Pattern Recognition Letters 102 (2018) 15–22

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Order embeddings and character-level convolutions for multimodal alignment

Jônatas Wehrmann, Anderson Mattjie, Rodrigo C. Barros*

Pontifícia Universidade Católica do Rio Grande do Sul, Av. Ipiranga, 6681, Porto Alegre, RS, 90619-900, Brazil

ARTICLE INFO

Article history:
Received 7 June 2017

MSC:
41A05
41A10
65D05
65D17

Keywords:
Character-level convolutions
Multimodal alignment
Order embeddings
Content retrieval

ABSTRACT

With the novel and fast advances in the area of deep neural networks, several challenging image-based tasks have been recently approached by researchers in pattern recognition and computer vision. In this paper, we address one of these tasks, which is to match image content with natural language descriptions, sometimes referred as multimodal content retrieval. Such a task is particularly challenging considering that we must find a semantic correspondence between captions and the respective image, a challenge for both computer vision and natural language processing areas. For such, we propose a novel multimodal approach based solely on convolutional neural networks for aligning images with their captions by directly convolving raw characters. Our proposed character-based textual embeddings allow the replacement of both word-embeddings and recurrent neural networks for text understanding, saving processing time and requiring fewer learnable parameters. Our method is based on the idea of projecting both visual and textual information into a common embedding space. For training such embeddings we optimize a contrastive loss function that is computed to minimize order-violations between images and their respective descriptions. We achieve state-of-the-art performance in the most well-known image-text alignment datasets, namely Microsoft COCO, Flickr8k, and Flickr30k, with a method that is conceptually much simpler and that possesses considerably fewer parameters than current approaches.

1. Introduction

To learn proper feature representation of input data is an essential part of any machine learning problem, as it directly impacts in the precision of the generated data-based models. Thanks to the fast pace in which computer hardware has been evolving in the last decade, along with the rapid development of computer vision and natural language processing technologies, great advances have been made in tasks that require a huge amount of computational power, in particular those that benefit from optimization-based approaches such as deep neural networks. One of these tasks, image-text alignment, has become an important problem in the latest years as it has many applications such as image and video retrieval, captioning, navigation for the blind, and many others. By successfully mapping image-to-text and text-to-image, we can significantly impact the general and broad task of information retrieval.

For this multimodal content retrieval task, state-of-the-art results [14,22,23] rely on either Long Short-Term Memory networks (LSTMs) [5] or Gated Recurrent Units (GRUs) [2] with word-embeddings [16]. Although such an approach has shown promising results, it has some drawbacks that are important to consider. First of all, it is costly due to the need of pre-training a word-embedding with a latent space informative enough to capture semantic relationships among words. Second, it takes a considerable amount of storage and memory for dealing with these word embeddings, depending on the size of the dictionary, in which often larger is better in terms of predictive accuracy.

By taking into consideration the aforementioned drawbacks, we propose a model that, instead of relying in similar recurrent LSTM/GRU-based architectures that depend on pre-trained word-embeddings, learns from scratch, character per character, how to retrieve captions from images and images from captions by making use of convolutional layers alone. Additionally, our model makes no assumptions regarding specific templates, guidelines, or classifications since it learns everything from scratch using the training data. Since image captioning can be seen as a special case of a single visual-semantic hierarchy over words, sentences, and images, we design a loss function based on the so-called order embeddings [21]. This particular type of embeddings are designed for explicitly modeling the partial order structure of the visual-semantic hierarchy existent in image captions. While typical approaches for image captioning rely on mapping words to vectors

* Corresponding author.
E-mail address: rodrigo.barros@pucrs.br (R.C. Barros).

https://doi.org/10.1016/j.patrec.2017.11.020
0167-8655/© 2017 Elsevier B.V. All rights reserved.

in a distance-preserving fashion [19], we believe order embeddings to be more suitable since the resulting mapping is not distance-preserving but order-preserving between the hierarchy and a partial order over the embedding space, making it easier to relate the naturally-hierarchical concepts within image captions.

In order to evaluate the performance of our model, we execute a series of experiments where we make particular architectural changes to the convolutional neural network by increasing the number of convolutional layers up to 5 and by changing the number of filters. We compare our proposed approach with the current state-of-the-art, and we show that our method achieves state-of-the-art results while often presenting a much lighter, simpler, and easier-to-train architecture.

This paper is organized as follows. Section 2 presents in detail our novel approach for multimodal content retrieval. Section 3 describes the methodology that we adopt for performing the experimental analysis, and the results are discussed in Section 4. Section 5 reviews related work, and we end this paper with our conclusions and future work directions in Section 6.

