Self-Supervised Learning from Web Data for Multimodal Retrieval

Raul Gomez[a,b,*], Lluis Gomez[b], Jaume Gibert[a], Dimosthenis Karatzas[b]

[a]*Eurecat, Centre Tecnològic de Catalunya, Unitat de Tecnologies Audiovisuals, Barcelona, Spain*
[b]*Computer Vision Center, Universitat Autònoma de Barcelona, Barcelona, Spain*

Abstract

Self-Supervised learning from multimodal image and text data allows deep neural networks to learn powerful features with no need of human annotated data. Web and Social Media platforms provide a virtually unlimited amount of this multimodal data. In this work we propose to exploit this free available data to learn a multimodal image and text embedding, aiming to leverage the semantic knowledge learnt in the text domain and transfer it to a visual model for semantic image retrieval. We demonstrate that the proposed pipeline can learn from images with associated text without supervision and analyze the semantic structure of the learnt joint image and text embedding space. We perform a thorough analysis and performance comparison of five different state of the art text embeddings in three different benchmarks. We show that the embeddings learnt with Web and Social Media data have competitive performances over supervised methods in the text based image retrieval task, and we clearly outperform state of the art in the MIRFlickr dataset when training in the target data. Further, we demonstrate how semantic multimodal image retrieval can be performed using the learnt embeddings, going beyond classical instance-level retrieval problems. Finally, we present a new dataset, InstaCities1M, composed by Instagram images and their associated texts that can be used for fair comparison of image-text embeddings.

Keywords: self-supervised learning, webly supervised learning, text embeddings, multimodal retrieval, multimodal embedding

1. Introduction

1.1. Annotating Data: A Bottleneck for Training Deep Neural Networks

Large annotated datasets, powerful hardware and deep learning techniques are allowing to get outstanding machine learning results. Not only in traditional classification problems but also in more challenging tasks such as image captioning or language translation. Deep neural networks allow building pipelines that can learn patterns from any kind of data with impressive results.

Deep learning has two strong requirements: Computation power and tons of data. The computation power requirement is fulfilled by GPUs and other AI specialized hardware, such as TPUs.

*Corresponding author
Email addresses: raul.gomez@cvc.uab.es (Raul Gomez), lgomez@cvc.uab.es (Lluis Gomez), jaume.gibert@eurecat.org (Jaume Gibert), dimos@cvc.uab.es (Dimosthenis Karatzas)

Moreover, the hardware power is evolving fast without an apparent roof together with deep learning algorithms requirements. The story with the data requirement is different. Despite the existence of large-scale annotated datasets such as ImageNet [1], COCO [2] or Places [3], the lack of data limits the application of deep learning to specific problems where it is difficult or economically non-viable to get proper annotations. Although there exist some tools to facilitate human data annotation, such as Amazon Mechanical Turk[1], annotating the tons of data required to train supervised deep learning models is a very expensive and manual task, whose efficiency cannot evolve over time.

1.2. Alternatives to Annotated Data

A common strategy to overcome the lack of annotated data is to first train models in generic datasets, as ImageNet, and then fine-tune them to other tasks using smaller, specific datasets [4]. But still we depend on the existence of annotated data to train our models. Another strategy to overcome the insufficiency of data is to use computer graphics techniques to generate artificial data inexpensively. However, while synthetic data has proven to be a valuable source of training data for many applications such as pedestrian detection [5], image semantic segmentation [6] and scene text detection and recognition [7, 8], nowadays it is still not easy to generate realistic complex images for some tasks.

An alternative to this strategies and a solution to overcome the annotated data requirements of supervised deep learning techniques are not fully supervised techniques. Among them, self-supervised learning exploits multimodal data to learn relations between two or more data modalities using paired instances. Web and Social Media offer an immense amount of images accompanied with other information such as the image title, description or date. This data is noisy and unstructured but it is free and nearly unlimited. We mentioned that data annotation efficiency does not improve with time. As a contrast, the amount of available multi-modal data in the Web does. Designing algorithms to learn from Web data is an interesting research area as it would disconnect the deep learning evolution from the scaling of human-annotated datasets, given the enormous amount of existing Web and Social Media data. We call this scenario self-supervised learning because it consists in exploiting relations between different modalities (in this case images and text) of multimodal data as supervision.

1.3. Exploiting Multimodal Web Data

Lately, Web data has been used to build classification datasets, such as in the WebVision Challenge [9] and in this Facebook work [10]. In these works, to build a classification dataset, queries are made to search engines using class names and the retrieved images are labeled with the querying class. In such a configuration the learning is limited to some pre-established classes, thus it could not generalize to new classes. While working with image labels is very convenient for training traditional visual models, the semantics in such a discrete space are very limited in comparison with the richness of human language expressiveness when describing an image. Instead we define here a scenario where, by exploiting distributional semantics in a given text corpus, we can

[1] `https://www.mturk.com`

learn from every word associated to an image. As illustrated in Figure 1, by leveraging the richer semantics encoded in the learnt embedding space, we can infer previously unseen concepts even though they might not be explicitly present in the training set.

The noisy and unstructured text associated to Web images provides information about the image content that we can use to learn visual features. A strategy to do that is to embed the multimodal data (images and text) in the same vectorial space. In this work we represent text using five different state of the art methods and eventually embed images in the learnt semantic space by means of a regression CNN. We compare the performance of the different text space configurations under a text based image retrieval task.

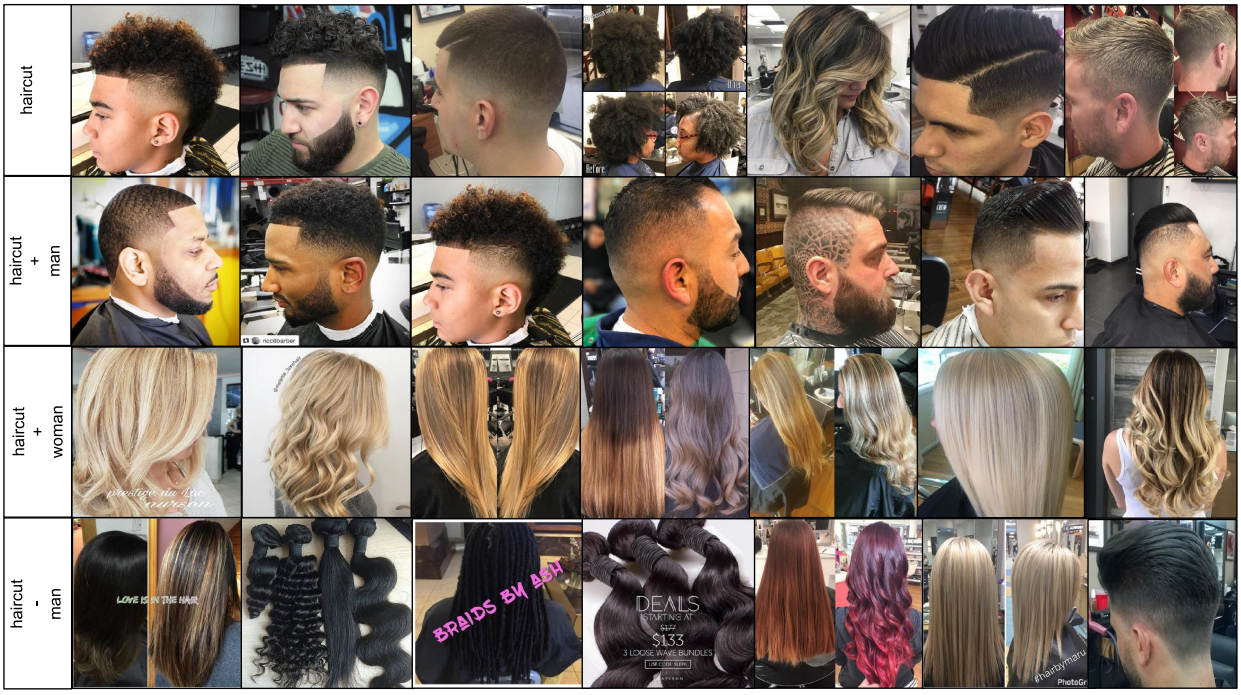

Figure 1: Top-ranked results of combined text queries by our semantic image retrieval model. The learnt joint image-text embedding permits to learn a rich semantic manifold even for previously unseen concepts even though they might not be explicitly present in the training set.

2. Related Work

Multimodal image and text embeddings have been lately a very active research area. The possibilities of learning together from different kinds of data have motivated this field of study, where both general and applied research has been done. DeViSE [11] proposes a pipeline that, instead of learning to predict ImageNet classes, learns to infer the Word2Vec [12] representations of their labels. The result is a model that makes semantically relevant predictions even when it makes errors, and generalizes to classes outside of its labeled training set. Gordo & Larlus [13] use captions associated to images to learn a common embedding space for images and text through which they perform semantic image retrieval. They use a *tf-idf* based BoW representation over the

image captions as a semantic similarity measure between images and they train a CNN to minimize a margin loss based on the distances of triplets of query-similar-dissimilar images. Gomez, Patel *et al.* [14, 15] use LDA [16] to extract topic probabilities from a bunch of Wikipedia articles and train a CNN to embed their associated images in the same topic space. Wang *et al.* [17] propose a method to learn a joint embedding of images and text for image-to-text and text-to-image retrieval, by training a neural net to embed in the same space Word2Vec [12] text representations and CNN extracted features.

