

Hierarchical Attentional Hybrid Neural Networks for Document Classification

Jader Abreu*, Luis Fred*, David Macêdo, and Cleber Zanchettin

Centro de Informática
Universidade Federal de Pernambuco
50.740-560, Recife, PE, Brazil
`{jaoa,lfgs,dlm,cz}@cin.ufpe.br`

Abstract. Document classification is a challenging task with important applications. The deep learning approaches to the problem have gained much attention recently. Despite the progress, the proposed models do not incorporate the knowledge of the document structure in the architecture efficiently and not take into account the contexting importance of words and sentences. In this paper, we propose a new approach based on a combination of convolutional neural networks, gated recurrent units, and attention mechanisms for document classification tasks. The main contribution of this work is the use of convolution layers to extract more meaningful, generalizable and abstract features by the hierarchical representation. The proposed method in this paper improves the results of the current attention-based approaches.

Keywords: Text classification · Attention mechanisms · Document classification · Convolutional Neural Networks.

1 Introduction

Text classification is one of the most classical and important tasks in the machine learning field. The document classification, which is essential to organize documents for retrieval, analysis, and curation, is traditionally performed by classifiers such as Support Vector Machines or Random Forests. As in different areas, the deep learning methods are presenting a performance quite superior to traditional approaches in this field [5]. Deep learning is also playing a central role in Natural Language Processing (NLP) through learned word vector representations. It aims to represent words in terms of fixed-length, continuous and dense feature vectors, capturing semantic word relations: similar words are close to each other in the vector space.

In most NLP tasks for document classification, the proposed models do not incorporate the knowledge of the document structure in the architecture efficiently and not take into account the contexting importance of words and sentences. Much of these approaches do not select qualitative or informative words

* Authors contributed equally and are both first authors.

and sentences since some words are more informative than others in a document. Moreover, these models are frequently based on recurrent neural networks only [6]. Since CNN has leveraged strong performance on deep learning models by extracting more abundant features and reducing the number of parameters, we guess it not only improves computational performance but also yields better generalization on neural models for document classification.

A recent trend in NLP is to use attentional mechanisms to modeling information dependencies without regard to their distance between words in the input sequences. In [6] is proposed a hierarchical neural architecture for document classification, which employs attentional mechanisms, trying to mirror the hierarchical structure of the document. The intuition underlying the model is that not all parts of a text are equally relevant to represent it. Further, determining the relevant sections involves modeling the interactions and importance among the words and not just their presence in the text.

In this paper, we propose a new approach for document classification based on CNN, GRU [4] hidden units and attentional mechanisms to improve the model performance by selectively focusing the network on essential parts of the text sentences during the model training. Inspired by [6], we have used the hierarchical concept to better representation of document structure. We call our model as Hierarchical Attentional Hybrid Neural Networks (HAHNN). We also used temporal convolutions [2], which give us more flexible receptive field sizes. We evaluate the proposed approach comparing its results with state-of-the-art models and the model shows an improved accuracy.

2 Hierarchical Attentional Hybrid Neural Networks

The HAHNN model combines convolutional layers, Gated Recurrent Units, and attention mechanisms. Figure 1 shows the proposed architecture. The first layer of HAHNN is a pre-processed word embedding layer (black circles in the Figure 1). The second layer contains a stack of CNN layers that consist of convolutional layers with multiple filters (varying window sizes) and feature maps. We also have performed some trials with temporal convolutional layers with dilated convolutions and gotten promising results. Besides, we used Dropout for regularization. In the next layers, we use a word encoder applying the attention mechanism on word level context vector. In sequence, a sentence encoder applying the attention on sentence-level context vector. The last layer uses a Softmax function to generate the output probability distribution over the classes.

We use CNN to extract more meaningful, generalizable and abstract features by the hierarchical representation. Combining convolutional layers in different filter sizes with both word and sentence encoder in a hierarchical architecture let our model extract more rich features and improves generalization performance in document classification. To obtain representations of more rare words, by taking into account subwords information, we used FastText [3] in the word embedding initialization.

We investigate two variants of the proposed architecture. There is a basic version, as described in Figure 1, and there is another which implements a TCN [2]

layer. The goal is to simulate RNNs with very long memory size by adopting a combination of dilated and regular convolutions with residual connections. Dilated convolutions are considered beneficial in longer sequences as they enable an exponentially larger receptive field in convolutional layers. More formally, for a 1-D sequence input $\mathbf{x} \in \mathbb{R}^n$ and a filter $f : \{0, \dots, k-1\} \rightarrow \mathbb{R}$, the dilated convolution operation F on element s of the sequence is defined as

$$F(s) = (x *_{d} f)(s) = \sum_{i=0}^{k-1} f(i) \cdot \mathbf{x}_{s-d \cdot i} \quad (1)$$

where d is the dilatation factor, k is the filter size, and $s - d \cdot i$ accounts for the past information direction. Dilation is thus equivalent to introducing a fixed step between every two adjacent filter maps. When $d = 1$, a dilated convolution reduces to a regular convolution. The use of larger dilation enables an output at the top level to represent a wider range of inputs, expanding the receptive field.