2. Character-based language embeddings

The use of word-embeddings [16] for text understanding has become a standard approach, being largely used across several tasks such as text classification, machine translation, image and video captioning, and information retrieval. Such an approach consists of representing a given word into a multidimensional latent space, $w \in \mathcal{R}^d$. These embeddings are often projected into a distance-preserving Euclidean space, in which semantic regularities can be easily identified and even manipulated. An example of manipulation of this distance-preserving word-embedding space is the following vector operation over the corresponding words:

$$king \ - man \ + woman \sim queen \tag{1}$$

Current state-of-the-art sentence embedding approaches [6,9,10,15,21] have demonstrated similar results when using RNNs and word-embeddings for encoding entire sentences into a d-dimensional embedding space. Nevertheless, it is important to emphasize that even though word-embeddings are employed in the current state-of-the-art approaches, they tend to present three major drawbacks: (i) they require pre-training word-embeddings or RNNs in very large corpuses (with millions or billions of words), which demands both time and computational power; (ii) in order to encode a single word or sentence, it is necessary to have “at hand” the whole word-dictionary containing all known words, which largely increases the memory requirements to store all data; (iii) for multilingual or informal domains (e.g., twitter and internet searches) the number of words in the dictionary increases according to the number of languages and, in addition, preprocessing is often employed for correcting typos and standardizing the words.

In this work, we propose a novel architecture for learning textual embeddings based on convolving raw characters. Our approach is designed to be simple, efficient, and fast, though still capable of generating state-of-the-art results. Our hypothesis is that a convolutional layer is capable of learning a proper latent embedding space for encoding text semantics. Hence, we replace the word dictionary by applying f convolutional filters over the input text in a temporal window of size t. Moreover, to keep t unchanged through the layer computation, we perform padded convolutions. The resulting feature map $\in \mathcal{R}^{t \times f}$ can be seen as a text-embedding where each character is projected onto a f-dimensional space by considering its relation to the adjacent characters (defined by the filters' length). Such an approach does not require fixed-sized dictionaries nor preprocessing of any kind, and its complexity does not raise with the complexity or availability of text.

2.1. Character-based convolutions

The core idea behind our approach is to replace the use of both RNNs and word dictionaries by applying convolutions to learn textual embeddings. A given text is represented by $\mathcal{T} \in \{0, 1\}^{n \times a}$, where n is the number of characters in the text and a is the alphabet size. Note that n is variable and changes according to the available text, whereas a must be unchanged given that the alphabet contains all known characters. We convolve characters of the input text $\mathcal{T}$ by applying f convolutional filters of length l, where the jth filter in the ith convolutional layer generates feature map $\mathcal{F}_{ij}$ whose xth position is given by :

$$\mathcal{F}^x_{ij} = \phi\left(b_{ij} + \sum_{m=0}^{f_{i-1}} \sum_{p=0}^{l-1} w^p_{ijm} \mathcal{F}^{(x+p)}_{(i-1)m}\right) \tag{2}$$

where ϕ is an activation function, b_{ij} is the bias for the respective convolutional filter, m iterates over the feature maps (channels), p indexes the position of the kernel, w^p_{ijm} is the filter weight and $\mathcal{F}^{x+p}_{(i-1)m}$ is the value of the previous feature map (or input). Note that m iterates over the alphabet size for the case of the first convolution, and over the f_i feature maps of the previous layer for the subsequent convolutions.

A known restriction of applying a single convolutional layer for embedding texts is the size of the receptive field. A convolutional filter of length $l = 7$ is capable of learning information of 7 neighboring characters. This size is probably enough for learning word-based information. Standard strategies for allowing the global learning of the whole text include: (i) increasing l, which leads to an exponential growth of parameters, eventually making the learning unfeasible for large-size texts; (ii) adding more convolutional layers, hence requiring more processing resources and parameters depending on the number of filters of the subsequent convolutions; and (iii) using local or global pooling layers. In order to keep our architecture compact, we explore up to five convolutional layers and a maximum filter size of 7. The final text-embedding vectors are generated by applying a max-pooling-over-time layer, which selects the most important features across the temporal dimension of feature map $\mathcal{F}$. Note that the number of filters in the last convolutional layer defines the length of the embedding vector. Fig. 1 presents a schematic of text-embedding via character-based convolutions, which we have named “convolutions-through-time” (CTT).

2.2. Architecture

Our architecture is designed to approximate two encoding functions, $f_t(\mathcal{T})$ and $f_i(\mathcal{I})$, whose goal is to project both text $\mathcal{T}$ 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 CTT module described in the previous section. For the image encoding function $f_i(\mathcal{I})$, we extract image features from two deep networks pre-trained in the ImageNet dataset [17]: VGG-19 [18] and Inception-Resnet-v2 (IRv2) [20]. From VGG-19 we extract the second fully-connected layer, which provides 4096-d vectors, while for Inception-Resnet-v2 we extract 1536-d vectors from the Global Pooling layer. For better feature representation, we use the 10-crop strategy: we scale the smallest size of each image to 256-pixels (VGG) and 340-pixels (IRv2), respectively, and we sample 224 × 224 (VGG) and 299 × 299 (IRv2) crops from the corners, center, and horizontal mirroring. Finally, features from all crops are averaged element-wise.

Let $\mathcal{C}(\mathcal{I})$ be features extracted from image $\mathcal{I}$ by the convolutional neural network. Images are projected onto the $\mathcal{R}^d_+$