Other than semantic retrieval, joint image-text embeddings have also been used in more specific applications. Patel *et al.* [18] use LDA [16] to learn a joint image-text embedding and generate contextualized lexicons for images using only visual information. Gordo *et al.* [19] embed word images in a semantic space relying in the graph taxonomy provided by WordNet [20] to perform text recognition. In a more specific application, Salvador *et al.* [21] propose a joint embedding of food images and their recipes to identify ingredients, using Word2Vec [12] and LSTM representations to encode ingredient names and cooking instructions and a CNN to extract visual features from the associated images. Exploiting Instagram publications related to #Barcelona, Gomez *et al.* [22] learn relations between words, images and Barcelona neighbourhoods to study which words and visual features tourist and locals relate with each neighbourhood.

The robustness against noisy data has also been addressed by the community, though usually in an implicit way. Patrini *et al.* [23] address the problem of training a deep neural network with label noise with a loss correction approach and Xiau *et al.* [24] propose a method to train a network with a limited number of clean labels and millions of noisy labels. Fu *et al.* [25] propose an image tagging method robust to noisy training data and Xu *et al.* [26] address social image tagging correction and completion. Zhang *et al.* [27] show how label noise affects the CNN training process and its generalization error.

2.1. Contributions

The work presented here brings in a performance comparison between five state of the art text embeddings in self-supervised learning, showing results in three different datasets. Furthermore it proves that self-supervised multimodal learning can be applied to Web and Social Media data achieving competitive results in text-based image retrieval compared to pipelines trained with human annotated data. Finally, a new dataset formed by Instagram images and their associated text is presented: InstaCities1M.

3. Multimodal Text-Image Embedding

One of the objectives of this work is to serve as a fair comparative of different text embeddings methods when learning from Web and Social Media data. Therefore we design a pipeline to test the different methods under the same conditions, where the text embedding is a module that can be replaced by any text representation.

The proposed pipeline is as follows: First, we train the text embedding model on a dataset composed by pairs of images and correlated texts (I, x). Second, we use the text embedding model to generate vectorial representations of those texts. Given a text instance x, we denote its embedding by $\phi(x) \in \mathbb{R}^D$. Third, we train a CNN to regress those text embeddings directly from

the correlated images. Given an image I, its representation in the embedding space is denoted by $\psi(I) \in \mathbb{R}^D$. Thereby the CNN learns to embed images in the vectorial space defined by the text embedding model. The trained CNN model is used to generate visual embeddings for the test set images. Figure 2 shows a diagram of the visual embedding training pipeline and the retrieval procedure.

In the image retrieval stage the vectorial representation in the joint text-image space of the querying text is computed using the text embedding model. Image queries can also be handled by using the visual embedding model instead of the text embedding model to generate the query representation. Furthermore, we can generate complex queries combining different query representations applying algebra in the joint text-image space. To retrieve the most semantically similar image I_R to a query x_q, we compute the cosine similarity of its vectorial representation $\phi(x_q)$ with the visual embeddings of the test set images $\psi(I_T)$, and retrieve the nearest image in the joint text-image space:

$$\underset{I_T \in \text{Test}}{\arg\min} \frac{\langle \phi(x_q), \psi(I_T) \rangle}{||\phi(x_q)|| \cdot ||\psi(I_T)||}. \tag{1}$$

State of the art text embedding methods trained on large text corpus are very good generating representations of text in a vector space where semantically similar concepts fall close to each other. The proposed pipeline leverages the semantic structure of these text embedding spaces training a visual embedding model that generates vectorial representations of images in the same space, mapping semantically similar images close to each other, and also close to texts correlated to the image content. Note that the proposed joint text-image embedding can be extended to other tasks besides image retrieval, such as image annotation, tagging or captioning.

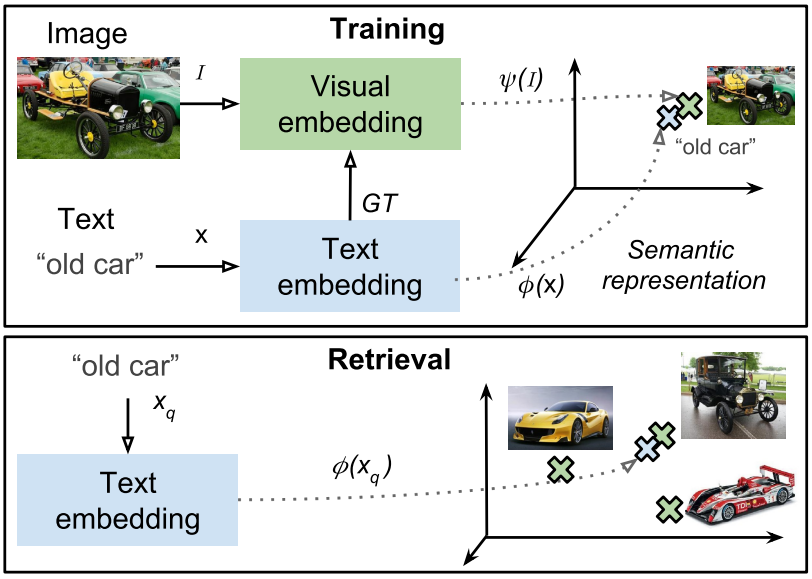

Figure 2: Pipeline of the visual embedding model training and the image retrieval by text.

A CNN is trained to regress text embeddings from the correlated images minimizing a sigmoid cross-entropy loss. This loss is used to minimize distances between the text and image

embeddings. Let $\{(I_n, x_n)\}_{n=1:N}$ be a batch of image-text pairs. If $\sigma(\cdot)$ is the component-wise sigmoid function, we denote $p_n = \sigma(\phi(x_n))$ and $\hat{p}_n = \sigma(\psi(I_n))$. Note p_n, $\hat{p}_n \in \mathbb{R}^D$ where D is the dimensionality of the joint embedding space. Let the loss be:

$$L = -\frac{1}{ND} \sum_{n=1}^{N} \sum_{d=1}^{D} [\, p_{n_d} \log \hat{p}_{n_d} + (1 - p_{n_d}) \log(1 - \hat{p}_{n_d}) \,], \tag{2}$$

The GoogleNet architecture [28] is used, customizing the last layer to regress a vector of the same dimensionality as the text embedding. We train with a Stochastic Gradient Descent optimizer with a learning rate of $1e^{-3}$, multiplied by 0.1 every 100k iterations, and a momentum of 0.9. The batch size is set to 120 and random cropping and mirroring are used as online data augmentation. With these settings the CNN trainings converge after around 300K-500K iterations. We use the Caffe [29] framework and initialize with the ImageNet [1] trained model to make the training faster. Notice that, despite initializing with a model trained with human-annotated data, this does not denote a dependence on annotated data, since the resulting model can generalize to much more concepts than the ImageNet classes. We trained one model from scratch obtaining similar results, although more training iterations were needed. Cross Entropy Loss is not usually used for regression problems, where Mean Square Error loss is often used. We chose Cross Entropy Loss empirically, since it was the one providing an stable training and better performance. Although Cross Entropy Loss tends to be considered a loss for classification, it is also suitable for regression problems: despite this loss will not be zero when the regression solution matches the groundtruth, it will always be minimum compared to other solutions.

4. Text Embeddings

Text vectorization methods are diverse in terms of architecture and the text structure they are designed to deal with. Some methods are oriented to vectorize individual words and others to vectorize full texts or paragraphs. In this work we consider the top-performing text embeddings and test them in our pipeline to evaluate their performance when learning from Web and Social Media data. Here we explain briefly the main characteristics of each text embedding method used.

LDA [16]. Latent Dirichlet Allocation learns latent topics from a collection of text documents and maps words to a vector of probabilities of those topics. It can describe a document by assigning topic distributions to it, which in turn have word distributions assigned. An advantage of this method is that it gives interpretable topics.

GloVe [30]. It is a count-based model. It learns the vectors by essentially doing dimensionality reduction on the co-occurrence counts matrix. Training is performed on aggregated global word-word co-occurrence statistics from a corpus.

Word2Vec [12]. Learns representations for words based on their context using a single hidden layer feed-forward neural network. It has two variants: In the CBOW (Continuous Bag of Word) approach, the neural network is trained to predict a word given as input its surrounding context

(surrounding words). In the Skip-gram model, opposite to the CBOW model, the neural network is trained to predict a word context given that word as an input. In this work we use the most extended and efficient CBOW approach.

Doc2Vec [31]. Extends the Word2Vec idea to documents, being able to create a numeric representation for them, regardless of their length. Extending Word2Vec CBOW model, it adds another input vector to the input context, which is the paragraph identifier. When training the word vectors, the document vector is trained as well, and at the end it holds a numeric representation of the whole document. As with Word2Vec, in this work we use the CBOW approach.

FastText [32]. It is an extension of Word2Vec which treats each word as composed of character ngrams, learning representations for ngrams instead of words. The idea is to take into account and exploit the morphology of words. Each word is split in n-grams which are all inputted separately to the model, which can be trained using the CBOW or the skip-gram approach. The vector for each word is made of the sum of its character n grams, so it can generate embeddings for out of vocabulary words. By exploiting words morphology, FastText tries to generate better embeddings for rare words, assuming their character ngrams are shared with other words. It also allows to generate embeddings for out of vocabulary words. To train FastText we use the originaly proposed and most extended skigram approach.