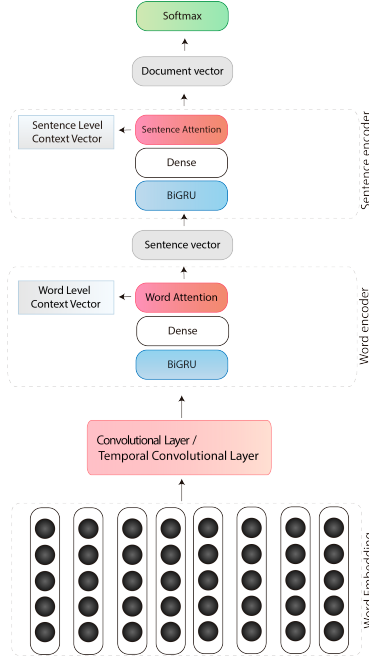


Fig.1: Our HAHNN Architecture include an CNN layer after the embedding layer. In addition, we have created a variant which includes a temporal convolutional layer [2] after the embedding layer.

The proposed model takes into account that the different parts of a document have no similar relevant information. Moreover, determining the relevant sections involves modeling the interactions among the words, not just their isolated presence in the text. Therefore, to consider this aspect, the model includes two levels of attention mechanisms [1]. One structure at the word level and other at the sentence level, which let the model pay more or less attention to individual words and sentences when constructing the document representation.

The strategy consists of different parts: 1) A word sequence encoder and a word-level attention layer; and 2) A sentence encoder and a sentence-level attention layer. In the word encoder, the model uses bidirectional GRU [1] to produce annotations of words by summarizing information from both directions. Therefore, it incorporates the contextual information in the annotation. The attention levels let the model pay more or less attention to individual words and sentences when constructing the representation of the document [6].

Given a sentence with words $w_{it}, t \in [0, T]$ and an embedding matrix W_e , a bidirectional GRU contains the forward $GRU \vec{f}$ which reads the sentence s_i from w_{i1} to w_{iT} and a backward $GRU \overleftarrow{f}$ which reads from w_{iT} to w_{i1} :

$$x_{it} = W_e w_{it}, t \in [1, T], \quad (2) \quad \vec{h}_{it} = \overrightarrow{GRU}(x_{it}), t \in [1, T], \quad (3)$$

$$\overleftarrow{h}_{it} = \overleftarrow{GRU}(x_{it}), t \in [T, 1]. \quad (4)$$

An annotation for a given word w_{it} is obtained by concatenating the forward hidden state and backward hidden state, i.e., $h_{it} = [\vec{h}_{it}, \overleftarrow{h}_{it}]$, which summarizes the information of the whole sentence. We use the attention mechanism to evaluate words that are important to the meaning of the sentence and to aggregate the representation of those informative words into a sentence vector. Specifically,

$$u_{it} = \tanh(W_w h_{it} + b_w) \quad (5) \quad \alpha_{it} = \frac{\exp(u_{it}^\top u_w)}{\sum_t \exp(u_{it}^\top u_w)} \quad (6)$$

$$s_i = \sum \alpha_{it} h_{it} \quad (7)$$

The model measures the importance of a word as the similarity of u_{it} with a word level context vector u_w and learns a normalized importance weight α_{it} through a softmax function. After that, the architecture computes the sentence vector s_i as a weighted sum of the word annotations based on the weights. The word context vector u_w is randomly initialized and jointly learned during the training process.

Given the sentence vectors s_i , and the document vector, the sentence attention is obtained as:

$$\vec{h}_{it} = \overrightarrow{GRU}(s_i), i \in [1, L], \quad (8) \quad \overleftarrow{h}_{it} = \overleftarrow{GRU}(s_i), i \in [L, 1]. \quad (9)$$

The proposed solution concatenates $h_i = [\vec{h}_i, \overleftarrow{h}_i]$ which summarizes the neighbor sentences around sentence i but still focus on sentence i . To reward sentences that are relevant to correctly classify a document, the solution again use attention mechanism and introduce a sentence level context vector u_s using it to measure the importance of the sentences:

$$u_{it} = \tanh(W_s h_i + b_s) \quad (10) \quad \alpha_{it} = \frac{\exp(u_i^\top u_s)}{\sum_i \exp(u_i^\top u_s)} \quad (11)$$

$$v = \sum \alpha_i h_i \quad (12)$$

In the above equation, v is the document vector that summarizes all the information of sentences in a document. Similarly, the sentence level context vector u_s can be randomly initialized and jointly learned during the training process. The output of the sentence attention layer feeds a fully connected softmax layer.

