Better Text Understanding Through Image-To-Text Transfer

Karol Kurach[1], Sylvain Gelly[1], Michal Jastrzebski[1], Philip Haeusser[1, 2], Olivier Teytaud[1], Damien Vincent[1], and Olivier Bousquet[1]

[1]Google Brain
[2]Technische Universität München

Abstract

Generic text embeddings are successfully used in a variety of tasks. However, they are often learnt by capturing the co-occurrence structure from pure text corpora, resulting in limitations of their ability to generalize. In this paper, we explore models that incorporate visual information into the text representation. Based on comprehensive ablation studies, we propose a conceptually simple, yet well performing architecture. It outperforms previous multimodal approaches on a set of well established benchmarks. We also improve the state-of-the-art results for image-related text datasets, using orders of magnitude less data.

1 Introduction

The problem of finding good representations of text data is a very actively studied area of research. Many models are able to learn representations directly by optimizing the end-to-end task in a supervised manner. This, however, often requires an enormous amount of labeled data which is not available in many practical applications and gathering such data can be very costly. A common solution that requires an order of magnitude less labeled examples is to reuse pre-trained embeddings.

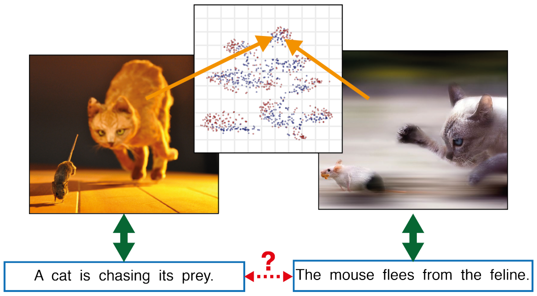

Figure 1: Two images are close in the visual space, which can be quantified via a CNN. Their descriptions convey the same concept, yet using an entirely different vocabulary. Our method improves the understanding of text by leveraging knowledge about visual properties of corresponding images. Photo credit: [17, 22]

A large body of work in this space is focused on training embeddings from pure text data. However, there are many types of relations and co-occurrences that are hard to grasp from pure text. Instead, they appear naturally in other modalities, such as images. In particular, from a similarity measure in the image space and pairs of images and sentences, a similarity measure for sentences can be induced, as illustrated in Figure 1.

In this work, we study how to build sentence embeddings from text-image pairs that are good in terms of sentence similary metrics. This extends previous works, for example [12] or [15].

We propose a conceptually simple, but well performing model that we call Visually Enhanced Text Embeddings (VETE). It takes advantage of visual information from images in order to improve the quality of sentence embeddings. This model uses simple ingredients that already exist and combines them properly. Using a pre-trained Convolutional Neural Network (CNN) for the image embedding, the sentence embeddings are obtained as the normalized sum of the word embeddings. Those are trained end-to-end to be aligned with the corresponding image embeddings and not aligned with mismatching pairs, optimizing the Pearson correlation.

Despite its simplicity, the model significantly outperforms pure-text models and the best multimodal model from [15] on a set of well established text similarity benchmarks from the SemEval competition [18]. In particular, for image-related datasets, our model matches state-of-the-art results with substantially less training data. These results indicate that exploring image data can significantly improve the quality of text embeddings and that incorporating images as a source of information can result in text representations which effectively captures visual knowledge. We also conduct a detailed ablation study to quantify the effect of different factors on the embedding quality in the image-to-text transfer setup.

In summary, the contributions of this work are:

- We propose a simple multimodal model that outperforms previous image-text approaches on a wide variety of text similarity tasks. Furthermore, the proposed model matches state-of-the-art results on image-related SemEval datasets, despite being trained with substantially less data.
- We perform a comprehensive study of image-to-text transfer, comparing different model types, text encoders, loss types and datasets.
- We provide evidence that the approach of learning sentence embeddings directly outperforms methods that learn word embeddings and then combine them.

2 Related Work

Many works study the use of multimodal data, in particular pairs of images and text. Most of them explore how these pairs of data can be leveraged for tasks that require knowledge of both modalities, like captioning or image retrieval [1, 7, 11, 12, 29, 8, 26].

While this line of work is very interesting as common embeddings can directly be applied to captioning or image retrieval tasks, the direct use of text embeddings for NLP tasks, using images only as auxiliary training data, is less explored.