To the best of our knowledge, this is the first time these text embeddings are trained from scratch on the same corpus and evaluated under the image retrieval by text task. We used Gensim[2] implementations of LDA, Word2Vec, FastText and Doc2Vec and the GloVe implementation by Maciej Kula[3]. While LDA and Doc2Vec can generate embeddings for documents, Word2Vec, GloVe and FastText only generate word embeddings. To get documents embeddings from these methods, we consider two standard strategies: First, computing the document embedding as the mean embedding of its words. Second, computing a *tf-idf* weighted mean of the words in the document. For all embeddings a dimensionality of 400 has been used. The value has been selected because is the one used in the Doc2Vec paper [31], which compares Doc2Vec with other text embedding methods, and it is enough to get optimum performances of Word2Vec, FastText and GloVe, as [12, 32, 30] show respectively. For LDA a dimensionality of 200 has also been considered.

5. Benchmarks

In this section we present the datasets used in this work and show some examples of their images and their associated text.

5.1. InstaCities1M

A dataset formed by Instagram images associated with one of the 10 most populated English speaking cities all over the world (in the images captions one of the names of these cities

[2] `https://radimrehurek.com/gensim`
[3] `https://github.com/maciejkula/glove-python`

appears). It contains 100K images for each city, which makes a total of 1M images, split in 800K training images, 50K validation images and 150K test images. The interest of this dataset is that is formed by recent Social Media data. The text associated with the images is the description and the hashtags written by the photo up-loaders, so it is the kind of free available data that would be very interesting to be able to learn from. Figure 3 shows some examples of InstaCities1M images and their associated text. The InstaCities1M dataset is available on `https://gombru.github.io/2018/08/01/InstaCities1M/`.

Figure 3: Examples of InstaCities1M dataset images.

5.2. WebVision

The Webvision dataset [33] contains more than 2.4 million images crawled from the Flickr Website and Google Images search. The same 1,000 concepts as the ILSVRC 2012 dataset [1] are used for querying images. The textual information accompanying those images (caption, user tags and description) is provided. The validation set, which is used as test in this work, contains 50K images. Figure 4 shows some examples of WebVision images and their associated text.

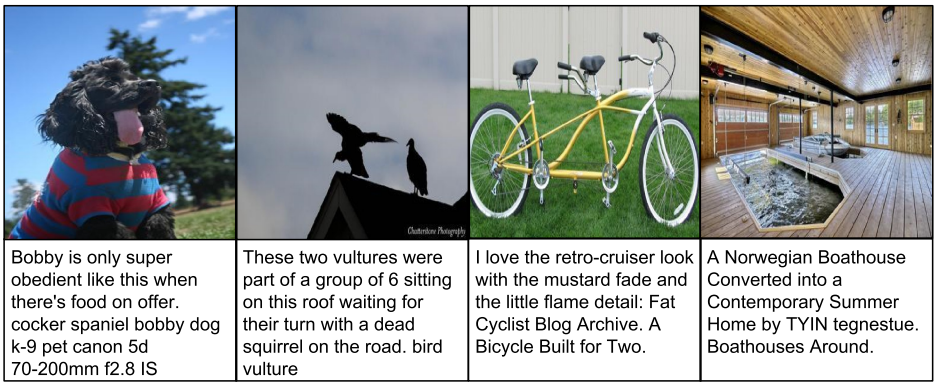

Figure 4: Examples of WebVision dataset images.

5.3. MIRFlickr

The MIRFlickr dataset [34] contains 25,000 images collected from Flickr, annotated using 24 predefined semantic concepts. 14 of those concepts are divided in two categories: 1) strong correlation concepts and 2) weak correlation concepts. The correlation between an image and a concept is strong if the concept appears in the image predominantly. For differentiation, we denote strong correlation concepts by a suffix "*". Finally, considering strong and weak concepts separately, we get 38 concepts in total. All images in the dataset are annotated by at least one of those concepts. Additionally, all images have associated tags collected from Flickr. Following the experimental protocol in [35, 36, 37, 38] tags that appear less than 20 times are first removed and then instances without tags or annotations are removed. Figure 5 shows some examples of MIRFlickr images and their associated text.

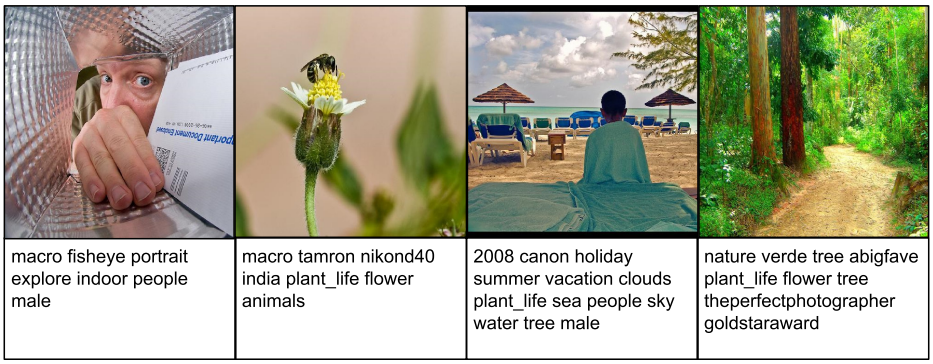

Figure 5: Examples of MIRFlickr dataset images.

6. Retrieval on InstaCities1M and WebVision Datasets

In this section we perform image retrieval experiments in the InstaCites1M and the WebVision datasets, comparing the performance of the different text embeddings in our pipeline. We analyze the performance of each text embedding, present an error analysis of our pipeline and show qualitative retrieval results of both image by text retrieval and image retrieval using multimodal queries.

6.1. Experiment Setup

To evaluate the learnt joint embeddings, we define a set of textual queries and check visually if the TOP-5 retrieved images contain the querying concept. We define 24 different queries. Half of them are single word queries and the other half two word queries. They have been selected to cover a wide area of semantic concepts that are usually present in Web and Social Media data. Both simple and complex queries are divided in four different categories: Urban, weather, food and people. Queries are listed in Table 1. For complex queries, only images containing both querying concepts are considered correct.

Table 1: Queries for the retrieval experiments on InstaCities1M and WebVision datasets.

	Simple	**Complex**
Urban	car, skyline, bike	yellow+car, skyline+night, bike+park
Weather	sunrise, snow, rain	sunrise+beach, snow+ski, rain+umbrella
Food	ice-cream, cake, pizza	ice-cream+beach, chocolate+cake, pizza+wine
People	woman, man, kid	woman+bag, man+boat, kid+dog

Table 2: Performance on InstaCities1M and WebVision. First column shows the mean P@5 for all the queries, second for the simple queries and third for complex queries.

Text embedding	**InstaCities1M**			**WebVision**		
Queries	All	S	C	All	S	C
LDA 200	0.40	0.73	0.07	0.11	0.18	0.03
LDA 400	0.37	0.68	0.05	0.14	0.18	0.10
Word2Vec mean	0.46	0.71	**0.20**	0.37	0.57	0.17
Word2Vec tf-idf	0.41	0.63	0.18	**0.41**	0.58	0.23
Doc2Vec	0.22	0.25	0.18	0.22	0.17	**0.27**
GloVe	0.41	0.72	0.10	0.36	**0.60**	0.12
GloVe tf-idf	**0.47**	**0.82**	0.12	0.39	0.57	0.22
FastText tf-idf	0.31	0.50	0.12	0.37	0.60	0.13

Table 3: Performance on transfer learning. First column shows the mean P@5 for all the queries, second for the simple queries and third for complex queries.

Text embedding	**Train: WebVision Test: InstaCities**			**Train: InstaCities Test: WebVision**		
Queries	All	S	C	All	S	C
LDA 200	0.14	0.25	0.03	0.33	0.55	0.12
LDA 400	0.17	0.25	0.08	0.24	0.39	0.10
Word2Vec mean	0.41	**0.63**	0.18	0.33	0.52	0.15
Word2Vec tf-idf	**0.42**	0.57	**0.27**	0.32	0.50	0.13
Doc2Vec	0.27	0.40	0.15	0.24	0.33	0.15
GloVe	0.36	0.58	0.15	0.29	0.53	0.05
GloVe tf-idf	0.39	0.57	0.22	**0.51**	**0.75**	**0.27**
FastText tf-idf	0.39	0.57	0.22	0.18	0.33	0.03

Table 4: Performance on InstaCities1M using GloVe tf-idf introducing noise by changing the indicated % of captions by random captions from the training set.

Experiment	**InstaCities1M**		
Queries	All	S	C
Without introduced noise	0.47	0.82	0.12
10% introduced noise	0.25	0.43	0.07
20% introduced noise	0.18	0.32	0.05
30% introduced noise	0.15	0.25	0.05

6.2. Results and Conclusions

Tables 2 and 3 show the mean Precision at 5 for InstaCities1M and WebVision datasets and transfer learning between those datasets. To compute transfer learning results, we train the model with one dataset and test with the other. Table 4 shows the mean precision at 5 for InstaCities1M with introduced additional noise and of a model trained with Mean Square Error loss. The noise is introduced by changing the indicated % of captions to random captions from the training set. Figures 1, 6 and 7 show the first retrieved images for some complex textual queries. Figure 7 also shows results for non-object queries, proving that our pipeline works beyond traditional instance-level retrieval. Figures 8 and 9 show that retrieval also works with multimodal queries combining an image and text.

