Bidirectional Retrieval Made Simple

Jônatas Wehrmann
School of Technology
Pontifícia Universidade Católica
do Rio Grande do Sul
jonatas.wehrmann@acad.pucrs.br

Rodrigo C. Barros
School of Technology
Pontifícia Universidade Católica
do Rio Grande do Sul
rodrigo.barros@pucrs.br

Abstract

This paper provides a very simple yet effective character-level architecture for learning bidirectional retrieval models. Aligning multimodal content is particularly challenging considering the difficulty in finding semantic correspondence between images and descriptions. We introduce an efficient character-level inception module, designed to learn textual semantic embeddings by convolving raw characters in distinct granularity levels. Our approach is capable of explicitly encoding hierarchical information from distinct base-level representations (e.g., characters, words, and sentences) into a shared multimodal space, where it maps the semantic correspondence between images and descriptions via a contrastive pairwise loss function that minimizes order-violations. Models generated by our approach are far more robust to input noise than state-of-the-art strategies based on word-embeddings. Despite being conceptually much simpler and requiring fewer parameters, our models outperform the state-of-the-art approaches by 4.8% *in the task of description retrieval and* 2.7% *(absolute R@1 values) in the task of image retrieval in the popular MS COCO retrieval dataset. We also show that our models present solid performance for text classification, specially in multilingual and noisy domains.*

1. Introduction

The problem that we address in this paper is bidirectional retrieval, also regarded as multimodal content retrieval or image-text alignment. In this scenario, the main target is to retrieve content from a modality (e.g., image) given some input content from another modality (e.g., textual description). Several important applications benefit from successful retrieval strategies, such as image and video retrieval, captioning [23, 30], and navigation for the blind.

State-of-the-art results for bidirectional retrieval [5, 10, 29, 31] are based on networks trained over word-embeddings [20] that encode the sentences with either Recurrent Neural Networks (LSTMs [9], GRUs [3]) or with handcrafted non-linear transformations such as Fisher vectors [16]. Despite showing promising results, these strategies present several drawbacks. First, they are costly due to the need of training word-embeddings within a latent space to capture semantic relationships among words. Second, it takes a considerable amount of storage and memory for dealing with these embeddings depending on the size of the dictionary, in which often larger is better in terms of results.

We address those two problems by proposing an approach that does not rely on RNNs or handcrafted transformations over pre-trained word-embeddings. Our architecture learns from scratch, in a character basis, how to retrieve descriptions from images and images from descriptions, and for that it relies exclusively on convolutional layers. It does not make assumptions on specific templates, guidelines, or previous knowledge since it learns everything from scratch using the available training data.

Bearing in mind that bidirectional retrieval can be seen as a special case of a single visual-semantic hierarchy over words, sentences, and images, we employ a loss function based on order embeddings [29], which are designed to model the partial order structure of the visual-semantic hierarchy existing in image descriptions. While typical strategies for bidirectional retrieval rely on distance-preserving strategies, our approach performs order-preserving optimization, making the process of relating the naturally-hierarchical concepts within descriptions much easier.

We perform thorough experiments in order to evaluate multiple aspects of our architecture. In particular, we analyze the impact of using (1) distinct number of filters; (2) depth-wise convolutions; and (3) distinct latent embedding size. Moreover, we measure the robustness of our approach to input noise. We compare our proposed architecture with the current state-of-the-art approaches in the well-known MS COCO [17] retrieval dataset, and we show that it outperforms all other approaches while presenting a much lighter and simpler retrieval architecture. Finally, we show that our models perform well for text classification tasks.

2. CHAIN-VSE

We propose a bidirectional retrieval architecture that relies on novel inception-based [28] modules named **Cha**racter-level **In**ception for **V**isual-**S**emantic **E**mbeddings (CHAIN-VSE). These modules are designed to understand descriptions directly from raw characters (similarly to [4,35]), projecting the sentence semantics onto a $\mathbb{R}^d$ representation that explicitly encodes information in a hierarchical fashion for leveraging both fine-grained and global-level features.

State-of-the-art approaches for multimodal alignment/retrieval are often based on word-embeddings and either RNNs or handcrafted transformations for encoding sentences. Such a strategy presents some drawbacks: (i) it requires training word-embeddings and RNNs in very large corpora (with millions or billions of words), consuming a lot of time and demanding high computational power; (ii) to encode a single word or sentence, it is necessary to have at disposal the whole word-dictionary containing all known words, largely increasing the memory requirements to store all data; (iii) for multilingual or informal domains such as tweets and search queries, the number of words in the dictionary increases with the number of languages; (iv) a preprocessing step is required for correcting typos and standardizing the words.

The idea behind CHAIN-VSE is to replace both word-embeddings and RNNs/handcrafted transformations by a simple yet effective architecture that is exclusively based on character-level convolutions. The advantage of using convolutions instead of RNNs is that one can use parallelism for convolving temporal data, while in RNNs a given temporal iteration depends on the previous one. This allows for much faster computation, especially in GPUs. By employing a character-level textual representation instead of word-level (as in [19]), we can build descriptions across several languages with a small finite set of characters. With word-embeddings, on the other hand, we would need to store thousands or millions of word-vectors.

Our architectures depend on two main hyper-parameters: p, that regulates the number of filters in the convolutional layers, and d, which defines the latent-embedding size. Next, we describe two main variations of our architecture, namely CHAIN-VSE-[v1, v2].

2.1. CHAIN-VSE-v1

Figure 1 depicts the overall structure of the first version of CHAIN-VSE. It consists of two inception modules [28] for capturing distinct granularity levels of the original sentence. The first module maps the binary character-level content from the sentences to a dense representation by convolving the characters with filter sizes $f \in \{7, 5, 3\}$, generating word-embedding-like vectors.

