FC7-MH, FNE-MH).

Table 5
Hyper-parameter configuration and results for the experiments on MOE training behaviour. Success indicates the number of times that experiment succeeded in starting training (*i.e.*, score > 10) over total repetitions

Model	L.rate	Margin	Abs. val.	Success
FC7-MOE-bl	0.0002	0.2	✗	5/5
FC7-MOE-bl-abs	0.0002	0.2	✓	0/5
FC7-MOE-abs	0.0001	0.05	✓	0/5
FNE-MOE-bl	0.0002	0.2	✗	0/5
FNE-MOE-bl-abs	0.0002	0.2	✓	0/5
FNE-MOE-abs	0.0001	0.05	✓	4/5

6. Experiments on MOE training behaviour

When training models using the maximum order embedding (MOE and MOE-bl), we observed instability issues. For some configurations of hyper-parameters, the model does not start learning, even after extending the number of epochs significantly. To obtain some insights on that behaviour, we trained the same model five times with different random initialisations. The configurations tested are shown in Table 5. The combinations of learning rate, margin and absolute value are taken from the original works of [11, 37].

The rest of the hyper-parameters are kept the same for all experiments. The dimensionality of the word embedding is 300, and the multimodal embedding has 1,024 dimensions. The maximum number of epochs is 200. We run all the tests on Flickr8K to minimise computational cost, although we observed this behaviour in Flickr30K and MSCOCO too.

To evaluate these experiments, we count the number of times the algorithm succeeded in starting training. We consider it does not train if validation and test scores are below 10 (regular scores are higher than 200). The results obtained are shown in Table 5.

Results, quite surprisingly, do not point to a single variable as the cause of the problem. For the FC7 embedding, it did not train when absolute value was used, independently of the learning rate and margin. The experiment with the same configuration that worked well with FC7 does not train with FNE. On the other hand, the original configuration from [37] (but using max loss) successfully trained on FNE embedding, but this behaviour is not entirely robust since it failed once.

These experiments show that the instability of the training does not come from the choice of embedding, but instead on the hyper-parameter selection and parameter initialisation. While these experiments help to shed light on the problem, further work is required to completely understand the cause.

The proposed curriculum learning methodology (see Section 3.5) effectively solved this problem in all our experiments, as it initialises the network using the more robust sum loss. None of the experiments we did using the proposed curriculum learning methodology for different hyper-parameters configurations failed to start training.

7. Conclusions

For the multimodal pipeline of Kiros *et al.*[21] and the other methods based on it [11, 37], using the FNE results in consistently higher performances than using a one-layer image embedding. These results suggest that the visual representation provided by the FNE is superior to the current standard for the construction of most multimodal embeddings. In fact, the impact FNE has on performance is significantly superior to the improvement resultant of combining the main contributions from [37] and [11]. These results confirm our initial hypothesis that the richer and discrete representation obtained with FNE is more convenient for the construction of multimodal embeddings than the widely used single-layer real-valued embeddings.

The results of our comparative study of the different variants from [11, 21, 37] pointed up the need of properly assessing the sources of empirical gains. We consider it is a key aspect of research that should be further encouraged. We hope that our experimental study can help other researchers with design decisions from the text pre-processing to the loss choice including ranges of optimal dimensionalities and other hyper-parameters.

Another issue we tackled was the instability of MOE models. Depending on the random initialization of the weights, the same model may start learning or not. Our experiments showed that the combination of hyper-parameters also plays a role in these difficulties. However further study is required to get a real insight into the mechanisms causing this problem. In any case, the proposed curriculum learning method of pre-training using a sum of losses effectively alleviates this problem while increasing performance.

When compared to the current state of the art, the results obtained from the studied variants using FNE are below the results reported through other methods. This difference is often the result of using a more substantial amount of training data. Indeed, results given in

[11] indicate that models based on the pipeline of [21] can obtain state-of-the-art results when using enough data.

Finally, let us remark that the FNE is straightforward compatible with most multimodal pipelines based on CNN embeddings. The constant improvement in the results observed here for the variants proposed by [11, 21, 37] suggest that other methods can also boost its performance incorporating the FNE. These results also encourage us to consider the modifications required to be able to introduce attention mechanisms (*e.g.*, DAN) in our methodology in a future work.