For complex queries, where we demand two concepts to appear in the retrieved images, we obtain good results for those queries where the concepts tend to appear together. For instance, we generally retrieve correct images for "skyline + night" and for "bike + park", but we do not retrieve images for "dog + kid". When failing with this complex queries, usually images where only one of the two querying concepts appears are retrieved. Figure 10 shows that in some cases images

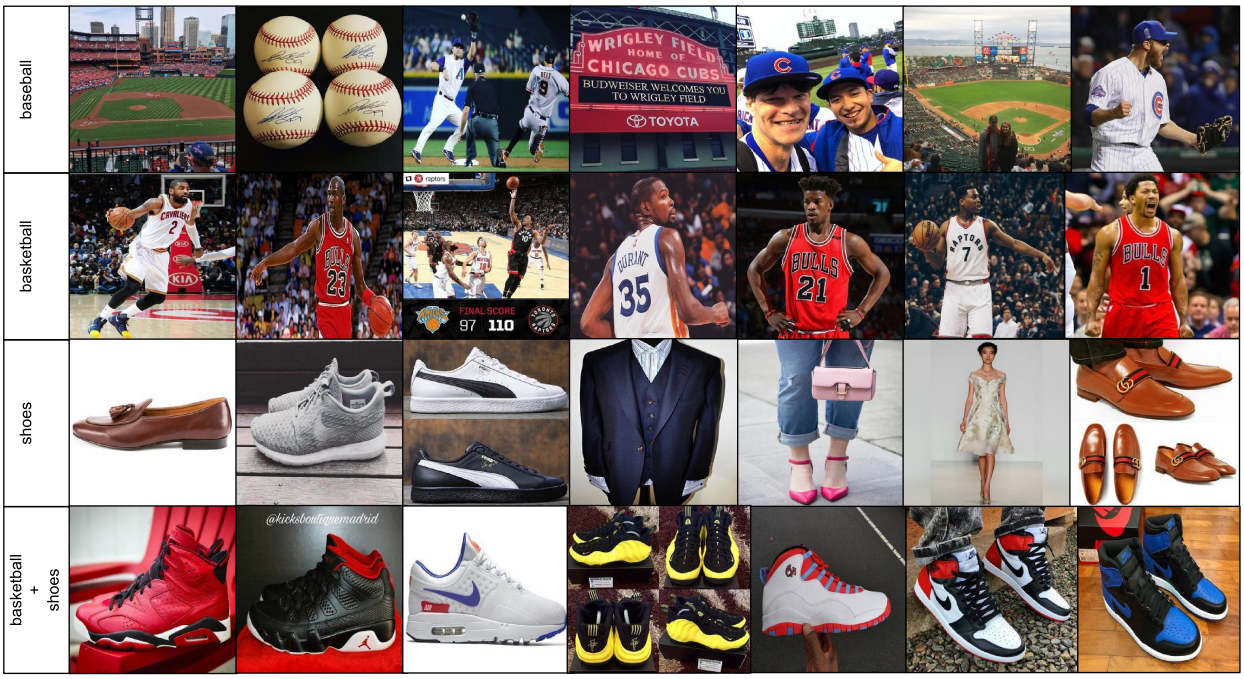

Figure 6: First retrieved images for complex queries with Word2Vec on InstaCites1M.

corresponding to semantic concepts between the two querying concepts are retrieved. That proves that the common embedding space that has been learnt has a semantic structure. The performance is generally better in InstaCities1M than in WebVision. The reason is that the queries are closer to the kind of images people tend to post in Instagram than to the ImageNet classes. However, the results on transfer learning show that WebVision is a better dataset to train than InstaCities1M. That's because WebVision has more images than InstaCities1M (2.4M training images vs 800k training images) and shows that the learned models are robust, general and scalable: Having more data, even if it's not specifically related with the target task, allows learning embedding models that perform better in that task. Results show that all the tested text embeddings methods work quite well for simple queries. Though, LDA fails when is trained in WebVision. That is because LDA learns latent topics with semantic sense from the training data. Every WebVision image is associated to one of the 1,000 ImageNet classes, which influences a lot the topics learning. As a result, the embedding fails when the queries are not related to those classes. The top performing methods are GloVe when training with InstaCities1M and Word2Vec when training with WebVision, but the difference between their performance is small. FastText achieves a good performance on WebVision but a bad performance on InstaCities1M compared to the other methods. An explanation is that, while Social Media data contains more colloquial vocabulary, WebVision contains domain specific and diverse vocabulary, and since FastText learns representations for character ngrams, is more suitable to learn representations from corpus that are morphologically rich. Doc2Vec does not work well in any database. That is because it is oriented to deal with larger texts than the ones we find accompanying images in Web and Social Media. For word embedding methods Word2Vec and GloVe, the results computing the text representation as the mean or as the *tf-idf* weighted mean

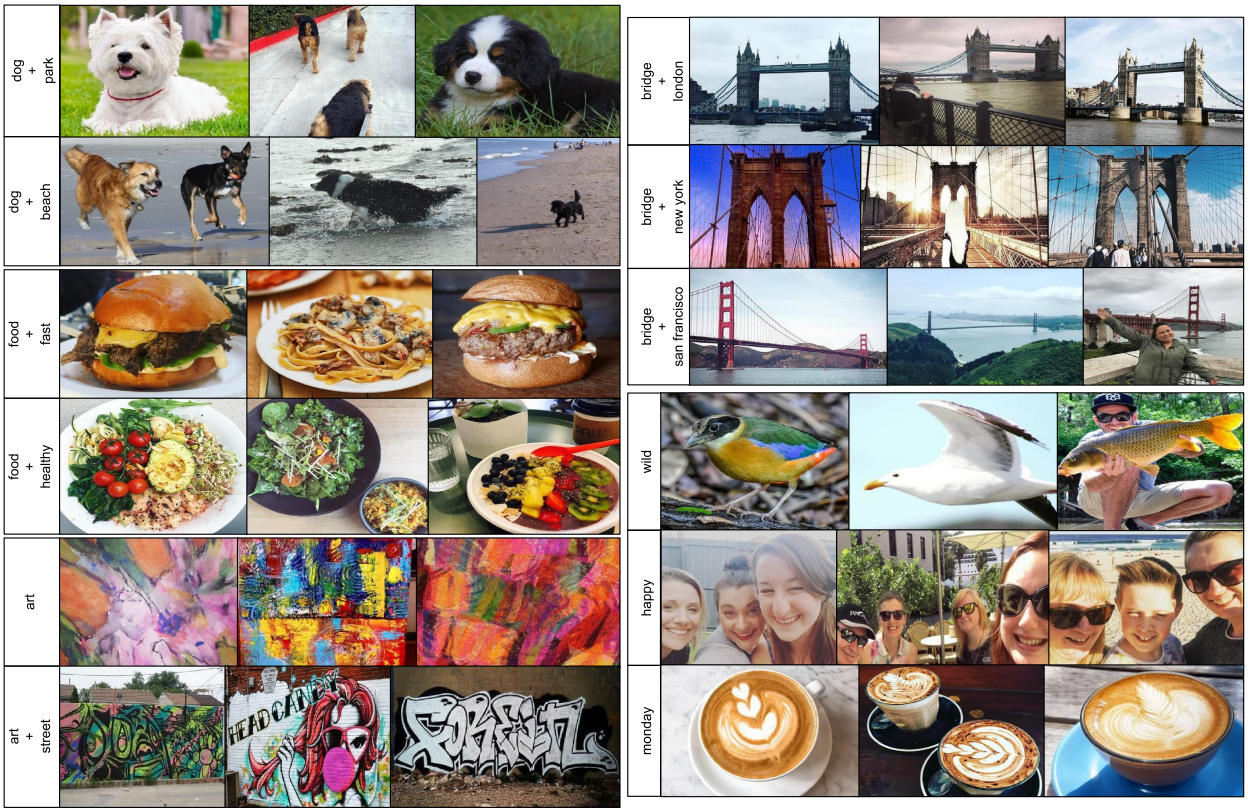

Figure 7: First retrieved images for complex queries (left), city related complex queries (top-right) and non-object queries (bottom-right) with Word2Vec on InstaCites1M.

of the words embeddings are similar.

The overall conclusion of the performance comparison between text embeddings in this experiment is that word level text embeddings such as Word2Vec and GloVe perform better than document level text embeddings (LDA, Doc2Vec) and character ngrams level text embeddings (FastText). The reason is that captions associated to images in Social Media tend to be quite concise, so averaging the word-level embeddings of a caption still gives us an informative representation that allows us to take profit of the rich semantic space learnt by this kind of embeddings. The fact that this semantic space is quite sparse allows us to perform arithmetic between embeddings in it, and also to be able to learn from those representations averaged over caption's words. The introduction of additional artificial noise deteriorates the results heavily. This indicates that, despite the proposed learning pipeline can learn powerful visual features from Web and Social Media data with its inherent noise, reducing it may lead to huge performance improvements.

6.3. Error Analysis

Remarkable sources of errors are listed and explained in this section.

6.3.1. Visual Features Confusion

Errors due to the confusion between visually similar objects. For instance retrieving images of a quiche when querying "pizza". Those errors could be avoided using more data and a higher

Figure 8: First retrieved images for multimodal queries (concepts are added or removed to bias the results) with Word2Vec on WebVision.

dimensional representations, since the problem is the lack of training data to learn visual features that generalize to unseen samples.

6.3.2. Errors from the Dataset Statistics

An important source of errors is due to dataset statistics. As an example, the WebVision dataset contains a class which is "snow leopard" and it has many images of that concept. The word "snow" appears frequently in the images correlated descriptions, so the net learns to embed together the word "snow" and the visual features of a "snow leopard". There are many more images of "snow leopard" than of "snow", therefore, when we query "snow" we get snow leopard images. Figure 11 shows this error and how we can use complex multimodal queries to bias the results.

