Structural Attention Neural Networks for improved sentiment analysis

Filippos Kokkinos[1] **and Alexandros Potamianos**[1]

[1]School of E.C.E. , National Technical University of Athens , 15773 Athens, Greece
{el11142,potam}@central.ntua.gr

Abstract

We introduce a tree-structured attention neural network for sentences and small phrases and apply it to the problem of sentiment classification. Our model expands the current recursive models by incorporating structural information around a node of a syntactic tree using both bottom-up and top-down information propagation. Also, the model utilizes structural attention to identify the most salient representations during the construction of the syntactic tree. To our knowledge, the proposed models achieve state of the art performance on the Stanford Sentiment Treebank dataset.

1 Introduction

Sentiment analysis deals with the assessment of opinions, speculations, and emotions in text (Zhang et al., 2012; Pang and Lee, 2008). It is a relatively recent research area that has attracted great interest as demonstrated by a series of shared evaluation tasks, e.g., analysis of tweets (Nakov et al., 2016). In (Turney and Littman, 2002), the affective ratings of unknown words were predicted utilizing the affective ratings of a small set of words (seeds) and the semantic relatedness between the unknown and the seed words. An example of sentence-level analysis was proposed in (Malandrakis et al., 2013). Other application areas include the detection of public opinion and prediction of election results (Singhal et al., 2015), correlation of mood states and stock market indices (Bollen et al., 2011).

Recently, Recurrent Neural Network (RNN) with Long-Short Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) or Gated Recurrent Units (GRU) (Chung et al., 2014) have been applied to various Natural Language Processing tasks. Tree structured neural networks, which are found in literature as Recursive Neural Networks, hold a linguistic interest due to their close relation to syntactic structures of sentences being able to capture distributed information of structure such as logical terms(Socher et al., 2012). These syntactic structures are N-ary trees which represent either the underlying structure of a sentence, known as constituency trees or the relations between words known as dependency trees.

This paper focuses on sentence-level sentiment classification of movie reviews using syntactic parse trees as input for the proposed networks. In order to solve the task of sentiment analysis of sentences, we work upon a variant of Recursive Neural Networks which recursively create representation following the syntactic structure. The proposed computation model exploits information from subnodes as well as parent nodes of the node under examination. This neural network is referred to as Bidirectional Recursive Network (Irsoy and Cardie, 2013). The model is further enhanced with memory units and the proposed structural attention mechanism. It is observed that different nodes of a tree structure hold information of variable saliency. Not all nodes of a tree are equally informative, so the proposed model selectively weights the contribution of each node regarding the sentence level representation using structural attention model.

We evaluate our approach on the sentence-level sentiment classification task using one standard movie review dataset (Socher et al., 2013). Experimental results show that the proposed model outperforms the state-of-the art methods.

Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, pages 586–591,
Valencia, Spain, April 3-7, 2017. ©2017 Association for Computational Linguistics

2 Tree-Structured GRUs

Recursive GRUs (TreeGRU) upon tree structures are an extension of the sequential GRUs that allow information to propagate through network topologies. Similar to Recursive LSTM network on tree structures (Tai et al., 2015), for every node of a tree, the TreeGRU has gating mechanisms that modulate the flow of information inside the unit without the need of a separate memory cell. The activation h_j of TreeGRU for node j is the interpolation of the previous calculated activation h_{jk} of its kth child out of N total children and the candidate activation $\widetilde{h}_j$.

$$h_j = z_j * \sum_{k=1}^{N} h_{jk} + (1 - z_j) * \widetilde{h}_j \tag{1}$$

where z_j is the update function which decide the degree of update that will occur on the activation based on the input vector x_j and previously calculated representation h_{jk} :

$$z_j = \sigma(U_z * x_j + \sum_{k=1}^{N} W_z^i * h_{jk}) \tag{2}$$

The candidate activation $\widetilde{h}_j$ for a node j is computed similarly to that of a Recursive Neural Network as in (Socher et al., 2011):

$$\widetilde{h}_j = f(U_h * x_j + \sum_{k=1}^{N} W_h^k * (h_{jk} * r_j)) \tag{3}$$

where r_j is the reset gate which allows the network to forget effectively previous computed representations when the value is close to 0 and it is computed as follows:

$$r_j = \sigma(U_r * x_j + \sum_{k=1}^{N} W_r^k * h_{jk}) \tag{4}$$

Every part of a gated recurrent unit $x_j, h_j, r_j, z_j, \widetilde{h}_j \in \mathbb{R}^d$ where d is the input vector dimensionality. σ is the sigmoid function and f is the non-linear tanh function.The set of matrices $W^k, U \in \mathbb{R}^{dxd}$ used in 2 - 4 are the trainable weight parameters which connect the kth children node representation with the jth node representation and the input vector x_j.