Acknowledgements

This work is partially supported by the Joint Study Agreement no. W156463 under the IBM/BSC Deep Learning Center agreement, by the Spanish Government through Programa Severo Ochoa (SEV-2015-0493), by the Spanish Ministry of Science and Technology through TIN2015-65316-P project and by the Generalitat de Catalunya (contracts 2014-SGR-1051), and by the Core Research for Evolutional Science and Technology (CREST) program of Japan Science and Technology Agency (JST).

References

[1] H. Azizpour, A.S. Razavian, J. Sullivan, A. Maki and S. Carlsson, Factors of transferability for a generic convnet representation, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **38**(9) (2016), 1790–1802. doi:10.1109/TPAMI.2015.2500224.

[2] Y. Bengio, J. Louradour, R. Collobert and J. Weston, Curriculum learning, in: *Proceedings of the 26th annual international conference on machine learning*, ACM, 2009, pp. 41–48. doi:10.1145/1553374.

[3] S. Bird, NLTK: the natural language toolkit, in: *Proceedings of the COLING/ACL on Interactive presentation sessions*, Association for Computational Linguistics, 2006, pp. 69–72.

[4] J. Camacho-Collados and M.T. Pilehvar, On the Role of Text Preprocessing in Neural Network Architectures: An Evaluation Study on Text Categorization and Sentiment Analysis, in: *EMNLP 2018 Workshop: Analyzing and interpreting neural networks for NLP, arXiv preprint arXiv:1707.01780*, 2018.

[5] K. Cho, B. van Merriënboer, D. Bahdanau and Y. Bengio, On the Properties of Neural Machine Translation: Encoder–Decoder Approaches, *Syntax, Semantics and Structure in Statistical Translation* (2014), 103.

[6] K. Church, Word2Vec, *Natural Language Engineering* **23**(1) (2017), 155–162. doi:10.1017/S1351324916000334.

[7] S. Dill, N. Eiron, D. Gibson, D. Gruhl, R. Guha, A. Jhingran, T. Kanungo, S. Rajagopalan, A. Tomkins, J.A. Tomlin et al., SemTag and Seeker: Bootstrapping the semantic web via automated semantic annotation, in: *Proceedings of the 12th international conference on World Wide Web*, ACM, 2003, pp. 178–186. doi:10.1145/775152.775178.

[8] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng and T. Darrell, DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition., in: *Icml*, Vol. 32, 2014, pp. 647–655.

[9] J. Dong, X. Li and C.G. Snoek, Word2VisualVec: Cross-Media Retrieval by Visual Feature Prediction, *CoRR* **abs/1604.06838** (2016).

[10] A. Eisenschtat and L. Wolf, Linking Image and Text with 2-Way Nets, in: *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, IEEE, 2017, pp. 1855–1865. doi:10.1109/CVPR.2017.201.

[11] F. Faghri, D.J. Fleet, J.R. Kiros and S. Fidler, VSE++: Improving Visual-Semantic Embeddings with Hard Negatives, *arXiv preprint arXiv:1707.05612* (2017).

[12] D. Garcia-Gasulla, A. Vilalta, F. Parés, J. Moreno, E. Ayguadé, J. Labarta, U. Cortés and T. Suzumura, An Out-of-the-box Full-network Embedding for Convolutional Neural Networks, *arXiv preprint arXiv:1705.07706* (2017).

[13] D. Garcia-Gasulla, F. Parés, A. Vilalta, J. Moreno, E. Ayguadé, J. Labarta, U. Cortés and T. Suzumura, On the Behavior of Convolutional Nets for Feature Extraction, *Journal of Artificial Intelligence Research* **61** (2018), 563–592. doi:10.1613/jair.5756.

[14] 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** (2013), 853–899. doi:10.1613/jair.3994.

[15] Y. Huang, W. Wang and L. Wang, Instance-Aware Image and Sentence Matching with Selective Multimodal LSTM, in: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Vol. 00, 2018, pp. 7254–7262, ISSN 1063-6919. doi:10.1109/CVPR.2017.767.

[16] A. Karpathy and L. Fei-Fei, Deep visual-semantic alignments for generating image descriptions, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3128–3137. doi:10.1109/TPAMI.2016.2598339.

[17] A. Karpathy, A. Joulin and F.F.F. Li, Deep fragment embeddings for bidirectional image sentence mapping, in: *Advances in neural information processing systems*, 2014, pp. 1889–1897.

[18] D. Kingma and J. Ba, Adam: A method for stochastic optimization, International Conference on Learning Representations (ICLR 2015), 2015.