[12] propose to extend the skip-gram algorithm [16] to incorporate image data. [4] also took a similar approach before. In the original skip-gram algorithm, each word embedding is optimized to increase the likelihood of the neighboring words conditioned on the center word. In addition to predicting contextual words, [12]'s models maximize the similarity between image and word embeddings. More precisely:

$$L_{ling}(w_t) = \sum_{-c \leq j \leq c, j \neq 0} \log p(w_j|w_t)$$

where $p(w_j|w_t)$ is given by a softmax formulation:

$$p(w_j|w_t) = \frac{e^{f(w_j) \cdot f(w_t)}}{\sum_w e^{f(w) \cdot f(w_t)}}$$

$f(w)$ is the embedding vector of the word w and $v \cdot v'$ is the dot product between the vectors v and v'. To inject visual information, [12] add a max margin loss between the image embedding and the word embedding:

$$L_{vision}(w_t) = \mathbb{E}_{w'} \max(0, \gamma - \cos(f(w_t), g(w_t)) + \cos(f(w_t), g(w')))$$

with $g(w)$ being the average embedding of the images paired with the word w.

They show that they can augment word embeddings learnt from large text sources with visual information and in addition to image labeling and retrieval, they show that those embeddings perform better on word similarity metrics.

[11] also use a max margin loss to co-embed images and corresponding text. For the text embeddings, they use different models depending on the task. For the image captioning or image retrieval task,

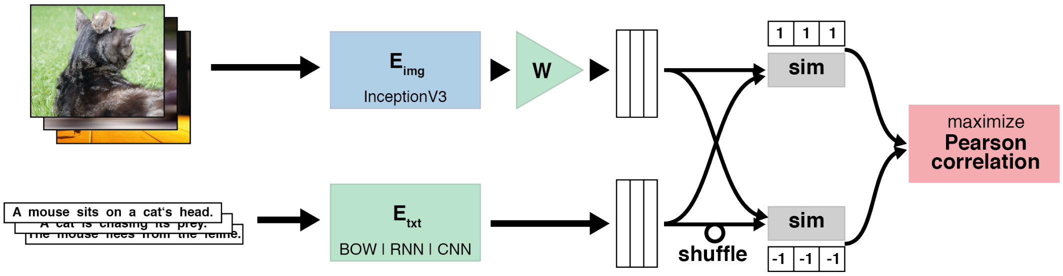

Figure 2: An overview of the VETE model. Images are fed into a pre-trained CNN (E_{img}). Their representation is then transformed via W to match the dimension of the sentence embeddings. The sentences are encoded via a text embedding model (E_{txt}). Finally, the embeddings are paired in two ways: matching pairs and incorrect pairs. For each set of pairs, we compute the similarity. The loss signal comes from the Pearson correlation, which should be 1 for matching pairs and -1 for incorrect pairs. Only green shaded modules are trained. Photo credit: [3, 17, 22]

they employ an LSTM encoder. To explore the resulting word embedding properties, in particular arithmetics, they use a simpler word embedding model. Some interesting arithmetical properties of the embeddings are demonstrated, like "image of a blue car" - "blue" + "red" leading to an image of a red car. However, there is no quantitative evaluation of the quality of the text embeddings in terms of text similarity.

Some more recent works investigate phrase embeddings trained with visual signals and their quality in terms of text similarity metrics. For example, [15] use an Recurrent Neural Network (RNN) as a language model in order to learn word embeddings which are then combined to create a phrase embedding. They propose three models. For their model A, they use a similar setup as the captioning model from [27], with an RNN decoder conditioned on a pre-trained CNN embedding. The RNN (GRU in that case) reads the text, trying to predict the next token. The initial state is set to a transformation of the last internal layer of a pre-trained VGGNet [24]. Let that vector be v_{image}. Model B tries to match the final RNN state with v_{image}. Finally, model C develops the multimodal skip-gram [12] by adding an additional loss measuring the distance between the word embeddings and v_{image}. The authors' experiments show that model A performs best and we use that model as a baseline in our experiments.

3 VETE Models

Our setup aims at directly transferring knowledge between image and text representations and the main goal is to find reusable sentence embeddings. We make direct use of paired data, consisting of pictures and text describing them. We propose a model consisting of two separate encoders - one for images and another one for text. An overview of the archiecture is presented in Figure 2.