6.3.3. Words with Different Meanings or Uses

Words with different meanings or words that people use in different scenarios introduce unexpected behaviors. For instance when we query "woman + bag" in the InstaCities1M dataset we usually retrieve images of pink bags. The reason is that people tend to write "woman" in an image caption when pink stuff appears. Those are considered errors in our evaluation, but inferring which images people relate with certain words in Social Media can be a very interesting research.

7. Retrieval in the MIRFlickr Dataset

To compare the performance of our pipeline to other image retrieval by text systems we use the MIRFlickr dataset, which is typically used to train and evaluate image retrieval systems. The objective is to prove the quality of the multimodal embeddings learnt solely with Web data comparing them to supervised methods.

Table 5: MAP on the image by text retrieval task on MIRFlickr as defined in [36, 38].

Method	Train	map
LDA 200	InstaCites1M	0.736
LDA 400	WebVision	0.627
Word2Vec tf-idf	InstaCites1M	0.720
Word2Vec tf-idf	WebVision	0.738
GloVe tf-idf	InstaCites1M	**0.756**
GloVe tf-idf	WebVision	0.737
FastText tf-idf	InstaCities1M	0.677
FastText tf-idf	WebVision	0.734
Word2Vec tf-idf	MIRFlickr	0.867
GloVe tf-idf	MIRFlickr	**0.883**
DCH [36]	MIRFlickr	0.813
LSRH [37]	MIRFlickr	0.768
CSDH [38]	MIRFlickr	0.764
SePH [35]	MIRFlickr	0.735
SCM [39]	MIRFlickr	0.631
CMFH [40]	MIRFlickr	0.594
CRH [41]	MIRFlickr	0.581
KSH-CV [42]	MIRFlickr	0.571

Table 6: MAP on the image by text retrieval task on MIRFlickr as defined in [43].

Method	Train	map
GloVe tf-idf	InstaCites1M	**0.57**
GloVe tf-idf	MIRFlickr	**0.73**
MML [43]	MIRFlickr	0.63
InfR [43]	MIRFlickr	0.60
SBOW [43]	MIRFlickr	0.59
SLKL [43]	MIRFlickr	0.55
MLKL [43]	MIRFlickr	0.56

Table 7: AP scores for 38 semantic concepts and MAP on MIRFlickr. Underlined numbers compare our method trained with InstaCities and other methods trained with the target dataset.

Method	**GloVe tf-idf**	**MMSHL** [44]	**SCM** [39]	**GloVe tf-idf**
Train	**MIRFlickr**			**InstaCities**
animals	**0.775**	0.382	0.353	0.707
baby	**0.337**	0.126	0.127	0.264
baby*	**0.627**	0.086	0.086	0.492
bird	**0.556**	0.169	0.163	0.483
bird*	**0.603**	0.178	0.163	0.680
car	**0.603**	0.297	0.256	0.450
car*	**0.908**	0.420	0.315	0.858
female	**0.693**	0.537	0.514	0.481
female*	**0.770**	0.494	0.466	0.527
lake	**0.403**	0.194	0.182	0.230
sea	**0.720**	0.469	0.498	0.565
sea*	**0.859**	0.242	0.166	0.731
tree	**0.727**	0.423	0.339	0.398
tree*	**0.894**	0.423	0.339	0.506
clouds	**0.792**	0.739	0.698	0.613
clouds*	**0.884**	0.658	0.598	0.710
dog	**0.800**	0.195	0.167	0.760
dog*	**0.901**	0.238	0.228	0.865
sky	**0.900**	0.817	0.797	0.809
structures	**0.850**	0.741	0.708	0.703
sunset	**0.601**	0.596	0.563	0.590
transport	**0.650**	0.394	0.368	0.287
water	**0.759**	0.545	0.508	0.555
flower	**0.715**	0.433	0.386	0.645
flower*	**0.870**	0.504	0.411	0.818
food	**0.712**	0.419	0.355	0.683
indoor	**0.806**	0.677	0.659	0.304
plant _life	**0.846**	0.734	0.703	0.564
portrait	**0.825**	0.616	0.524	0.474
portrait*	**0.841**	0.613	0.520	0.483
river	**0.436**	0.163	0.156	0.304
river*	**0.497**	0.134	0.142	0.326
male	**0.666**	0.475	0.469	0.330
male*	**0.743**	0.376	0.341	0.338
night	**0.589**	0.564	0.538	0.542
night*	**0.804**	0.414	0.420	0.720
people	**0.910**	0.738	0.715	0.640
people*	**0.945**	0.677	0.648	0.658
MAP	**0.738**	0.451	0.415	0.555

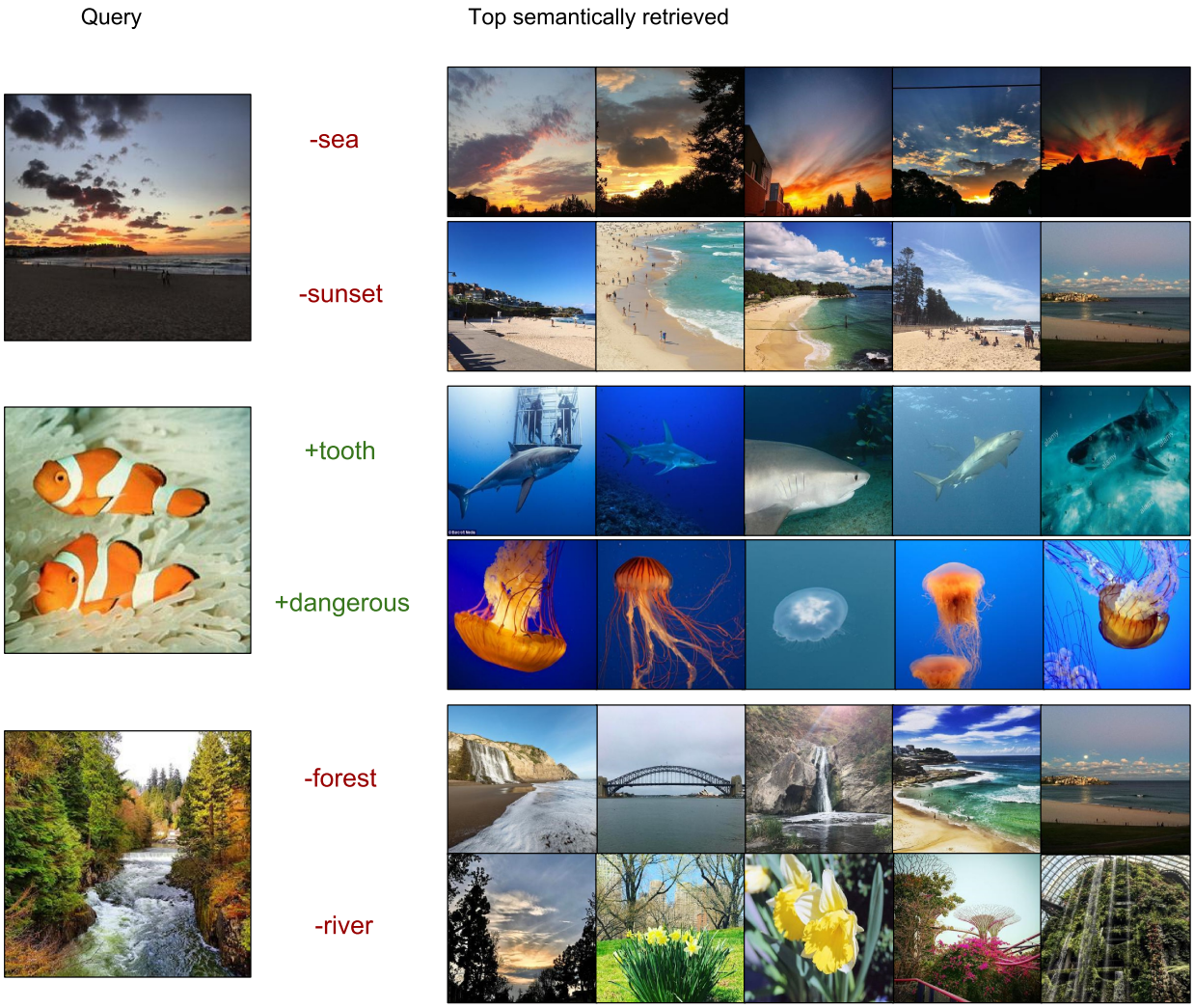

Figure 9: First retrieved images for multimodal complex queries with Word2Vec on WebVision.

7.1. Experiment Setup

We consider three different experiments: 1) Using as queries the tags accompanying the query images and computing the MAP of all the queries. Here a retrieved image is considered correct if it shares at least one tag with the query image. For this experiment, the splits used are 5% queries set and 95% training and retrieval set, as defined in [36, 38]. 2) Using as queries the class names. Here a retrieved image is considered correct if it is tagged with the query concept. For this experiment, the splits used are 50% training and 50% retrieval set, as defined in [44]. 3) Same as experiment 1 but using the MIRFlickr train-test split proposed in Zhang et al. [43].