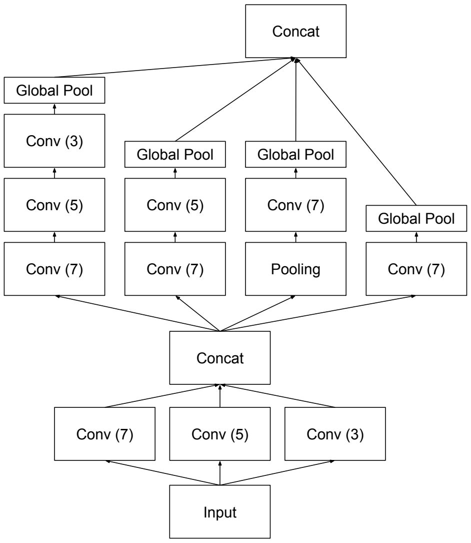

Figure 1. CHAIN-VSE-v1 module.

This dense representation is then fed to the second module, which consists in using four independent convolutional streams. The first stream comprises three convolutional layers with filter sizes $f \in \{7, 5, 3\}$, which are capable of learning relationships among a reduced set of words, depending on the word length. For instance, it could generalize trigram-like features. The second and fourth streams are mostly responsible for understanding fine-grained information (e.g., similar to bigrams and unigrams), being capable of learning from single words and short character sequences (e.g., :), : /, and *goood*). The third stream first performs an average pooling that reduces by half the temporal dimension, and is thus designed to feature-wise average sequences of characters, helping in exploiting the receptive field for learning mid-term dependencies (see Figure 2). The final layer of each convolutional stream within CHAIN-VSE-v1 performs a Max Global Pooling (often called max[average]-over-time-pooling [13]) that summarizes the temporal dimension. Each vector is then concatenated for building the final textual semantic representation. Finally, by adopting an inception-like architecture, our models (with four convolutional streams of 256 filters) present a reduced number of parameters when compared to a similar network built over a single convolutional stream of three layers with 1024 filters.

2.2. CHAIN-VSE-v2

We also designed lighter modules that are based on separable depth-wise convolutions. Figure 3 depicts an example of a separable depth-wise convolutional layer being applied to bidimensional data such as character-based inputs. This layer comprises two steps: (i) a separable convolution, which processes each channel of the input data individually; and (ii) a regular convolution with filter size $f = 1$,

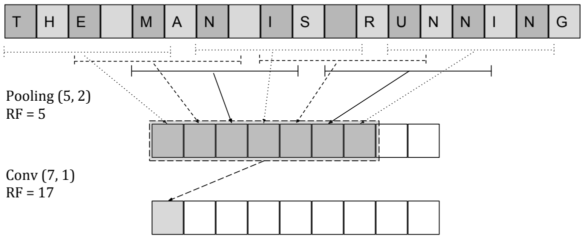

Figure 2. Impact of the pooling layer (filter size of 5 and stride of 2) in the receptive field of a character-based textual representation. For simplicity, we are ignoring the existence of the first inception module.

which merges information across all features. By using this particular strategy, one can basically halve the number of parameters and floating-point operations. Note that this architecture employs separable convolutions after the first inception layer. Hence, convolutions applied directly to the character-level input use filters $7 \times \eta$, where η is the alphabet size.

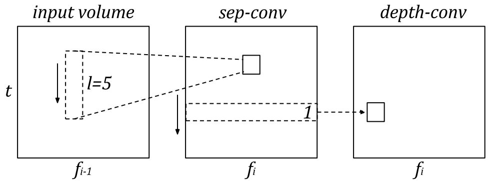

Figure 3. Illustration of a depth-wise separable convolution applied to a bidimensional input of size $t \times f_{t-1}$, where t denotes the size of the temporal dimension (text length), and f_{t-1} is the number of filters of the previous convolutional layer. The filter length in the example is set to $l = 5$, as seen in the second convolutional layers within CHAIN-VSE streams.

Formally, note that convolving a $W \times H$ input with a filter $P \times Q$ implies a complexity of $\mathcal{O}(W \times H \times P \times Q)$. When using a separable filter, the first-step complexity is $\mathcal{O}(W \times H \times P)$, while the second-step is $\mathcal{O}(W \times H \times Q)$, resulting in a total complexity of $\mathcal{O}(W \times H \times (P + Q))$ or $\mathcal{O}(W \times H \times P \times 2)$ if $P = Q$. Therefore, the computational advantage in using separable filters against regular ones is $P \times Q/(P + Q)$. For a 7×7 filter, it results in a theoretical speed-up of 3.5. A final note regarding the use of separable filters is that the application of 1×1 convolutions introduce, *per se*, extra non-linearity to the model, eliminating the need of additional activation functions. Nevertheless, we notice that using Maxout over the feature maps lead to far better performance.

2.3. Overall Architecture

We approximate two encoding functions, $f_t(\mathcal{T})$ and $f_i(\mathcal{I})$, whose goal is to project both description $\mathcal{T}$ and image $\mathcal{I}$ into the same embedding space. In such a space, correlated image-text pairs should be close to each other, and the distance of non-correlated pairs should necessarily be larger than the correlated ones. For the text encoding function $f_t(\mathcal{T})$, we make use of the CHAIN-VSE modules described in Sections 2.1 and 2.2. For the image encoding function $f_i(\mathcal{I})$, we extract image features from the global layer of a ConvNet (VGG-19, Inception-ResNet-v2 [IRv2], or ResNet-152) [8, 26, 27] pre-trained in the ImageNet dataset [25]. Each image $\mathcal{I}$ is then represented by c-long vectors, extracted using the 10-crop strategy. Respectively, VGG-19, IRv2 and ResNet-152 present $c = 4096, 1536, 2048$. Let $\mathcal{C}(\mathcal{I})$ be features extracted from image $\mathcal{I}$ by the convolutional neural network; images are projected onto the $\mathcal{R}_+^d$ embedding-space based on a linear mapping:

$$f_i(\mathcal{I}) = |W_i \cdot \mathcal{C}(\mathcal{I})| \tag{1}$$

where $W_i \in \mathcal{R}^{d \times c}$ is a learned weight matrix and d is the number of dimensions of the embedding space.