For the text encoder, we consider three families of models which combine words into text representations. For the bag-of-words (*BOW*) model, the sentence embedding is simply a normalized sum of vectors corresponding to the individual words. For the *RNN* model, we create a stacked recurrent neural network encoder (LSTM or GRU based). Finally, for the *CNN* model, the encoder includes a convolutional layer followed by a fully connected layer, as described in [9].

For encoding images, we use a pre-trained *InceptionV3* network [25] which provides a 2048-dimensional feature vector for each image in the dataset (images are rescaled and cropped to 300 x 300 px).

Let $E_{img}(I)$ denote the 2048-dimensional embedding vector for an image I and $E_{txt}(S)$ the N-dimensional embedding for sentence S produced by $txt \in \{BOW, RNN, CNN\}$. Throughout this paper, we will refer to the cosine similarity of two vectors v_1, v_2 as $sim(v_1, v_2) = \frac{v_1 \cdot v_2}{\|v_1\|\|v_2\|}$. Informally speaking, our training goal is to maximize $sim(E_{img}(I), E_{txt}(S))$, when sentence S is paired with (i.e. describes) image I, and minimize this value otherwise.

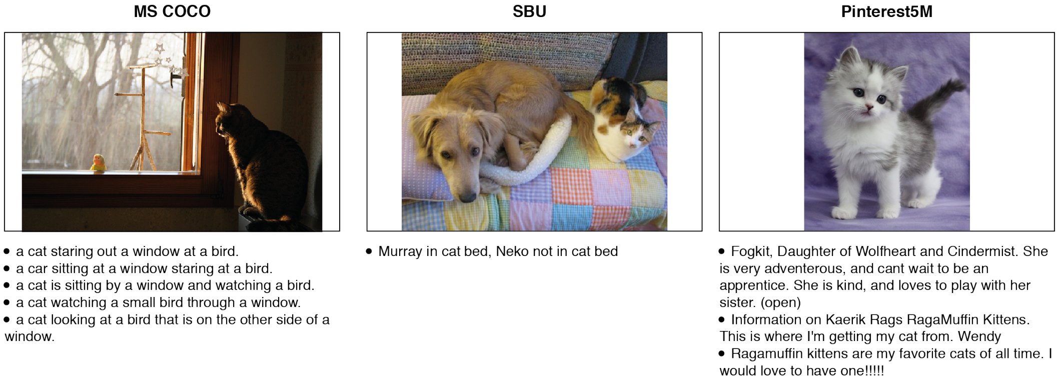
MS COCO
SBU
Pinterest5M
• a cat staring out a window at a bird.
• a car sitting at a window staring at a bird.
• a cat is sitting by a window and watching a bird.
• a cat watching a small bird through a window.
• a cat looking at a bird that is on the other side of a window.
• Murray in cat bed, Neko not in cat bed
• Fogkit, Daughter of Wolfheart and Cindermist. She is very adventerous, and cant wait to be an apprentice. She is kind, and loves to play with her sister. (open)
• Information on Kaerik Rags RagaMuffin Kittens. This is where I'm getting my cat from. Wendy
• Ragamuffin kittens are my favorite cats of all time. I would love to have one!!!!!

Figure 3: Examples from the three datasets. Shown are images and all provided captions. Notice how the language varies from formal descriptions geared towards a general audience (MS COCO) to more informal posts (SBU, Pinterest5M).

Formally, let V be the vocabulary, N the embedding dimensionality and $txt \in \{BOW, RNN, CNN\}$ the text encoding model. Let's define an affine transformation $W \in \mathbb{R}^{2048 \times N}$ that transforms the 2048-dimensional image embeddings to N dimensions. Our learnable parameters consist of a word embedding matrix $\in \mathbb{R}^{|V| \times N}$, the internal parameters of the txt model (when txt is an RNN or CNN encoder), as well as the transformation matrix W. For each batch of image-sentence pairs of the form $(I_1, S_1), ..., (I_B, S_B)$, we randomly shuffle the sentences $S_1, ..., S_B$, and add the following "incorrect" pairs to the batch $(I_1, S_{\sigma(1)}), ..., (I_B, S_{\sigma(B)})$, with $\{\sigma(1), .., \sigma(B)\}$ a random permutation of $\{1, ..., B\}$. If we denote

$$sim(I_i, S_j) := sim(WE_{img}(I_i), E_{txt}(S_j))$$

then our training goal is to maximize the Pearson correlation $\rho(x, y) := \frac{\text{Cov}(x,y)}{\text{Std}(x)\text{Std}(y)}$ between the vectors

$$[\underbrace{sim(I_1, S_1), ..., sim(I_B, S_B)}_{B \text{ correct pairs}}, \underbrace{sim(I_1, S_{\sigma(1)}), ..., sim(I_B, S_{\sigma(B)})}_{B \text{ wrong pairs}}]$$

and

$$[\underbrace{1, 1, ..., 1}_{B \text{ positive labels}}, \underbrace{-1, -1, ..., -1}_{B \text{ negative labels}}].$$

We will denote models trained this way VETE-BOW, VETE-RNN and VETE-CNN, respectively.