2.1 Bidirectional TreeGRU

A natural extension of Tree-Structure GRU is the addition of a bidirectional approach. TreeGRUs calculate an activation for node j with the use of previously computed activations lying lower in the tree structure. The bidirectional approach for a tree structure uses information both from under and lower nodes of the tree for a particular node j. In this manner, a newly calculated activation incorporates content from both the children and the parent of a particular node.

The bidirectional neural network can be trained in two seperate phases: i) the Upward phase and ii) the Downward phase. During the Upward phase, the network topology is similar to the topology of a TreeGRU, every activation is calculated based on the previously calculated activations which are found lower on the structure in a bottom up fashion. When every activation has been computed, from leaves to root, then the root activation is used as input of the Downward phase. The Downward phase calculates the activations for every child of a node using content from the parent in a top down fashion. The process of computing the internal representations between the two phases is separated, so in a first pass the network compute the upward activation and after this is completed, then the downward representations are computed.

The upward activation $h^{\uparrow}_j$ similarly to TreeGRU for node j is the interpolation of the previous calculated activation $h^{\uparrow}_{jk}$ of its kth child out of N total children and the candidate activation $\widetilde{h}^{\uparrow}_j$.

$$h^{\uparrow}_j = z^{\uparrow}_j * \sum_{k=1}^{N} h^{\uparrow}_{jk} + (1 - z^{\uparrow}_j) * \widetilde{h}^{\uparrow}_j \tag{5}$$

The update gate, rest gate and candidate activation are computed as follows:

$$z^{\uparrow}_j = \sigma(U_z * x^{\uparrow}_j + \sum_{k=1}^{N} W_z^k * h^{\uparrow}_{jk}) \tag{6}$$

$$r^{\uparrow}_j = \sigma(U_r * x^{\uparrow}_j + \sum_{k=1}^{N} W_r^k * h^{\uparrow}_{jk}) \tag{7}$$

$$\widetilde{h}^{\uparrow}_j = f(U_h * x^{\uparrow}_j + \sum_{k=1}^{N} W_r^k * (h^{\uparrow}_{jk} * r^{\uparrow}_j)) \tag{8}$$

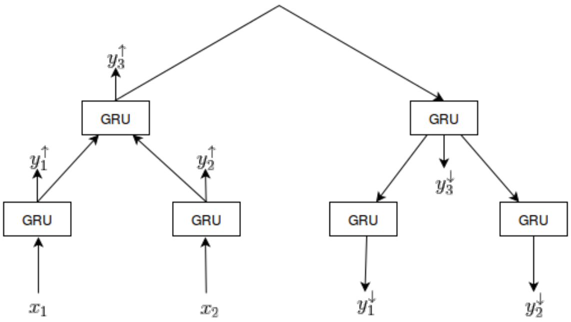

Figure 1: A tree-structured bidirectional neural network with Gated Recurrent Units. The input vectors x are given to the model in order to generate the phrase representations $y^\uparrow$ and $y^\downarrow$.

The downward activation $h_j^\uparrow$ for node j is the interpolation of the previous calculated activation $h_{p(j)}^\uparrow$, where the function p calculates the index of the parent node, and the candidate activation $\widetilde{h}_j^\downarrow$.

$$h_j^\downarrow = z_j^\downarrow * h_{p(j)}^\downarrow + (1 - z_j^\downarrow) * \widetilde{h}_j^\downarrow \tag{9}$$

The update gate, reset gate and candidate activation for the downward phase are computed as follows:

$$z_j^\downarrow = \sigma(U_z^d * h_j^\uparrow + W_z^d * h_{p(j)}^\downarrow) \tag{10}$$

$$r_j^\downarrow = \sigma(U_r^d * h_j^\uparrow + W_r^d * h_{p(j)}^\downarrow) \tag{11}$$

$$\widetilde{h}_j^\downarrow = f(U_h^d * h_j^\uparrow + W_h^d * (h_{p(j)}^\downarrow * r_j^\downarrow)) \tag{12}$$

During downward phase, matrix $U^d \in \mathbb{R}^{dxd}$ connects the upward representation of node j with the respective jth downward node while $W^d \in \mathbb{R}^{dxd}$ connect the parent representation $p(j)$.