For embedding text, we use the proposed character-based approach $f_t(\cdot)$ with the two main variations previously discussed. Our models provide a l-long vector representation that carries the textual semantic information, where l depends on the number of filters used. For instance, CHAIN-VSE-v1 ($p = 1$) produces a $l = 1024$ vector. Similarly to $f_i(\cdot)$, we linearly project such representation onto $\mathcal{R}_+^d$ by using a learned $W_t \in \mathcal{R}^{d \times l}$ weight matrix.

Note that one of the best approaches for image-text alignment [29] makes use of GRU networks fed with word-embeddings, requiring $\approx$ 15M parameters for the word-embeddings (considering $50,000$ words) and $\approx$ 5M parameters for the GRU itself. Our largest architecture contains roughly the same amount of parameters, while the smallest architecture contains only 1.5M parameters, which is more than one order of magnitude lighter. Another top-performing approach employs two-way nets [5], with fully-connected layers that result in more than 100M total parameters, which also depends on pre-trained word-embeddings and handcrafted nonlinear transformations.

2.4. Loss function

Let $f_t(\mathcal{T}) = \mathbf{m}$ be the sentence embedding vector and $f_i(\mathcal{I}) = \mathbf{v}$ be the image embedding. We first scale $\mathbf{m}$ and $\mathbf{v}$ to have unit norm, so that the inner product of both results in the cosine distance. Instead of directly optimizing the cosine distance as in [15], we follow [29] by optimizing the alignment preserving the order relationships among the visual-semantic hierarchy, given that asymmetric distances are naturally more well-suited for image-sentence alignment. Hence, we apply an order-violation constraint by penalizing an ordered pair (x, y) of points in $\mathcal{R}_+^N$:

$$s(x, y) = -|max\{0, y - x\}|^2 \tag{2}$$

The order violation penalties are used as a similarity distance, and optimized by the following constrastive pairwise ranking loss:

$$\mathcal{L} = \sum_{\mathbf{m}} \sum_{k} max\{0, \alpha - s(\mathbf{m}, \mathbf{v}) + s(\mathbf{m}, \mathbf{v}_k)\} + \sum_{\mathbf{v}} \sum_{k} max\{0, \alpha - s(\mathbf{v}, \mathbf{m}) + s(\mathbf{v}, \mathbf{m}_k)\} \quad (3)$$

where $\mathbf{m_k}$ and $\mathbf{v_k}$ are the sentence and image contrastive examples (i.e., uncorrelated). This loss function encourages the similarity $s(x, y)$ for proper image-text pairs to be larger than the contrastive pairs by a margin of at least α.

3. Experimental Setup

3.1. Dataset

For analyzing the performance of our proposed approach, we make use of the Microsoft COCO dataset [17]. We have used the same data splits from [12]: 113,287 images for training, 5,000 images for validation, and 5,000 images for testing.

MS COCO has been extensively employed in the recent years for image-text retrieval challenges. Note that, for the $5k$ images in the test set, there are three distinct evaluation protocols employed by the research community, because the test images were further divided into 5 folds of $1k$ images each. Some studies present results on the entire test set of $5k$ images, a protocol we refer to as COCO-$5k$; others present results only for a subset of $1k$ images, which we refer to as COCO-$1k$; finally, there are studies that present the average result over the 5 folds, referred to as COCO-$5cv$.

3.2. Hyper-Parameters and Training Details

We choose hyper-parameters via non-exhaustive random search based on the results over the validation data. We employ Adam for optimization, given its capacity in adjusting per-weight learning rates during training. We use Adam's default initial learning rate of 1×10^{-3}. In addition, we found it was beneficial to reduce the learning rate by 10× whenever the validation error *plateaus*. Inspired by [29], we use a batch size of 128 (127 contrastive examples) and margin $\alpha = 0.05$. Neither weight decay nor dropout were used, since we believe the loss function itself is enough to regularize the model by including several contrastive examples that naturally inject some amount of noise during training.

The three convolutions in the first inception module comprise 32 filters for all of our models. For the second inception module, the width varies according to p. The default number of convolutional filters for each layer in the second module is set to 256. This means that a network with $p = 1$ has 256 filters in each convolutional layer, while a network with $p = 0.5$ comprises layers with 128 filters. In addition, the default size of the latent multimodal space is set to $d = 1024$. We early-stop the training when the sum of all metrics calculated in the validation data stops improving for 10 consecutive epochs.

3.3. Evaluation Measures

For evaluating the results, we use the same measures as those in [29]: $R@K$ (reads "Recall at K") is the percentage of queries in which the ground-truth term is one of the first K retrieved results. The higher its value, the better. We also show the results of *Med* r and *Mean* r, which represent respectively the median and mean of the ground-truth ranking. Since they are ranking-based measures, the smaller their values the better.

4. Experimental Analysis

In this section, we provide a thorough analysis of the performance of our proposed approach. First, we analyze the impact of different architectural choices for CHAIN-VSE by looking exclusively to results on validation data. Then, we compare our best approach with the state-of-the-art in bidirectional retrieval (results over the test set).