4 Experiments

4.1 Training Datasets

In our experiments we consider three training datasets: MS COCO, SBU and Pinterest5M. They are described in detail below. One important note is that we modify datasets that contain multiple captions per image (MS COCO and Pinterest5M) to keep only one caption. This was done to prevent the network from "cheating" by using the image feature vector only as a way of joining text pairs. To the best of our knowledge, we are the first to notice this issue in the evaluation of the quality of multimodal image-text models. It is known that training similarity models directly on text-text pairs yields good results [30] but here we want to investigate only the effect of knowledge transfer from images.

MS COCO The MS COCO dataset [13] contains 80 image categories. For each image five high-quality captions are provided. We use the MS COCO 2014 dataset using the same train/validation/test split as the im2txt [23] Tensorflow implementation of [28]. Initially, our train/validation/test sets contain 586k/10k/20k examples, respectively. Then, we filter the sets to keep only one caption per image, so the "text part" of our final datasets is five times smaller.

Model	imagess2014	images2015	COCO-Test	Pin-Test	avg2014	avg2015
Word2Vec	0.466	0.441	0.379	0.383	0.343	0.367
PureTextRnn	0.662	0.692	0.705	0.484	0.517	0.568
PinModelA	0.671	0.683	0.709	0.536	0.493	0.573
VETE-RNN	0.838	0.835	0.901	0.549	0.538	0.587
VETE-CNN	0.808	0.773	**0.911**	0.528	0.435	0.486
VETE-BOW	**0.861**	**0.855**	0.894	**0.579**	**0.579**	**0.622**

Table 1: Results of models trained only on MS COCO data with one sentence per image.

SBU The Stony Brook University dataset[19] consists of 1M image-caption pairs collected from Flickr, with only one caption per image. We randomly split this dataset into train/validation/test sets with sizes 900k/50k/50k, respectively.

Pinterest5M The original Pinterest40M dataset [15] contains 40M images. However, only 5M image urls were released at the time of this submission. Unfortunately, some images are no longer available so we were able to collect approx. 3.9M images from this dataset. For every image we keep only one caption. Then, we randomly split the data into 3.8M/50k/50k train/validation/test sets, respectively.

The training data in all datasets is lowercased and tokenized using the Stanford Tokenizer [14]. We also wrap all sentences with "<S>" and "</S>" marking the beginning and the end of the sentence.

4.2 Hyperparameter selection and training

The performance of every machine learning model highly depends on the choice of its hyperparameters. In order to fairly compare our approach to previous works, we follow the same hyperparameter search protocol for all models. We choose the average score on the SemEval 2016 (c.f. Section 4.3) as the validation metric and refer to this as “avg2016”.

Algorithm 1: Protocol for hyperparameter search.

```
for i=1,2,...,100 do
    Sample a set of hyperparameters in the allowed ranges;
    Run training and evaluate on “avg2016”;
end
Report the results on all benchmarks of the model that has the highest score on “avg2016”;
```

If a hyperparameter has a similar meaning in two models (e.g., learning rate, initialization scale, lr decay, etc.), the ranges searched were set to be the same. Additionally, we ensured that the parameters reported by the authors are included in the ranges.

In all models, we train using the Adam optimizer[10], for 10 (MS COCO, SBU) or 5 epochs (Pinterest5M). The final embeddings have size of 128. For all VETE models we use the Pearson loss (see ablation study in Section 5.2).

4.3 Evaluation

Our goal is to create good text embeddings that encode knowledge from corresponding images. To evaluate the effect of this knowledge transfer from images to text, we use a set of textual semantic similarity datasets from the SemEval 2014 and 2015 competitions[18]. Unfortunately, we could not compare our models directly on the Gold RP10K dataset introduced by [15] as it was not publicly released.