2.2 Structural Attention

We introduce Structural Attention, a generalization of sequential attention model (Luong et al., 2015) which extracts informative nodes out of a syntactic tree and aggregates the representation of those nodes in order to form the sentence vector. We feed representation h_j of node through a one-layer Multilayer Perceptron with $W_w \in \mathbb{R}^{dxd}$ weight matrix to get the hidden representation u_j.

$$u_j = tanh(W_w * h_j) \tag{13}$$

Using the softmax function, the weights a_j for each node are obtained based on the similarity of the hidden representation u_j and a global context vetor $u_w \in \mathbb{R}^d$. The normalized weights a_j are used to form the final sentence representation $s \in \mathbb{R}^d$ which is a weighted summation of every node representation h_j.

$$a_j = \frac{u_j^\top * u_w}{\sum_{i=1}^{N} u_i^\top * u_w} \tag{14}$$

$$s = \sum_{i=1}^{N} a_i h_i \tag{15}$$

The proposed attention model is applied on structural content since all node representations contain syntactic structural information during training because of the recursive nature of the network topology.

3 Experiments

We evaluate the performance of the aforementioned models on the task of sentiment classification of sentences sampled from movie reviews. We use the Stanford Sentiment Treebank (Socher et al., 2013) dataset which contains sentiment labels for every syntactically plausible phrase out of the 8544/1101/2210 train/dev/test sentences. Each phrase is labeled with respect to a 5-class sentiment value, i.e. very negative, negative, neutral, positive, very positive. The dataset can also be used for a binary classification subtask by excluding any neutral phrases for the original splits. The binary classification subtask is evaluated on 6920/872/1821 train/dev/test splits.

3.1 Sentiment Classification

For all of the aforementioned architectures at each node j we use a softmax classifier to predict the sentiment label $\hat{y}_j$. For example, the predicted label $\hat{y}_j$ corresponds to the sentiment class of the spanned phrase produced from node j. The classifier for unidirectional TreeGRU architectures uses the hidden state h_j produced from recursive computations till node j using a set x_j of input nodes to predict the label as follows:

$$\hat{p}_\theta(y|x_j) = softmax(W_s * h_j) \tag{16}$$

where $W_s \in \mathbb{R}^{dxc}$ and c is the number of sentiment classes.

The classifier for bidirectional TreeBiGRU architectures uses both the hidden state $h_j^\uparrow$ and $h_j^\downarrow$ produced from recursive computations till node j during Upward and Downward Phase using a set x_j of input nodes to predict the label as follows:

$$\hat{p}_\theta(y|x_j) = softmax(W_s^\uparrow * h_j^\uparrow + W_s^\downarrow * h_j^\downarrow) \tag{17}$$

where $W_s^{\uparrow}, W_s^{\downarrow} \in \mathbb{R}^{dxc}$ and c is the number of sentiment classes. The predicted label $\hat{y}_j$ is the argument with the maximum confidence:

$$\hat{y}_j = \underset{y}{\operatorname{argmax}}(\hat{p}_\theta(y|x_j)) \tag{18}$$

For the Structural Attention models, we use for the final sentence representation s to predict the sentiment label $\hat{y}_j$ where j is the corresponding root node of a sentence. The cost function used is the negative log-likelihood of the ground-truth label y^k at each node:

$$E(\theta) = \sum_{k=1}^{m} \hat{p}_\theta(y^k|x^k) + \frac{\lambda}{2}||\theta||^2 \tag{19}$$

where m is the number of labels in a training sample and λ is the L2 regularization hyperparameter.

Network Variant	d	$\|\theta\|$
TreeGRU		
-without attention	300	7323005
-with attention	300	7413605
TreeBiGRU		
-without attention	300	8135405
-with attention	300	8317810

Table 1: Memory dimensions d and total network parameters $|\theta|$ for every network variant evaluated

3.2 Results

The evaluation results are presented in Table 2 in terms of accuracy, for several state-of-the-art models proposed in the literature as well as for the TreeGRU and TreeBiGRU models proposed in this work. Among the approaches reported in the literature, the highest accuracy is yielded by DRNN and DMN for the binary scheme (88.6), and by DMN for the fine-grained scheme (52.1). We observe that the best performance is achieved by TreeBiGRU with attention, for both binary (89.5) and fine-grained (52.4) evaluation metrics, exceeding any previously reported results. In addition, the attentional mechanism employed in the proposed TreeGRU and TreeBiGRU models improve the performance for both evaluation metrics.

4 Hyperparameters and Training Details

The evaluated models are trained using the AdaGrad (Duchi et al., 2010) algorithm using 0.01 learning rate and a minibatch of size 25 sentences. L2-regularization is performed on the model parameters with a λ value 10^{-4}. We use dropout with probability 0.5 on both the input layer and the softmax layer.