4.1. Impact of p

We also analyze the impact of the hyper-parameter p that regulates the width of the network. We vary $p \in \{0.5, 0.75, ..., 1.50\}$, allowing us to discover the minimum number of neurons that are required for achieving good performance. It is also useful for finding models that present a good *performance-complexity* trade-off.

We report in Table 1 results for both CHAIN-VSE-v1 (top section) and CHAIN-VSE-v2 (bottom section). Both architectures employ Maxout activations. Note that traditional approaches for separable convolutions make use of linear layers (no activations), given that the single-sized depth-wise convolution itself is responsible for increasing the non-linearity of the model. Nevertheless, we achieved top performance by using Maxout activations.

Table 1. Impact of p (width) in CHAIN-VSE. Bidirectional results on MS COCO validation set.

	Image to text		Text to image			
CHAIN-VSE (p)	R@1	Mean r	R@1	Mean r	#Params	Flops $\times 10^{10}$
v1 (0.50)	48.5	4.8	39.7	6.3	1.47M	4.55
v1 (0.75)	48.7	4.4	40.9	6.3	2.52M	8.43
v1 (1.00)	**50.6**	**4.3**	**41.3**	**6.3**	3.79M	13.41
v1 (1.25)	49.2	4.4	40.9	6.9	5.27M	19.48
v1 (1.50)	48.8	4.7	40.6	6.7	6.96M	26.64
v2 (0.50)	46.5	4.8	38.6	6.5	1.16M	2.93
v2 (0.75)	47.0	4.8	39.3	6.7	1.80M	4.75
v2 (1.00)	47.9	4.7	39.4	6.5	2.50M	6.83
v2 (1.25)	48.6	4.5	39.8	6.5	3.26M	9.15
v2 (1.50)	47.7	4.6	40.0	6.6	4.06M	11.73

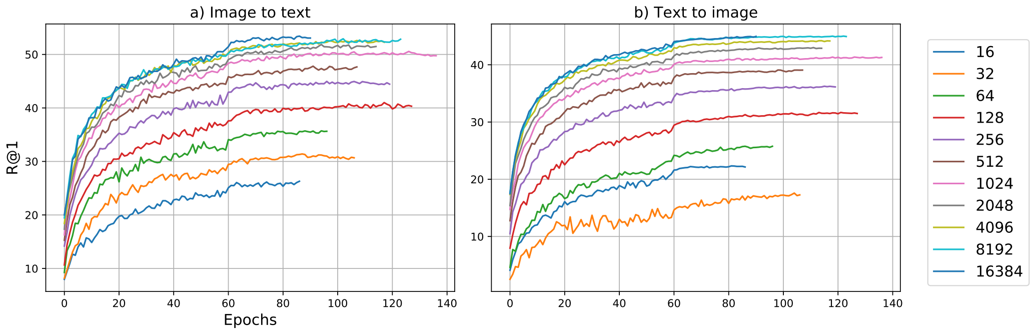

Figure 4. Values of R@1 during training. We compare the impact of latent embedding sizes $d \in \{2^4, 2^6, ..., 2^{14}\}$.

The use of separable convolutions does not seem to be recommended for reaching top performance, though it becomes an interesting option when the goal is to generate faster and lighter models, requiring roughly half the floating-point operations. Also note that using $p = 0.75$ leads to an $\approx 2\%$ drop in terms of $R@1$, but also reducing $\approx 1.6\times$ the required floating-point operations and requiring 1.27M fewer parameters. Models with $p = 0.5$ perform similarly to $p = 0.75$ while being much lighter.

4.2. Impact of the Latent Embedding Size

Our architecture for training both image and text encoders rely on the use of a contrastive loss function that optimizes a metric so that correlated image-text pairs lie close in a multimodal embedding space. As far as we know, this is the first study to investigate the impact of the resulting multimodal embedding size. We trained several models of CHAIN-VSE-v1 by varying $d \in \{2^4, 2^6, ..., 2^{14}\}$. The training behavior can be observed in Figure 4, where we report values of $R@1$ on the validation set across all training epochs. For the *text to image* retrieval task, note that using $d = 2048$ leads to much better results than using the default embedding size (1024). In addition, it seems that $d = 16384$ brings little to no impact when compared to $d = 8192$ for that same task, but it seems to definitely help for the description retrieval. In a nutshell, we conclude that suitable high-performing d values are $512 < d < 16384$.

4.3. Robustness to Input Noise

Our models are built based on character-level inputs. Theoretically, this strategy is much more robust to input noise such as typos and prolonged words (e.g., *thorought*, *gooood*). This advantage is inherent to the fact that using the atomic part of the sentence for learning semantics makes it easier to learn syntactic and semantic similarities across words. Note that input noise is challenging for word-embedding-based approaches. For handling noise in those approaches, we must implement ad-hoc strategies such as dictionary look-ups to correct typos. This poses an unnecessary cost that can be avoided by character-based models.

For evaluating the robustness of our models to input noise, we randomly change a given ratio of the description characters. We compute validation set results ($R@1$ and $R@10$) while varying the noise ratio in the interval $\in \{1\%, 2\%, 3\%, ..., 25\%\}$. We make sure that when the noise ratio is set to $>= 1\%$ at least one character is changed in the original description. This is necessary because MS COCO dataset is mostly comprised of short descriptions.

Figure 5 confirms our hypothesis that character-level models are far more robust to input noise. Our approach only suffers a significant drop in $R@10$ when we set the noise ratio to $> 15\%$ (i.e., 15% of the original text is randomly changed). On the other hand, the word-embedding based approach by Vendrov et al. [29] presents a dramatic drop of both evaluation measures even for very small amounts of noise. This result was expected given that by changing a single character, the word may not be found in the trained word-embedding.