We also use two additional custom test sets: *COCO-Test* and *Pin-Test*. These were created from the MS COCO and Pinterest5M test datasets, respectively, by randomly sampling $1,000$ semantically related captions (from the same image) and $1,000$ non-related captions from different images. As opposed to the SemEval datasets, the similarity score is binary in this case. The goal was to check the performance on the same task as SemEval but with data from the same distribution of words as our training data.

Model	images2014	images2015	COCO-Test	Pin-Test
Glove (R)	0.624	0.686	0.668	0.422
Glove (NR)	0.625	0.688	0.667	0.471
PinModelA	0.671	0.683	0.708	0.536
M-Skip-Gram (R)	0.764	0.767	0.784	0.608
M-Skip-Gram (NR)	0.765	0.767	0.784	**0.654**
PP-XXL (R)	0.802	0.831	0.770	0.609
PP-XXL (NR)	0.804	0.833	0.770	0.638
Best SemEval	0.834	**0.871**	N/A	N/A
VETE-BOW (our)	**0.861**	0.855	**0.894**	0.579

Table 2: Comparison of models on image-related text datasets. VETE and PinModelA were trained only on MS COCO.

For every model type we select the best model according to the average score on the SemEval 2016 datasets. Then, we report the results on all other test datasets.

4.4 Results

Table 1 presents the scores obtained by models trained only on MS COCO datasets. This allows us to fairly compare only the algorithms, not the data used. In Section 5.3, we analyze the robustness of our methods on two additional datasets (SBU and Pinterest5M).

As a direct comparison, we implement $ModelA$ as described in [15], which we refer to as *PinModelA*. Our implementation uses a pre-trained InceptionV3 network for visual feature extraction, as in the VETE models. To understand the impact of adding information from images to text data, we also evaluate two models trained purely on text:

- **RNN-based language model** This model learns sentence embeddings via an RNN based language model. It corresponds to the PureTextRnn baseline from [15].
- **Word2Vec** We trained Word2Vec word embeddings [21] where the corpus consists of sentences from MS COCO.

All VETE models outperform pure-text baselines and PinModelA. Similarly to [30], we observed that RNN-based encoders are outperformed by a simpler BOW model. We also show that this holds for CNN-based encoders. It is worth noting that it is mostly a domain adaptation issue, as both RNN and CNN encoders perform better than BOW on *COCO-Test*, where the data has the same distribution as the training data. We analyze the effect of changing the text encoder in Section 5.1.

To put our results in context, Table 2 compares them to other methods trained with much larger corpora. We used word embeddings obtained from three methods:

- **Glove**: embeddings proposed in [20], trained on a Common Crawl dataset with 840 billion tokens.
- **M-Skip-Gram**: embeddings proposed in [12], trained on Wikipedia and a set of images from ImageNet.
- **PP-XXL**: the best embeddings from the [30], trained on 9M phrase pairs from PPDB.

For all three approaches, we consider two versions that differ in the vocabulary allowed at inference time. One experiment was done with a vocabulary restricted to MS COCO (marked with “R”) and the non-restricted version (“NR”) where we use the whole vocabulary for given embeddings. The vocabulary size has a significant impact on the final score for the Pinterest-Test benchmark, where 16.5% of all tokens are not in the MS COCO vocabulary. That means that 97.6% of all sentences have at least one missing token.

Finally, we also include the best results from the SemEval competition, where available. Note that those were obtained from heavily tuned and more complex models, trained without any data restrictions. Still, our VETE model is able to match their results.

5 Ablation studies

To analyze the impact of different components of our architecture, we perform ablation studies on the employed text encoder, loss type and training dataset. We also investigate the effect of training on word or sentence level. In all cases, we follow a similar protocol as in Section 4.2:

```
Algorithm 2: Protocol for hyperparameter ablation study.
Randomly generate 100 sets of combinations for all hyperparameters.;
for Hyperparameter p (e.g, "loss type") do
    for v in the allowed range of values for p do
        Run training using the 100 sets of hyperparameters, keeping p=v fixed.;
    end
    Choose the best one based on "avg2016" validation metric, and report the scores.;
end
```

5.1 Encoder

We study the impact of the different text encoders on the VETE model. The results are summarized in Table 3. "RNN-GRU" and "RNN-LSTM" denote RNN encoders with GRU [2] and LSTM [5] cells, respectively. For BOW, we try two options: either we use the sum or the mean of word embeddings. Both bag-of-words encoders perform better than RNN encoders, although RNNs are slightly better on the test data which has the same distribution as the training data.