[19] R. Kiros, visual-semantic-embedding. https://github.com/ryankiros/visual-semantic-embedding.

[20] R. Kiros, R. Salakhutdinov and R. Zemel, Multimodal neural language models, in: *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 2014a, pp. 595–603.

[21] R. Kiros, R. Salakhutdinov and R.S. Zemel, Unifying visual-semantic embeddings with multimodal neural language models, *arXiv preprint arXiv:1411.2539* (2014b), Advances in neural information processing systems 2014 deep learning workshop.

[22] B. Klein, G. Lev, G. Sadeh and L. Wolf, Associating neural word embeddings with deep image representations using fisher vectors, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4437–4446. doi:10.1109/CVPR.2015.7299073.

[23] Y. LeCun, Y. Bengio and G. Hinton, Deep learning, *nature* **521**(7553) (2015), 436. doi:10.1038/nature14539.

[24] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár and C.L. Zitnick, Microsoft COCO: Common Objects in Context, in: *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele and T. Tuytelaars, eds, Springer International Publishing, Cham, 2014, pp. 740–755. ISBN 978-3-319-10602-1. doi:10.1007/978-3-319-10602-1_48.

[25] Z.C. Lipton and J. Steinhardt, Troubling Trends in Machine Learning Scholarship, *arXiv preprint arXiv:1807.03341* (2018), International Conference on Machine Learning (ICML 2018).

[26] L. Ma, Z. Lu, L. Shang and H. Li, Multimodal convolutional neural networks for matching image and sentence, in: *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2623–2631. doi:10.1109/ICCV.2015.301.

[27] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado and J. Dean, Distributed representations of words and phrases and their compositionality, in: *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[28] A. Mousavian and J. Kosecka, Deep convolutional features for image based retrieval and scene categorization, *arXiv preprint arXiv:1509.06033* (2015).

[29] H. Nam, J.-W. Ha and J. Kim, Dual Attention Networks for Multimodal Reasoning and Matching, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 299–307. doi:10.1109/CVPR.2017.232.

[30] C. Rashtchian, P. Young, M. Hodosh and J. Hockenmaier, Collecting image annotations using Amazon's Mechanical Turk, in: *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, Association for Computational Linguistics, 2010, pp. 139–147.

[31] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein et al., Imagenet large scale visual recognition challenge, *International Journal of Computer Vision* **115**(3) (2015), 211–252. doi:10.1007/s11263-015-0816-y.

[32] A. Sharif Razavian, H. Azizpour, J. Sullivan and S. Carlsson, CNN features off-the-shelf: an astounding baseline for recognition, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2014, pp. 806–813. doi:10.1109/CVPRW.2014.131.

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

[34] Q. Sun, S. Lee and D. Batra, Bidirectional Beam Search: Forward-Backward Inference in Neural Sequence Models for Fill-in-the-Blank Image Captioning, in: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 7215–7223, ISSN 1063-6919. doi:10.1109/CVPR.2017.763.

[35] I. Sutskever, O. Vinyals and Q.V. Le, Sequence to sequence learning with neural networks, in: *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[36] Theano Development Team, Theano: A Python framework for fast computation of mathematical expressions, *arXiv preprint arXiv:1605.02688* (2016). http://arxiv.org/abs/1605.02688.

[37] I. Vendrov, R. Kiros, S. Fidler and R. Urtasun, Order-embeddings of images and language, *arXiv preprint arXiv:1511.06361* (2015), International Conference on Learning Representations (ICLR 2015).

[38] A. Vilalta, Full network multimodal embeddings. https://github.com/armandvilalta/Full-network-multimodal-embeddings.

[39] A. Vilalta, D. Garcia-Gasulla, F. Parés, J. Moreno, E. Ayguadé, J. Labarta, U. Cortés and T. Suzumura, Full-Network Embedding in a Multimodal Embedding Pipeline, in: *Proceedings of the 2nd Workshop on Semantic Deep Learning (SemDeep-2)*, 2017, pp. 24–32.

[40] C. Wang, H. Yang and C. Meinel, Image Captioning with Deep Bidirectional LSTMs and Multi-Task Learning, *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* **14**(2s) (2018a), 40. doi:10.1145/3115432.

[41] L. Wang, Y. Li, J. Huang and S. Lazebnik, Learning two-branch neural networks for image-text matching tasks, *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2018b). doi:10.1109/TPAMI.2018.2797921.

[42] 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** (2014), 67–78.