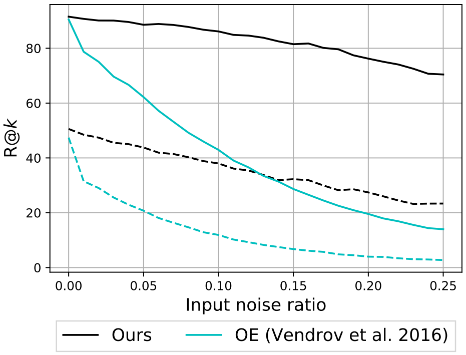

Figure 5. Analysis of performance given random input noise. Continuous lines depict $R@10$ values whereas dotted lines depict $R@1$ values.

4.4. CHAIN-VSE *vs.* State-of-the-art

For comparing our models with the state-of-the-art, we selected those models from each CHAIN-VSE's variation that presented the best performance on validation data. Our models are compared to the published state-of-the-

Table 2. Bidirectional results on COCO-$1k$ test set. Bold values indicate the current state-of-the-art results. Underlined values outperform the best published results.

		Image to text					Text to image				
Method	ConvNet	R@1	R@5	R@10	Med r	Mean r	R@1	R@5	R@10	Med r	Mean r
m-CNN (Ens) [19]	VGG-19	42.80	-	84.10	2.00	-	32.60	-	82.80	3.00	-
UVS [15]	VGG-19	43.40	-	85.80	2.00	-	31.00	-	79.90	3.00	-
Embedding Network [31]	VGG-19	50.40	79.30	89.40	-	-	39.80	75.30	86.80	2.00	-
sm-LSTM [10]	VGG-19	52.40	81.70	90.80	**1.00**	-	38.60	73.40	84.60	2.00	-
2WayNet [5]	VGG-19	55.80	-	75.20	-	-	39.70	-	63.30	-	-
Order (d=1024)	VGG-19	47.80	81.60	91.30	2.00	4.90	39.40	75.70	87.00	2.00	7.20
Order (d=8192)	VGG-19	52.00	82.30	92.20	**1.00**	4.60	41.30	76.40	87.60	2.00	6.90
Order (d=1024)	IRv2	49.10	80.90	90.30	**1.00**	5.30	39.80	75.00	86.80	2.00	7.30
Order (d=8192)	IRv2	50.60	81.80	90.80	**1.00**	4.60	40.10	75.90	87.50	2.00	6.90
CHAIN-VSE-v1 (d=1024, p=1)	VGG-19	53.10	82.40	91.50	**1.00**	5.20	38.30	75.10	86.60	2.00	6.70
CHAIN-VSE-v1 (d=8192, p=1)	VGG-19	54.00	83.70	91.80	**1.00**	5.00	40.30	76.40	87.80	2.00	6.10
CHAIN-VSE-v1 (d=1024, p=1)	IRv2	52.50	84.40	92.50	**1.00**	4.00	41.10	77.80	89.90	2.00	5.90
CHAIN-VSE-v1 (d=8192, p=1)	IRv2	55.80	87.00	94.90	**1.00**	3.40	42.60	79.20	90.40	2.00	5.40
RFF-Net [18]	ResNet-152	56.40	85.30	91.50	-	-	43.90	78.10	88.60	-	-
Order (d=2084)	ResNet-152	55.00	86.70	94.50	**1.00**	3.40	43.30	79.70	89.20	2.00	6.30
CHAIN-VSE-v1 (d=1024, p=1)	ResNet-152	57.80	87.90	95.60	**1.00**	3.25	44.18	80.40	90.66	2.00	5.39
CHAIN-VSE-v1 (d=2048, p=1)	ResNet-152	59.90	**89.50**	94.80	**1.00**	3.18	45.08	80.64	90.54	2.00	5.76
CHAIN-VSE-v1 (d=8192, p=1)	ResNet-152	**61.20**	89.30	**95.80**	**1.00**	**2.85**	**46.60**	**81.90**	**90.92**	2.00	**5.21**

art approaches for multimodal retrieval, namely UVS [15], DVSA [12], FV [16], m-CNN [19], Order-Embeddings (OE) [29], Embedding Network [31], sm-LSTM [10], 2WayNet [5], SEAM-C [32], and RFF-Net [18]. For providing a fair comparison, we replicated the OE [29] results using their own source code, both with their default parameters and also with varied latent embedding sizes.

Note that several baselines employ text representation based on word-embeddings, which inflicts a minimum memory cost of $|\mathcal{V}| \times |\mathcal{D}|$, where $\mathcal{V}$ is the vocabulary that contains all known words and D is the word-vector. For instance, the vocabulary of the MS COCO dataset contains roughly 50,000 words (as in [29]), which inflicts a minimum cost of $50,000 \times 300$ (assuming $|\mathcal{D}| = 300$). On the other hand, our approach is capable of fully learning compact vectors for sentence encoding, not requiring pre-trained dictionary-based vectors nor handcrafted transformations. Note that our approach presents a constant data cost that is related to the alphabet size alone, i.e., it is independent of the number of words, which is an interesting property especially for multilingual learning (see Section 5 for results in multilingual and noisy data learning).

Table 2 presents results for the best selected versions of CHAIN-VSE alongside the state-of-the-art approaches when considering COCO-$1k$. Note that CHAIN-VSE's version using ResNet-152, $d = 8192$, and $p = 1$ establishes itself as the new state-of-the-art for both description and image retrieval tasks, outperforming RFF-Net by $\approx 6.1\%$ for image-to-text task in absolute R@1 values. Whereas using a better ConvNet definitely helps in achieving state-of-the-art results, note that the versions of CHAIN-VSE that employ either a IRv2 or a VGG-19 also outperform all baselines for all evaluation measures and retrieval tasks, with the exception of R@1 in the description retrieval task, where it is outperformed by 2WayNet (with a VGG-19). However, note that 2WayNet performs poorly regarding R@10, and that all versions of CHAIN-VSE are far superior in the image retrieval task. In addition, CHAIN-VSE is about two orders of magnitude lighter than 2WayNet.