7.2. Results and Conclusions

Tables 5 and 6 show the results for the experiments 1 and 3 respectively. We see that our pipeline trained with Web and Social Media data in a multimodal self-supervised fashion achieves

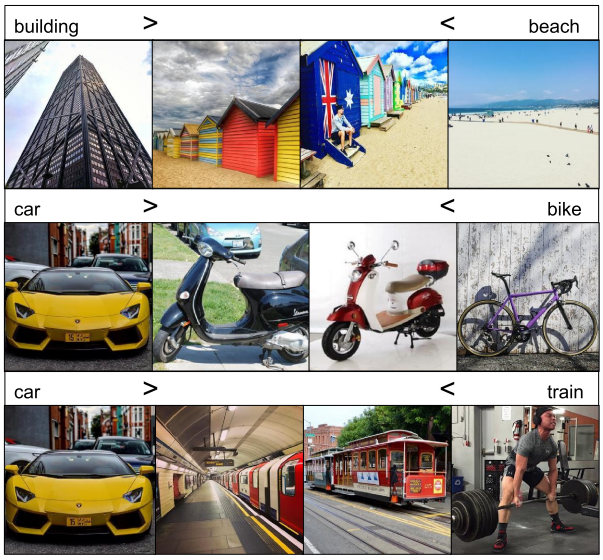

Figure 10: First retrieved images for simple (left and right columns) and complex weighted queries with Word2Vec on InstaCites1M.

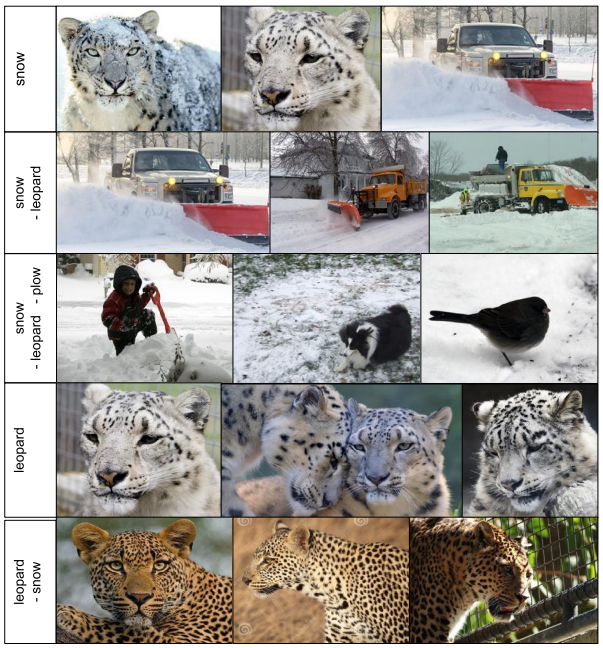

Figure 11: First retrieved images for text queries using Word2Vec on WebVision. Concepts are removed to bias the results.

competitive results. When trained with the target dataset, our pipeline outperforms the other methods. Table 7 shows results for the experiment 2. Our pipeline with the GloVe *tf-idf* text embedding trained with InstaCites1M outperforms state of the art methods in most of the classes and in MAP. If we train with the target dataset, results are improved significantly. Notice that despite being applied here to the classes and tags existing in MIRFlickr, our pipeline is generic and has learnt to produce joint image and text embeddings for many more semantic concepts, as seen in the qualitative examples.

8. Comparing the Image and Text Embeddings

In this section we analyze the semantic quality of the learnt joint embedding spaces showing how the CNN has learnt to embed images in them.

8.1. Experiment Setup

To evaluate how the CNN has learnt to map images to the text embedding space and the semantic quality of that space, we perform the following experiment: We build random image pairs from the MIRFlickr dataset and we compute the cosine similarity between both their image and their text embeddings. In Figure 12 we plot the images embeddings distance vs the text embedding distance of 20,000 random image pairs. If the CNN has learnt correctly to map images to the text embedding space, the distances between the embeddings of the images and the texts of a pair should be similar, and points in the plot should fall around the identity line $y = x$. Also, if

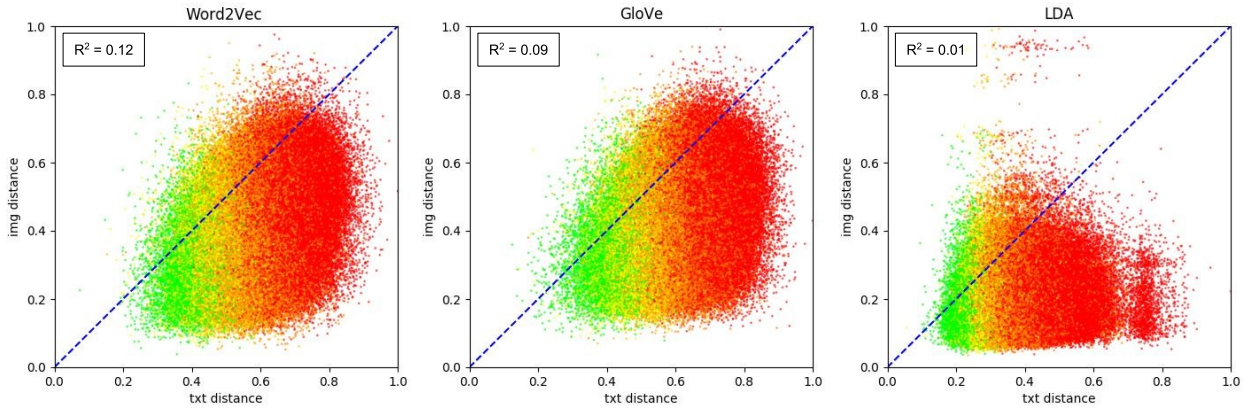

Figure 12: Text embeddings distance (X) vs the images embedding distance (Y) of different random image pairs for LDA, Word2Vec and GloVe embeddings trained with InstaCities1M. Distances have been normalized between [0,1]. Points are red if the pair does not share any tag, orange if it shares 1, light orange if it shares 2, yellow if it shares 3 and green if it shares more. R^2 is the coefficient of determination of images and texts distances.

the learnt space has a semantic structure, both the distance between images embeddings and the distance between texts embeddings should be smaller for those pairs sharing more tags: The plot points' color reflects the number of common tags of the image pair, so pairs sharing more tags should be closer to the axis origin.

As an example, take a dog image with the tag "dog", a cat image with the tag "cat" and one of a scarab with the tag "scarab". If the text embedding has been learnt correctly, the distance between the projections of dog and scarab tags in the text embedding space should be bigger than the one between dog and cat tags, but smaller than the one between other pairs not related at all. If the CNN has correctly learnt to embed the images of those animals in the text embedding space, the distance between the dog and the cat image embeddings should be similar than the one between their tags embeddings (and the same for any pair). So the point given by the pair should fall in the identity line. Furthermore, that distance should be nearer to the coordinates origin than the point given by the dog and scarab pair, which should also fall in the identity line and nearer to the coordinates origin that another pair that has no relation at all.

8.2. Results and Conclusions

The plots in Figure 12 for both the Word2Vec and the GloVe embeddings show a similar shape. The resulting blob is elongated along the $y = x$ direction, which proves that both image and text embeddings tend to provide similar distances for an image pair. The blob is thinner and closer to the identity line when the distances are smaller (so when the image pairs are related), which means that the embeddings can provide a valid distance for semantic concepts that are close enough (dog, cat), but fails inferring distances between weak related concepts (car, skateboard). The colors of the points in the plots show that the space learnt has a semantic structure. Points corresponding to pairs having more tags in common are closer to the coordinates origin and have smaller distances between the image and the text embedding. From the colors it can also be deducted that the CNN is good inferring distances for related images pairs: there are just a few images having more than

3 tags in common with image embedding distance bigger than 0.6, while there are many images with bigger distances that do not have tags in common. However, the visual embedding sometimes fails and infers small distances for image pairs that are not related, as those images pairs having no tags in common and an image embedding distance below 0.2.

The plot of the LDA embedding shows that the learnt joint embedding is not so good in terms of the CNN images mapping to the text embedding space nor in terms of the space semantic structure. The blob does not follow the identity line direction that much which means that the CNN and the LDA are not inferring similar distances for images and texts of pairs. The points colors show that the CNN is inferring smaller distances for more similar image pairs only when the pairs are very related.

The coefficient of determination R^2 shown at each graph measures the proportion of the variance in a dependent variable that is predicted by linear regression and a predictor variable. In this case, it can be interpreted as a measure of how much image distances can be predicted from text distances and, therefore, of how well the visual embedding has learnt to map images to the joint image-text space. It ratifies our plots' visual inspection proving that visual embeddings trained with Word2Vec and GloVe representations have learnt a much more accurate mapping than LDA, and shows that Word2Vec is better in terms of that mapping.

9. Visualizing CNN activation maps

We have proved that, using only Social Media data, state of the art CNNs can be trained in a self-supervised way to learn powerful visual features, capable to discriminate among a huge variety of scenes: from objects to outdoor scenes, abstract concepts or specific buildings. In this experiment we visualize the images from the InstaCities1M retrieval set that generated the highest activations in some CNN units, using the GoogleNet trained from scratch with InstaCites1M and GloVe tf-idf text embedding as self-supervision. We also show the regions of the images that activated most the selected units. To generate those activations maps we used *deconvnet*, proposed by Zeiler et al. [45] and the Caffe implementation presented in [46]. Figure 13 shows the results of a selection of neurons in the *pool5* layer of our model. We can notice that network units are selective to specific buildings, such as Golden Gate Bridge, objects such as guitars, drums or lights to identify concert scenes, or even basketball t-shirts.

10. Visualizing the Learned Semantic Space with t-SNE

In this section we use the t-SNE dimensionality reduction method to reduce the dimensionality of the joint embedding space to 2 dimensions and we show images in that space to visualize its semantic structure.