Encoder	images2014	images2015	COCO-Test	Pin-Test
RNN-GRU	0.834	0.821	**0.906**	0.507
RNN-LSTM	0.838	0.835	0.901	0.549
BOW-SUM	0.860	0.853	0.898	0.573
BOW-MEAN	**0.861**	**0.855**	0.894	**0.579**

Table 3: Results of applying different text encoders to the VETE model. The training data is MS COCO, and the RNN-based models learned to model this distribution better. However, BOW generalizes better to other datasets.

Loss type	Avg score
Covariance	0.594
$\text{SKT}_{.2}$	0.616
SKT_{1}	0.730
Rank loss	0.788
SKT_{5}	0.791
Pearson	**0.797**

Table 4: Comparison of different loss types with the VETE-BOW model.

5.2 Loss type

In this section, we describe various loss types that we trained our model with. Consider two paired variables x (similarity score between two embeddings) and $y \in \{-1, 1\}$. Then, the sample sets $(x_1, \ldots, x_n)$ and $(y_1, \ldots, y_n)$ stand for n corresponding realizations of x and y.

- **Covariance**: $\text{Cov}(x, y)$.
- **The Pearson correlation** ρ: measures the linearity of the link between two variables, estimated on a sample; it is defined as $\rho(x, y) = \frac{\text{Cov}(x,y)}{\text{Std}(x)\text{Std}(y)}$.
- **Surrogate Kendall** τ: The Pearson correlation takes into account only linear dependencies. To mitigate this, we experimented with the Kendall correlation τ which is only rank-dependent. Unfortunately, it is not differentiable. We therefore used its differentiable approximation: SKT_α [6] defined as $SKT_\alpha(x, y) = \frac{\sum_{i,j} \tanh(\alpha(x_i - x_j)(y_i - y_j))}{n(n-1)/2}$ for some $\alpha > 0$.
- **Rank loss**: Another cost function is the pairwise ranking loss. We follow closely the definition in [11].

Table 4 compares the effects of the various losses.

5.3 Dataset

We study the effect of the training dataset. The results of training the model on MS COCO, SBU and Pinterest5M dataset are presented in Table 5. Each cell of the table contains the average score of 4 evaluation datasets (images2014, images2014, COCO-Test, Pin-Test). The quality of image captions varies significantly between the datasets, as can be seen in Figure 3. However, we conclude that the relation between the models is preserved, that is: regardless of the dataset used for training, PinModelA is always worse than VETE-RNN, which in turn is worse than VETE-BOW.

Train Dataset	**Word2Vec**	**PinModelA**	**VETE-RNN**	**VETE-BOW**
MS COCO	0.417	0.650	0.780	**0.797**
SBU	0.413	0.632	0.737	**0.775**
Pinterest5M	0.408	0.609	0.753	**0.803**

Table 5: Comparison of average test scores when training on different datasets.

5.4 Sentence-level vs word-level embedding

Previous methods for transferring knowledge from images to text focused on improving the word-level embeddings. A sentence representation could then be created by combining them. In our work, we learn sentence embeddings as a whole, but the best performing text encoder turned out to be BOW. This raises the following question: could the model perform equally well if we train it on word-level, and then only combine word embeddings during inference? The comparison of these two approaches is presented in Table 6 which clearly shows the benefit of sentence-level training. This effect should be studied further, but while separately training word embeddings forces each of them to be close to the corresponding images, training at the sentence level gives the opportunity to have the word embeddings become complementary, each of them explaining a part of the image, and capturing co-occurences.

Model	**images2014**	**images2015**	**COCO-Test**	**Pin-Test**
Word-level	0.576	0.617	0.675	0.371
Sentence-level	**0.861**	**0.855**	**0.894**	**0.579**

Table 6: Comparison of training on a word-level vs sentence-level.

6 Conclusion

We studied how to improve text embeddings by leveraging multimodal datasets, using a pre-trained image model and paired text-image datasets. We showed that VETE, a simple approach which directly optimizes phrase embeddings to match corresponding image representations, outperforms previous multimodal approaches which are sometimes more complex and optimize word embeddings as opposed to sentence embeddings. We also showed that even for relatively complex similarity tasks at sentence levels, our proposed models can create very competitive embeddings, even compared to more sophisticated models trained on orders of magnitude more text data, especially when the vocabulary is related to visual concepts.