In Table 3, we present results for the best selected versions of CHAIN-VSE along OE [29] (and its modified versions), OECC [34], and SEAM-C [32], since those studies explicitly provide average results on COCO-$5cv$. Once again CHAIN-VSE ($d \in \{4096, 8192\}$), provides superior results for all evaluation measures and retrieval tasks regardless of the ConvNet it uses.

For better visualizing the trade-off between model complexity and predictive performance of CHAIN-VSE, we present in Figure 6 the effect of varying the amount of parameters versus R@1, and how CHAIN-VSE compares to OE [29] and Embedding Network [31], which are the two baselines that present the smallest amount of parameters. The ideal position is in the upper-left position (largest recall and smallest amount of parameters). For generating this visualization, we computed the number of trainable parameters for each method. Note that the word-embeddings are considered trainable parameters as well, considering all hyper-parameters and settings defined in [29] and [31].

Note that our models in Figure 6 present fewer parameters than the baselines. There are several models that are almost an order of magnitude lighter while presenting better results for both tasks. Our models seem to respond much better to larger embedding sizes than OE [29]. Moreover, using features from IRv2 provides a large gain for CHAIN-VSE, whereas that does not seem to be the case for OE. Finally, 2WayNet [5] is not included in Figure 6 since it has about two orders of magnitude more parameters than our models.

Table 3. Bidirectional results on COCO-5cv test set. Bold values indicate the current state-of-the-art results.

		Image to text					Text to image				
Method	ConvNet	R@1	R@5	R@10	Med r	Mean r	R@1	R@5	R@10	Med r	Mean r
Order [29]	VGG-19	46.70	-	88.90	2.00	-	37.90	-	85.90	2.00	-
OECC [34]	VGG-19	47.20	78.60	88.90	2.00	5.60	37.50	74.60	87.00	2.00	7.30
SEAM-C [32]	VGG-19	50.70	81.40	90.90	1.40	4.90	40.30	75.70	87.40	2.00	7.40
Order (d=1024)	VGG-19	46.20	78.80	89.10	2.00	5.40	37.70	73.40	85.60	2.00	7.90
Order (d=4096)	VGG-19	48.80	79.20	89.70	1.60	5.20	38.70	74.10	86.40	2.00	7.60
Order (d=8192)	VGG-19	50.10	80.20	90.30	1.40	5.10	39.10	74.40	86.30	2.00	7.60
CHAIN-VSE-v1 (d=1024, p=1)	VGG-19	49.50	80.80	90.00	1.60	5.30	36.80	73.60	85.90	2.00	7.30
CHAIN-VSE-v1 (d=4096, p=1)	VGG-19	52.00	82.30	90.70	1.20	5.00	38.30	74.80	87.00	2.00	6.80
CHAIN-VSE-v1 (d=8192, p=1)	VGG-19	51.60	82.00	91.30	1.20	4.70	38.60	75.10	87.20	2.00	6.70
OECC [34]	IRv2	49.50	81.70	91.30	1.60	4.50	40.40	77.40	88.60	2.00	6.80
Order (d=1024)	IRv2	47.30	78.60	88.70	1.80	5.50	37.70	73.10	85.50	2.00	7.80
Order (d=4096)	IRv2	49.10	79.40	89.50	1.40	5.20	38.20	74.50	86.50	2.00	7.60
Order (d=8192)	IRv2	50.20	79.50	89.20	1.20	5.30	38.20	74.20	86.30	2.00	7.40
CHAIN-VSE-v1 (d=1024, p=1)	IRv2	50.50	83.60	92.20	1.60	4.30	39.00	76.20	88.10	2.00	6.80
CHAIN-VSE-v1 (d=4096, p=1)	IRv2	52.80	84.40	92.60	**1.00**	4.10	40.70	77.40	88.90	2.00	6.50
CHAIN-VSE-v1 (d=8192, p=1)	IRv2	53.70	85.10	93.10	**1.00**	3.90	40.70	77.60	89.00	2.00	6.30
CHAIN-VSE-v1 (d=1024, p=1)	ResNet-152	55.14	86.08	93.86	**1.00**	3.76	41.20	78.01	89.22	2.00	6.38
CHAIN-VSE-v1 (d=4096, p=1)	ResNet-152	57.76	87.88	**94.46**	**1.00**	3.42	42.96	79.20	90.01	2.00	6.05
CHAIN-VSE-v1 (d=8192, p=1)	ResNet-152	**59.40**	**87.98**	94.24	**1.00**	**3.37**	**43.47**	**79.78**	**90.22**	2.00	**5.90**

Table 4. Impact of text length. d=2048, cnn=ResNet152.

		Image to text		Text to image	
Method	Length	R@1	Mean r	R@1	Mean r
OE	100 longest	86.0	1.3	76.4	1.5
CHAIN-VSE	100 longest	92.0	1.3	74.2	1.6
OE	100 shortest	85.0	1.4	69.8	1.7
CHAIN-VSE	100 shortest	80.0	1.6	81.0	1.5

Table 5. Text classificaton results.