10.1. Dimensionality Reduction with t-SNE

Inspired by A. Karpathy's work[4], who uses t-SNE to visualize CNN layer features, we use t-SNE[5] [47] to visualize the learnt joint visual and textual embedding. t-SNE is a non-linear

[4] https://cs.stanford.edu/people/karpathy/cnnembed/

[5] https://github.com/lvdmaaten/bhtsne/

dimensionality reduction method, which we use on our 400 dimensional embeddings to produce 2 dimensional embeddings. For each one of the given 400 dimensional visual or textual embeddings, t-SNE computes a 2 dimensional embedding arranging elements that have similar representations nearby, providing a way to visualize the learnt joint image-text space and analyze qualitatively its semantic structure.

10.2. Visualizing both image and text embeddings

As we have learnt a joint image and text embedding space, we can apply t-SNE to both modalities of embeddings at once. We apply t-SNE to a set formed by the visual embeddings of the images in test set of InstaCities1M and the text embeddings of the selected querying terms (Table 1). In this experiment, we use the Word2Vec model trained on InstaCities1M dataset.

10.3. Showing Images at the Embedding Locations

First, we set a canvas with predefined dimensions (2000x2000 pixels). Then we normalize the 2 dimensional embeddings given by t-SNE to fit in the canvas size. Finally, we visualize images at their embedding locations, setting their top-left corner at their embedding location and resizing them to 50x50 pixels. For text embeddings, we use an image containing its words as their representations in the canvas. To get an interpretable visualization avoiding images overlaps, if two images share any pixel in the output figure we omit one of them (prioritizing word images). Therefore, images surrounding word images are not necessary top retrieval results for that word, but they are the nearest images of the ones being represented in the figure.

10.4. Semantic Space Inspection

The joint embeddings 2 dimensional visualization in 14 shows the semantic structure of the learnt space. It shows semantic clusters that the joint embedding has learnt in a self-supervised way from the data distribution, that correspond to different kind of images people tend to post on Instagram. For instance, the figure shows a cluster for food images, a cluster for sport images, a cluster for sunrise images, or a cluster for animal images. It also shows that images of people are very numerous, and that the joint embedding groups them correctly. It can also be appreciated how images we might consider noise, such as images with logos or text, are clustered together. The majority of those images are far from the semantic clusters, isolated and near the figure edges. That is because the joint embedding hasn't been able to find semantic relations between these images and the rest, so it assigns to them embeddings that have not relation with the others. When computing t-SNE, as the objective is to place similar images nearby, this images without semantic relations are set far from the others. Therefore, we can conclude that the pipeline is quite robust to Social Media noise. More t-SNE visualizations of the learnt joint embeddings are avalaible in `https://gombru.github.io/2018/08/01/learning_from_web_data`.

11. Conclusions

In this work we learn a joint visual and textual embedding using Web and Social Media data and we benchmark state of the art text embeddings in the image retrieval by text task,

concluding that GloVe and Word2Vec are the best ones for this data, having a similar performance and competitive performances over supervised methods in the image retrieval by text task. We show that our models go beyond instance-level image retrieval to semantic retrieval and that can handle multiple concepts queries and also multimodal queries, composed by a visual query and a text modifier to bias the results. We clearly outperform state of the art in the MIRFlick dataset when training in the target data. The code used in this work is available on `https://github.com/gombru/LearnFromWebData`.

Acknowledgments

This work was supported by the Doctorats Industrials program from the Generalitat de Catalunya, the Spanish project TIN2017-89779-P, the H2020 Marie Skłodowska-Curie actions of the European Union, grant agreement No 712949 (TECNIOspring PLUS), and the Agency for Business Competitiveness of the Government of Catalonia (ACCIO).

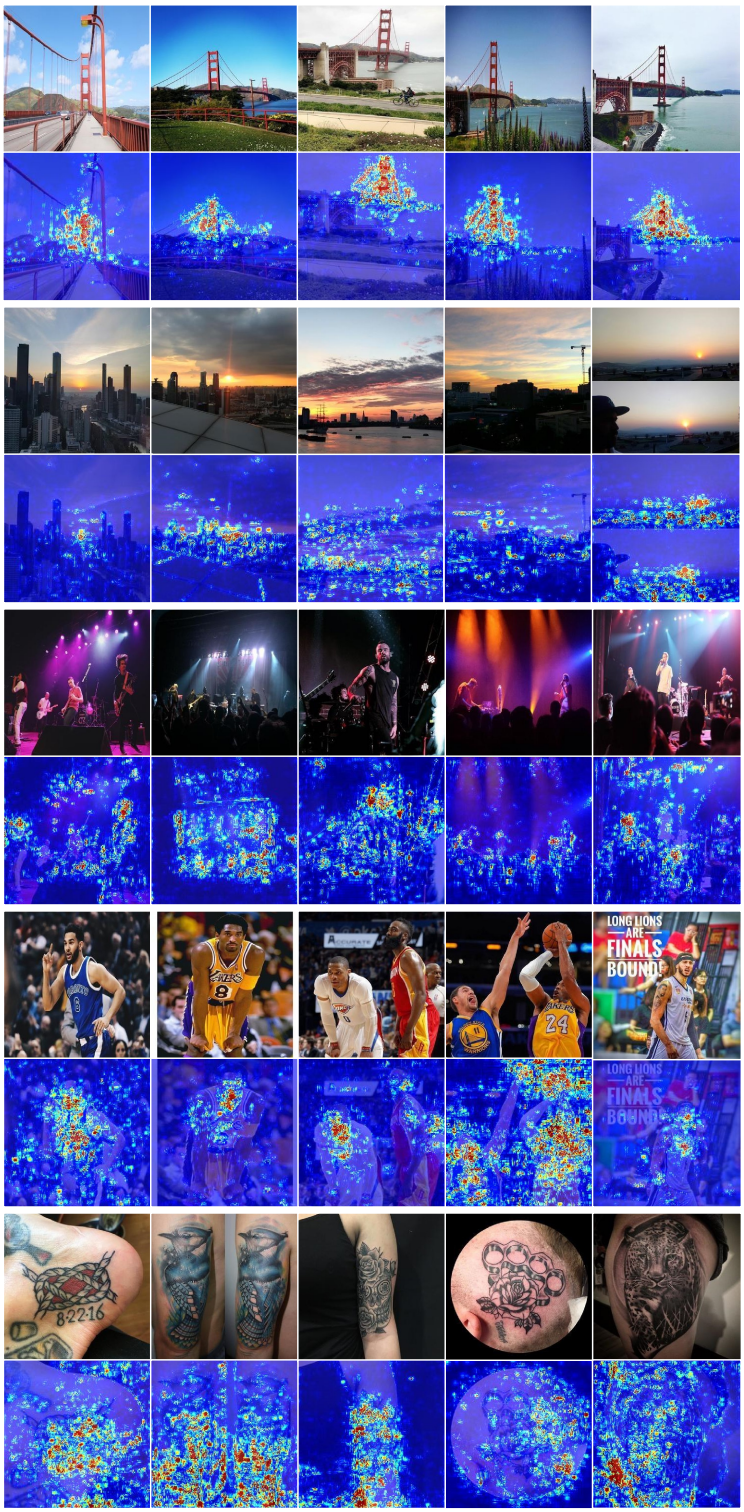

Figure 13: Top-5 activations for five units in pool5 layer of GoogleNet model trained from scratch with InstaCities1M using GloVe tf-idf as self-supervision and their activation maps.