System	Binary	Fine-grained
RNN	82.4	43.2
MV-RNN	82.9	44.4
RNTN	85.4	45.7
PVec	87.8	48.7
TreeLSTM	88.0	51.0
DRNN	86.6	49.8
DCNN	86.8	48.5
CNN-multichannel	88.1	47.4
DMN	88.6	52.1
TreeGRU		
- without attention	88.6	50.5
- with attention	89.0	51.0
TreeBiGRU		
- without attention	88.5	51.3
- with attention	**89.5**	**52.4**

Table 2: Test Accuracies achieved on the Stanford Sentiment Treebank dataset. RNN, MV-RNN and RNTN (Socher et al., 2013). PVec: (Mikolov et al., 2013). TreeLSTM (Tai et al., 2015). DRNN (Irsoy and Cardie, 2013). DCNN (Kalchbrenner et al., 2014).CNN-multichannel (Kim, 2014). DMN (Kumar et al., 2015)

The word embeddings are initialized using the public available Glove vectors with a 300 dimensionality. The Glove vectors provide 95.5% coverage for the SST dataset. All initialized word vectors are finetuned during the training process along with every other parameter. Every matrix is initialized with the identity matrix multiplied by 0.5 except for the matrices of the softmax layer and the attention layer which are randomly initialized from the normal Gaussian distribution. Every bias vectors is initialized with zeros.

The training process lasts for 40 epochs. During training, we evaluate the network 4 times every epoch and keep the parameters which give the best root accuracy on the development dataset.

5 Conclusion

In this short paper, we propose an extension of Recursive Neural Networks that incorporates a bidirectional approach with gated memory units as well as an attention model on structure level. The proposed models were evaluated on both fine-grained and binary sentiment classification tasks on a sentence level. Our results indicate that both the direction of the computation and the attention on a structural level can enhance the performance of neural networks on a sentiment analysis task.

6 Acknowledgments

This work has been partially funded by the BabyRobot project supported by the EU Horizon 2020 Programme, grant number 687831. Also, the authors would like to thank NVIDIA for supporting this work by donating a TitanX GPU.

References

Johan Bollen, Huina Mao, and Xiaojun Zeng. 2011. Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1):1–8.

Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555.

John Duchi, Elad Hazan, and Yoram Singer. 2010. Adaptive subgradient methods for online learning and stochastic optimization. Technical Report UCB/EECS-2010-24, EECS Department, University of California, Berkeley.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Ozan Irsoy and Claire Cardie. 2013. Bidirectional recursive neural networks for token-level labeling with structure. *CoRR*, abs/1312.0493.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 655–665, Baltimore, Maryland, June. Association for Computational Linguistics.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar, October. Association for Computational Linguistics.

Ankit Kumar, Ozan Irsoy, Jonathan Su, James Bradbury, Robert English, Brian Pierce, Peter Ondruska, Ishaan Gulrajani, and Richard Socher. 2015. Ask me anything: Dynamic memory networks for natural language processing. *CoRR*, abs/1506.07285.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal, September. Association for Computational Linguistics.

Nikolaos Malandrakis, Abe Kazemzadeh, Alexandros Potamianos, and Shrikanth Narayanan. 2013. Sail: A hybrid approach to sentiment analysis. *In Proceedings SemEval*, pages 438–442.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Preslav Nakov, Alan Ritter, Sara Rosenthal, Fabrizio Sebastiani, and Veselin Stoyanov. 2016. Semeval-2016 task 4: Sentiment analysis in twitter. In *Proceedings of the 10th international workshop on semantic evaluation (SemEval 2016)*.

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, 2(1-2):1–135.

Kartik Singhal, Basant Agrawal, and Namita Mittal. 2015. Modeling indian general elections: sentiment analysis of political twitter data. In *Information Systems Design and Intelligent Applications*, pages 469–477.

Richard Socher, Jeffrey Pennington, Eric H. Huang, Andrew Y. Ng, and Christopher D. Manning. 2011. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 151–161, Stroudsburg, PA, USA. Association for Computational Linguistics.

Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 1201–1211. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Y. Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1556–1566, Beijing, China, July. Association for Computational Linguistics.

Peter Turney and Michael L. Littman. 2002. Unsupervised learning of semantic orientation from a hundred-billion-word corpus.

Zhu Zhang, Xin Li, and Yubo Chen. 2012. Deciphering word-of-mouth in social media: Text-based metrics of consumer reviews. *ACM Trans. Manage. Inf. Syst.*, 3(1):5:1–5:23.