Method	Tweets	AgNews	DBPedia
Conv [13, 33]	71.8%	-	-
ConvChar [35]	70.6%	87.2%	98.3%
FastText [11]	71.3%	**91.5%**	98.1%
VCDNN [4]	-	91.3%	**98.7%**
CHAIN-VSE-v1	**73.5%**	**91.5%**	98.6%
CHAIN-VSE-v2	**72.1%**	91.0%	98.2%

Impact of text length. We also evaluate the impact of the text length in our retrieval results. In order to accomplish that, we separate 100 images that present the longest and shortest captions. Results are shown in Table 4. Our findings are twofold: (i) CHAIN-VSE is much better for annotating images with long captions; and (ii) it works better for retrieving images given short captions, probably due to the fact that our approach explicitly exploits short and mid-term aspects of the sentences. Finally, it seems to be hard for both methods to retrieve the correct images given long captions. This might be due to the data distribution in MS COCO, since it presents mostly short captions.

4.5. Limitations

CHAIN-VSE's main limitation comes from the fact that it learns textual features from scratch rather than using external corpora for learning word semantics: CHAIN-VSE may suffer from overfitting when dealing with smaller datasets, such as Flickr30k [22]. Apparently, those datasets do not present enough textual data to properly learn textual semantics from raw characters. We achieve $R@1 \approx 36$ (40) for caption retrieval and $R@1 \approx 26$ (31) for image retrieval using VGG-19 (IRv2), which outperforms [14, 16, 19, 29], but is outperformed by some recent approaches that employ external corpora to some extent.

5. Text Classification

We designed CHAIN-VSE in order to provide a simple yet efficient and robust method for learning textual semantics directly from characters. One of our findings was that CHAIN-VSE is far more robust to noise than state-of-the-art approaches. In this section we provide an analysis regarding the suitability of CHAIN-VSE for learning multi-language sentiment analysis models from noisy data and in widely used text classification datasets.

We test CHAIN-VSE on a multi-language Twitter corpora as well as on the widely used AGNews and DBPedia datasets. The Twitter corpora from [21] does not provide the tweet itself, but rather a URL that leads to the tweets. Due to this particularity, some tweets are no longer available. We adapt CHAIN-VSE by replacing the multimodal embedding layer with a fully-connected softmax layer for performing the final classification. Since the data used in this experiment contains about 5× fewer textual instances than MS COCO, we optimized the hyper-parameters for properly regularizing the network.

We optimized the width of the network, the latent space, regularization, and activation function. CHAIN-VSE is by far the lightest method in terms of parameters (DBPedia and Tweets require a vocabulary of 200,000 words). Our total

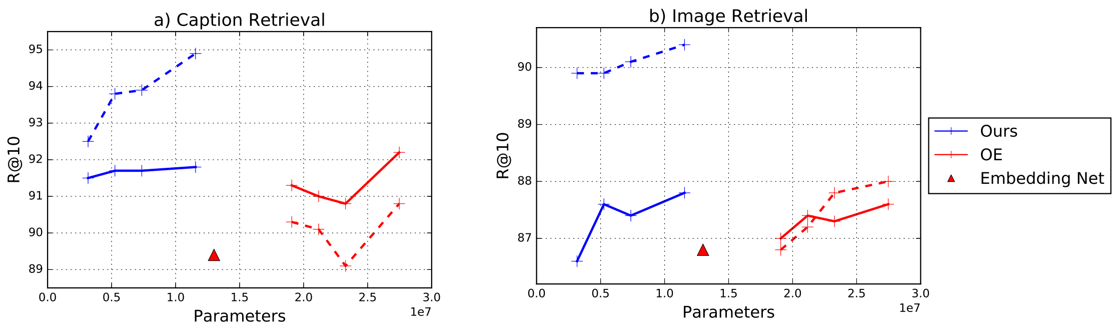

Figure 6. Trade-off between model complexity (#parameters) and predictive performance (R@10) in the COCO-1k test set. Results are for both image-to-text (left) and text-to-image (right) retrieval. Dashed lines are architectures that make use of IRv2 networks whereas continuous lines are architectures that use VGG-19. Each point in the line indicate a distinct latent embedding size. From left to right: $d = 1024, 2048, 4096, 8192$.

architecture comprises about 1.5M parameters, 400× fewer parameters than FastText [11] and Conv [13]. It is also 10× lighter than VCDNN [4].

6. Related Work

Karpathy and Fei-Fei [12] propose an architecture that makes use of features from detection-based systems, aligning image regions with a proper sentence fragment. Ma et al. [19] propose a multimodal ConvNet for aligning image and text by jointly convolving word-embeddings and image features. The learned similarity score predicts whether a pair is correlated or not. Vendrov et al. [29] propose sentence order-embeddings, which aim to preserve the partial order structure of a visual-semantic hierarchy. It allows learning ordered representation by applying order-penalties, and they show that asymmetric measures are better suited for image-sentence retrieval tasks.

Wang et al. [31] introduce a two-branch neural network for learning a multimodal embedding space. They encode text based on 300-d word-embeddings, where they apply ICA and construct a codebook with 30 centers using first and second-order information, resulting in a $18k$-dimensional representation. Next, they apply PCA to reduce the representation to $6k$ dimensions in order to reduce memory requirements and training time. The projection into the joint space is performed with dense layers.

Huang et al. [10] propose a selective multimodal LSTM (sm-LSTM). They introduce a multimodal context-modulated attention scheme at each time-step, which is capable of focusing on a text-image pair by predicting pairwise instance-aware saliency maps. Sentences are processed by a bidirectional LSTM that runs over word-embeddings. Image features are selected by using a strategy of instance candidates, which extracts local information from a $512 \times 14 \times 14$ tensor. They also make use of the 4096-d vectors for a global image representation, leveraging local and global information from both text and image.

In [5], the authors introduce a 2-Way-Network for mapping a modality into another. Similarly to [31], they use Fisher Vectors applied over $word2vec$ for sentence encoding. They concatenate the Fisher Vector encoding (GMM) and the Fisher Vector of the HGLMM distribution, resulting in a $36k$-dimensional vector per sentence.