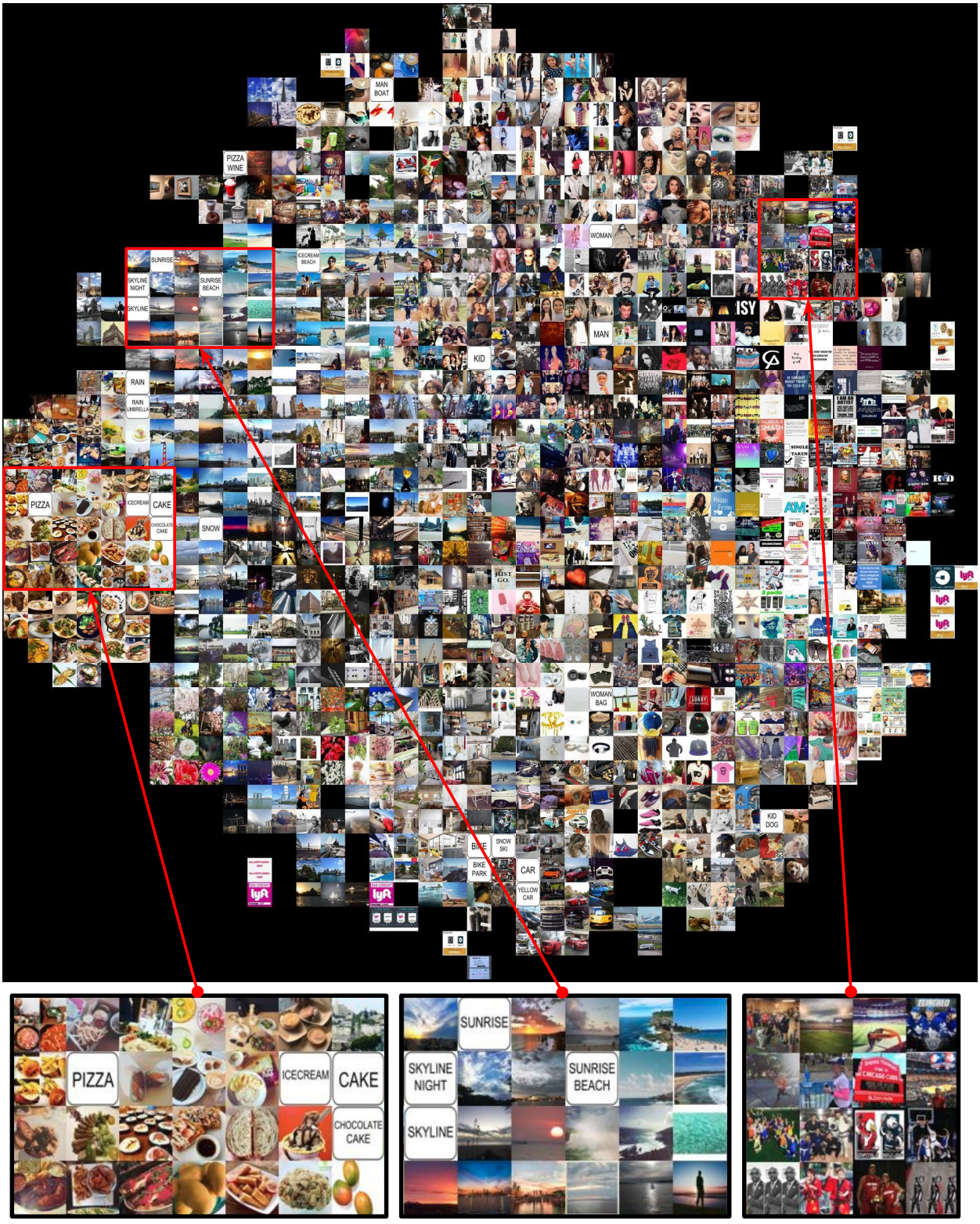

Figure 14: Visualization (2000x2000 px) of the joint embedding with Word2Vec on InstaCities1M dataset.

References

[1] D. Jia, D. Wei, S. R, L. Li-Jia, L. Kai, F.-F. Li, ImageNet: A large-scale hierarchical image database, in: CVPR, 2009.

[2] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, C. L. Zitnick, Microsoft COCO: Common objects in context, in: Lect. Notes Comput. Sci., 2014.

[3] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, A. Torralba, Places: A 10 million Image Database for Scene Recognition, in: TPAMI, 2017.

[4] J. Yosinski, J. Clune, Y. Bengio, H. Lipson, How transferable are features in deep neural networks?, in: NIPS, 2014.

[5] J. Mar, V. David, D. Ger, M. L. Antonio, Learning Apperance in Virtual Scenarios for Pedestrian Detection, in: CVPR, 2010.

[6] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, A. M. Lopez, The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes, in: CVPR, 2016.

[7] T. Q. Phan, P. Shivakumara, S. Tian, C. L. Tan, Recognizing text with perspective distortion in natural scenes, in: ICCV, 2013.

[8] A. Gupta, A. Vedaldi, A. Zisserman, Synthetic Data for Text Localisation in Natural Images, in: CVPR, 2016.

[9] W. Li, L. Wang, W. Li, E. Agustsson, J. Berent, A. Gupta, R. Sukthankar, L. Van Gool, WebVision Challenge: Visual Learning and Understanding With Web Data, in: arXiv, 2017.

[10] D. Mahajan, R. Girshick, V. Ramanathan, K. He, M. Paluri, Y. Li, A. Bharambe, L. Van Der, M. Facebook, Exploring the Limits of Weakly Supervised Pretraining, in: ECCV, 2018.

[11] M. Norouzi, T. Mikolov, S. Bengio, Y. Singer, J. Shlens, A. Frome, G. S. Corrado, J. Dean, Zero-Shot Learning by Convex Combination of Semantic Embeddings, in: NIPS, 2013.

[12] T. Mikolov, G. Corrado, K. Chen, J. Dean, Efficient Estimation of Word Representations in Vector Space, in: ICLR, 2013.

[13] A. Gordo, D. Larlus, Beyond Instance-Level Image Retrieval: Leveraging Captions to Learn a Global Visual Representation for Semantic Retrieval, in: CVPR, 2017.

[14] L. Gomez, Y. Patel, M. Rusiñol, D. Karatzas, C. V. Jawahar, Self-supervised learning of visual features through embedding images into text topic spaces, in: CVPR, 2017.

[15] Y. Patel, L. Gomez, R. Gomez, M. Rusiñol, D. Karatzas, C. V. Jawahar, TextTopicNet - Self-Supervised Learning of Visual Features Through Embedding Images on Semantic Text Spaces, in: arXiv, 2018.

[16] D. M. Blei, A. Y. Ng, M. I. Jordan, Latent Dirichlet Allocation, in: J. Mach. Learn. Res., 2003.

[17] L. Wang, Y. Li, S. Lazebnik, Learning deep structure-preserving image-text embeddings, in: CVPR, 2016.

[18] Y. Patel, L. Gomez, M. Rusiñol, D. Karatzas, Dynamic Lexicon Generation for Natural Scene Images, in: ECCV, 2016.

[19] A. Gordo, J. Almazan, N. Murray, F. Perronin, LEWIS: Latent embeddings for word images and their semantics, in: ICCV, 2015.

[20] Princeton University, WordNet (2010).

[21] A. Salvador, N. Hynes, Y. Aytar, J. Marin, F. Ofli, I. Weber, A. Torralba, Learning Cross-Modal Embeddings for Cooking Recipes and Food Images, in: CVPR, 2017.

[22] R. Gomez, L. Gomez, J. Gibert, D. Karatzas, Learning from #Barcelona Instagram data what Locals and Tourists post about its Neighbourhoods, in: ECCV Workshops, 2018.

[23] G. Patrini, A. Rozza, A. Menon, R. Nock, L. Qu, Making Deep Neural Networks Robust to Label Noise: a Loss Correction Approach, in: CVPR, 2016.

[24] T. Xiao, T. Xia, Y. Yang, C. Huang, X. Wang, Learning From Massive Noisy Labeled Data for Image Classification, in: CVPR, 2015.

[25] J. Fu, Y. Wu, T. Mei, J. Wang, H. Lu, Y. Rui, Relaxing from vocabulary: Robust weakly-supervised deep learning for vocabulary-free image tagging, in: ICCV, 2015.

[26] X. Xu, L. He, H. Lu, A. Shimada, R. I. Taniguchi, Non-Linear Matrix Completion for Social Image Tagging, in: IEEE Access, 2017.

[27] M. Melucci, Relevance Feedback Algorithms Inspired by Quantum Detection, in: IEEE Trans. Knowl. Data Eng., 2016.

[28] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Rethinking the Inception Architecture for Computer Vision, in: CVPR, 2016.
[29] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: Convolutional Architecture for Fast Feature Embedding, in: arXiv, 2014.
[30] J. Pennington, R. Socher, C. Manning, Glove: Global Vectors for Word Representation, in: EMNLP, 2014.
[31] Q. V. Le, T. Mikolov, Distributed Representations of Sentences and Documents, in: NIPS, 2014.
[32] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching Word Vectors with Subword Information, in: arXiv, 2016.
[33] W. Li, L. Wang, W. Li, E. Agustsson, L. Van Gool, WebVision Database: Visual Learning and Understanding from Web Data, in: arXiv, 2017.
[34] M. J. Huiskes, M. S. Lew, The MIR flickr retrieval evaluation, in: ACM Int. Conf. Multimed. Inf. Retr., 2008.
[35] Z. Lin, G. Ding, M. Hu, J. Wang, Semantics-preserving hashing for cross-view retrieval, in: CVPR, 2015.
[36] X. Xu, F. Shen, Y. Yang, H. T. Shen, X. Li, Learning Discriminative Binary Codes for Large-scale Cross-modal Retrieval, in: IEEE Trans. Image Process., 2017.
[37] K. Li, G. J. Qi, J. Ye, K. A. Hua, Linear Subspace Ranking Hashing for Cross-Modal Retrieval, in: IEEE Trans. Pattern Anal. Mach. Intell., 2017.
[38] L. Liu, Z. Lin, L. Shao, F. Shen, G. Ding, J. Han, Sequential discrete hashing for scalable cross-modality similarity retrieval, in: IEEE Trans. Image Process., 2017.
[39] D. Zhang, W.-j. Li, Large-Scale Supervised Multimodal Hashing with Semantic Correlation Maximization, in: AAAI, 2014.
[40] G. Ding, Y. Guo, J. Zhou, Collective matrix factorization hashing for multimodal data, in: CVPR, 2014.
[41] Y. Zhen, D.-Y. Yeung, Co-Regularized Hashing for Multimodal Data, in: NIPS, 2012.
[42] J. Zhou, G. Ding, Y. Guo, Q. Liu, X. Dong, Kernel-based supervised hashing for cross-view similarity search, in: IEEE Int. Conf. Multimed. Expo, 2014.
[43] X. Zhang, X. Zhang, X. Li, Z. Li, S. Wang, Classify social image by integrating multi-modal content, in: Multimed. Tools Appl., Springer US, 2018.
[44] J. Wang, G. Li, A Multi-modal Hashing Learning Framework for Automatic Image Annotation, in: Int. Conf. Data Sci. Cybersp., 2017.
[45] M. D. Zeiler, R. Fergus, Visualizing and Understanding Convolutional Networks, in: ECCV, Springer, Cham, 2014.
[46] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, H. Lipson, Understanding Neural Networks Through Deep Visualization, in: arXiv.
[47] L. Van Der Maaten, A. Courville, R. Fergus, C. Manning, Accelerating t-SNE using Tree-Based Algorithms, in: J. Mach. Learn. Res., 2014.