To our initial surprise, state-of-the-art encoder models, like LSTMs, performed significantly worse than much simpler encoders like bag-of-word models. While they achieve better results when evaluated on the same data distribution, their embeddings do not transfer well to other text distributions. General embeddings need to be robust to distribution shifts and applying such techniques can probably further improve the results.

Using a multimodal approach in order to improve general text embeddings is under-explored and we hope that our results motivate further developments. For example, the fact that the best models are very simple suggests that there is a large headroom in that direction.

References

[1] K. Barnard, P. Duygulu, D. Forsyth, N. d. Freitas, D. M. Blei, and M. I. Jordan. Matching words and pictures. *Journal of machine learning research*, 3(Feb):1107–1135, 2003.

[2] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[3] D. Defreyne. Flickr. `https://www.flickr.com/photos/denisdefreyne/1091487059`, 2007. [Online; accessed 17-May-2017].

[4] F. Hill and A. Korhonen. Learning abstract concept embeddings from multi-modal data: Since you probably can't see what i mean. In *EMNLP*, pages 255–265, 2014.

[5] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[6] W. Huang, K. L. Chan, H. Li, J. Lim, J. Liu, and T. Y. Wong. Content-based medical image retrieval with metric learning via rank correlation. In F. Wang, P. Yan, K. Suzuki, and D. Shen, editors, *Machine Learning in Medical Imaging, First International Workshop, MLMI 2010, Held in Conjunction with MICCAI 2010, Beijing, China, September 20, 2010. Proceedings*, volume 6357 of *Lecture Notes in Computer Science*, pages 18–25. Springer, 2010.

[7] Y. Jia, M. Salzmann, and T. Darrell. Learning cross-modality similarity for multinomial data. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2407–2414. IEEE, 2011.

[8] 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*, pages 3128–3137, 2015.

[9] Y. Kim. Convolutional neural networks for sentence classification. In *Empirical Methods in Natural Language Processing*, pages 1746–1751, 2014.

[10] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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

[12] A. Lazaridou, N. T. Pham, and M. Baroni. Combining language and vision with a multimodal skip-gram model. *CoRR*, abs/1501.02598, 2015.

[13] 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 *European Conference on Computer Vision*, pages 740–755. Springer, 2014.

[14] C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky. The stanford corenlp natural language processing toolkit. In *ACL (System Demonstrations)*, pages 55–60, 2014.

[15] J. Mao, J. Xu, K. Jing, and A. L. Yuille. Training and evaluating multimodal word embeddings with large-scale web annotated images. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 442–450. Curran Associates, Inc., 2016.

[16] T. Mikolov, W.-t. Yih, and G. Zweig. Linguistic regularities in continuous space word representations. In *Hlt-naacl*, volume 13, pages 746–751, 2013.

[17] J. Moes. Flickr. `https://www.flickr.com/photos/jeroenmoes/4265223393`, 2010. [Online; accessed 17-May-2017].

[18] P. Nakov, T. Zesch, D. Cer, and D. Jurgens, editors. *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. Association for Computational Linguistics, Denver, Colorado, June 2015.

[19] V. Ordonez, G. Kulkarni, and T. L. Berg. Im2text: Describing images using 1 million captioned photographs. In *Neural Information Processing Systems (NIPS)*, 2011.

[20] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543, 2014.

[21] R. Řehůřek and P. Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. `http://is.muni.cz/publication/884893/en`.

[22] F. Rosa. Flickr. `https://www.flickr.com/photos/kairos_of_tyre/6318245758`, 2011. [Online; accessed 17-May-2017].

[23] C. Shallue. Show and Tell: A Neural Image Caption Generator. `https://github.com/tensorflow/models/tree/master/im2txt`, 2016. [Online; accessed 10-May-2017].

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

[25] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.

[26] I. Vendrov, R. Kiros, S. Fidler, and R. Urtasun. Order-embeddings of images and language. *arXiv preprint arXiv:1511.06361*, 2015.

[27] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[28] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: Lessons learned from the 2015 MSCOCO image captioning challenge. *CoRR*, abs/1609.06647, 2016.

[29] L. Wang, Y. Li, and S. Lazebnik. Learning deep structure-preserving image-text embeddings. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5005–5013, 2016.

[30] J. Wieting, M. Bansal, K. Gimpel, and K. Livescu. Towards universal paraphrastic sentence embeddings. *arXiv preprint arXiv:1511.08198*, 2015.