In [32], the authors propose a fast approach for multimodal retrieval. They employ a self-attention mechanism in order to embed word-vectors onto a sentence-level embedding space. They achieved good results while their models were much faster for training and deployment than the RNN-based approaches.

7. Conclusions

In this paper, we presented a simple architecture for bidirectional retrieval capable of learning textual embedding based on raw characters, namely CHAIN-VSE. Even though it is conceptually a much simpler architecture than those found in related work, our approach achieves state-of-the-art results in both *text to image* and *image to text* tasks considering the most well-known retrieval dataset, namely MS COCO [17]. CHAIN-VSE is simple, effective, requires fewer parameters, and it is robust to input noise due to the fact that it learns sentence representation from character-level convolutions. In addition, it presents sound performance for text classification tasks, specially in noisy and multilingual scenarios. For future work, we intend to analyze the impact of CHAIN-VSE in recent work [6,7] and its performance in other tasks, such as VQA [1, 2] and video retrieval [24].

8. Acknowledgments

We thank Google, Motorola, and the Brazilian research agencies CAPES, CNPq, and FAPERGS for funding this research. We also acknowledge the support of NVIDIA Corporation for the donation of the GPUs used in this work.

References

[1] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. VQA: Visual Question Answering. In *International Conference on Computer Vision (ICCV)*, 2015.

[2] H. Ben-Younes, R. Cadène, N. Thome, and M. Cord. Mutan: Multimodal tucker fusion for visual question answering. *ICCV*, 2017.

[3] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Gated feedback recurrent neural networks. In *Proceedings of the 32nd International Conference on Machine Learning*, ICML'15, pages 2067–2075, 2015.

[4] A. Conneau, H. Schwenk, L. Barrault, and Y. Lecun. Very deep convolutional networks for text classification. In *EACL*, volume 1, pages 1107–1116, 2017.

[5] A. Eisenschtat and L. Wolf. Linking image and text with 2-way nets. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[6] F. Faghri, D. J. Fleet, R. Kiros, and S. Fidler. VSE++: improved visual-semantic embeddings. *CoRR*, abs/1707.05612, 2017.

[7] J. Gu, J. Cai, S. Joty, L. Niu, and G. Wang. Look, imagine and match: Improving textual-visual cross-modal retrieval with generative models. *arXiv preprint arXiv:1711.06420*, 2017.

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

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

[10] Y. Huang, W. Wang, and L. Wang. Instance-aware image and sentence matching with selective multimodal lstm. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[11] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*, 2016.

[12] 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 (CVPR 2015)*, pages 3128–3137, 2015.

[13] Y. Kim. Convolutional neural networks for sentence classification. In *In EMNLP*. Citeseer, 2014.

[14] R. Kiros, R. Salakhutdinov, and R. Zemel. Multimodal neural language models. In *Proceedings of the 31st International Conference on Machine Learning*, pages 595–603, 2014.

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

[16] B. Klein, G. Lev, G. Sadeh, and L. Wolf. Associating neural word embeddings with deep image representations using fisher vectors. In *CVPR*, pages 4437–4446, 2015.

[17] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: common objects in context. In *European Conference on Computer Vision*, 2014.

[18] Y. Liu, Y. Guo, E. M. Bakker, and M. S. Lew. Learning a recurrent residual fusion network for multimodal matching.

[19] L. Ma, Z. Lu, L. Shang, and H. Li. Multimodal convolutional neural networks for matching image and sentence. In *International Conference on Computer Vision (ICCV)*, 2015.

[20] 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*, pages 3111–3119, 2013.

[21] I. Mozetič, M. Grčar, and J. Smailović. Multilingual twitter sentiment classification: The role of human annotators. *PloS one*, 11(5):e0155036, 2016.

[22] B. A. Plummer, L. Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier, and S. Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. *Int. J. Comput. Vision*, 123(1):74–93, May 2017.

[23] A. Rohrbach, A. Torabi, M. Rohrbach, N. Tandon, C. Pal, H. Larochelle, A. Courville, and B. Schiele. Movie description. *International Journal of Computer Vision*, 123(1):94–120, 2017.

[24] A. Rohrbach, A. Torabi, M. Rohrbach, N. Tandon, C. Pal, H. Larochelle, A. Courville, and B. Schiele. Movie description. *International Journal of Computer Vision*, 2017.

[25] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[26] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[27] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. 2017.

[28] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9, 2015.

[29] I. Vendrov, R. Kiros, S. Fidler, and R. Urtasun. Order-embeddings of images and language. In *International Conference on Learning Representations (ICLR 2016)*, 2016.

[30] S. Venugopalan, L. A. Hendricks, M. Rohrbach, R. Mooney, T. Darrell, and K. Saenko. Captioning images with diverse objects. In *CVPR*, 2017.

[31] L. Wang, Y. Li, and S. Lazebnik. Learning deep structure-preserving image-text embeddings. In *CVPR*, pages 5005–5013, 2016.

[32] J. Wehrmann, M. Armani, M. D. More, and R. C. Barros. Fast self-attentive multimodal retrieval. *IEEE Winter Conf. on Applications of Computer Vision (WACV'18)*, 2018.

[33] J. Wehrmann, W. Becker, H. E. Cagnini, and R. C. Barros. A character-based convolutional neural network for language-agnostic twitter sentiment analysis. In *Neural Networks (IJCNN), 2017 International Joint Conference on*, pages 2384–2391. IEEE, 2017.

[34] J. Wehrmann, A. Mattjie, and R. C. Barros. Order embeddings and character-level convolutions for multimodal alignment. *Pattern Recognition Letters*, 102:15–22, 2018.

[35] X. Zhang, J. Zhao, and Y. LeCun. Character-level convolutional networks for text classification. In *NIPS*, pages 649–657, 2015.