It gives us a probability distribution over the classes. The proposed method is openly available in the github repository ¹.

3 Experiments and Results

We evaluate the proposed model on two document classification datasets using 90% of the data for training and the remaining 10% for tests. We split documents into sentences and tokenize each sentence. The word embeddings have dimension 200 and we use Adam optimizer with a learning rate of 0.001. The datasets used are the IMDb Movie Reviews ² and Yelp 2018 ³. The former contains a set of 25k highly polar movie reviews for training and 25k for testing, whereas the classification involves detecting positive/negative reviews. The latter include users ratings and write reviews about stores and services on Yelp, being a dataset for multiclass classification (ratings from 0-5 stars). Yelp 2018 contains around 5M full review text data, but we fix in 500k the number of used samples for computational purposes.

Table 1: Results in classification accuracies.

Method	Accuracy on test set	
	Yelp 2018 (five classes)	IMDb (two classes)
VDNN [7]	62.14	79.47
HN-ATT [6]	72.73	89.02
CNN [5]	71.81	91.34
Our model with CNN	73.28	92.26
Our model with TCN	72.63	95.17

Table 1 shows the experiment results comparing our results with related works. Note that HN-ATT [6] obtained an accuracy of 72,73% in the Yelp test set, whereas the proposed model obtained an accuracy of 73,28%. Our results also outperformed CNN [6] and VDNN [7]. We can see an improvement of the results in Yelp with our approach using CNN and varying window sizes in filters. The model also performs better in the results with IMDb using both CNN and TCN.

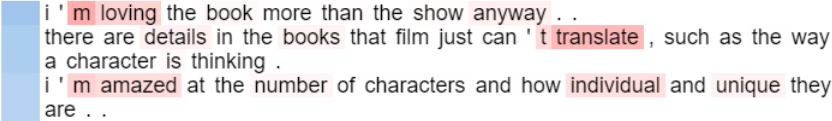
3.1 Attention Weights Visualizations

To validate the model performance in select informative words and sentences, we present the visualizations of attention weights in Figure 2. There is an example of the attention visualizations for a positive and negative class in test reviews. Every line is a sentence. Blue color denotes the sentence weight, and red denotes the word weight in determining the sentence meaning. There is a greater focus on more important features despite some exceptions. For example, the word “loving” and “amazed” in Figure 2 (a) and “disappointment” in Figure 2 (b).

¹ <https://github.com/luisfredgs/cnn-hierarchical-network-for-document-classification>

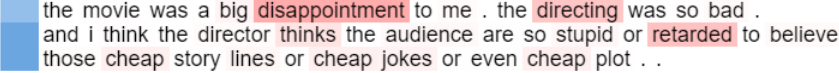
² <http://ai.stanford.edu/amaas/data/sentiment/>

³ <https://www.yelp.com/dataset/challenge>



Predicted rating: Positive

(a) A positive example of visualization of a strong word in the sentence.



Predicted rating: Negative

(b) A negative example of visualization of a strong word in the sentence.

Fig. 2: Visualization of attention weights computed by the proposed model

Occasionally, we have found issues in some sentences, where fewer important words are getting higher importance. For example, in Figure 2 (a) notes that the word “translate” has received high importance even though it represents a neutral word. These drawbacks will be taken into account in future works.

4 Final Remarks

In this paper, we have presented the HAHNN architecture for document classification. The method combines CNN with attention mechanisms in both word and sentence level. HAHNN improves accuracy in document classification by incorporate the document structure in the model and employing CNN’s for the extraction of more abundant features.

References

1. BAHDANAU, Dzmitry; CHO, Kyunghyun; BENGIO, Yoshua. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473, 2014.
2. BAI, Shaojie; KOLTER, J. Zico; KOLTUN, Vladlen. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv preprint arXiv:1803.01271, 2018.
3. BOJANOWSKI, Piotr et al. Enriching word vectors with subword information. arXiv preprint arXiv:1607.04606, 2016.
4. CHO, Kyunghyun et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078, 2014.
5. KIM, Yoon. Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882, 2014.
6. YANG, Zichao et al. Hierarchical attention networks for document classification. In: Conf. North Am. Chapter of the Assoc. for Comp. Ling. 2016. p.1480-1489, San Diego, CA, USA.
7. Conneau, Alexis, et al. "Very deep convolutional networks for text classification." arXiv preprint arXiv:1606.01781 (2016).