An Attention-based Recursive Neural Network for Semantic Representation

Yao Zhou, Cong Liu, Yan Pan
School of Data and Computer Science, Sun Yat-sen University
{zhouyao@mail2, liucong3@mail, panyan5@mail}.sysu.edu.cn

Abstract

Attention mechanisms have lately succeeded in many tasks, including neural machine translation (NMT), natural language inference (NLI), machine comprehension and question answering (QA). While existing attention-based models exert attention on a single sequential memory, we attempt to focus attention recursively on local structures. Specially, we use attention to select semantically more important modifiers, when constructing a sentence representation on a dependency tree using LSTM and GRU, respectively. Our attention-based recursive neural network outperform the non-attentional counterparts in two semantic evaluation tasks: semantic relatedness (SemEval 2014, Task 1) and sentiment classification (Stanford Sentiment Treebank).

1 Introduction

Recursive neural networks (Goller and Kuchler, 1996; Socher et al., 2011) are tree-structured models, which compose each phrase and sentence representation from its constituent subphrases according to a given syntactic structure over the sentence. In a dependency tree, each node is a word and each word can be presented as a word vector, known as distributed representation (Bengio et al., 2003). Compared with sequential models, such as *recurrent neural networks* (RNNs) (Elman, 1990; Mikolov, 2012), tree-structured models (Socher et al., 2013) can capture more semantics of natural language for the contribution of the syntactic structure.

Attentive neural networks have recently demonstrated success in a wide variety of tasks arranging from neural machine translation (Bahdanau et al., 2015; Luong et al., 2015), sentence summarization (Rush et al., 2015), natural language reason (Rocktaschel et al., 2015) to question answering (Weston et al., 2015; Sukhbaatar et al., 2015; Kumar et al., 2015). The key idea is to allow the model to attend over past output vectors. For example, in machine translation, the attention mechanism allows models to learn alignments between the words across languages.

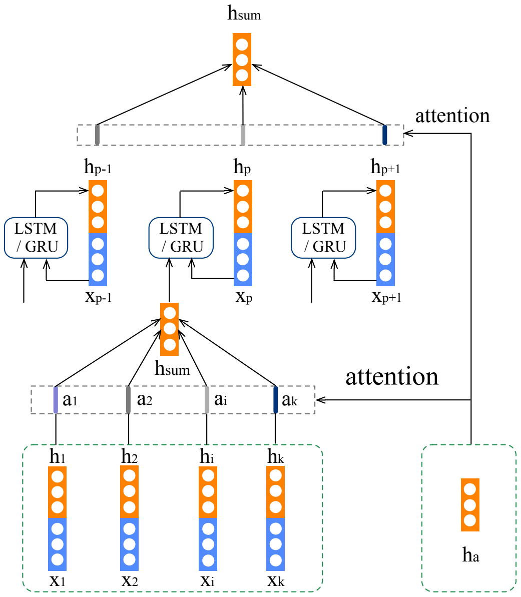

Figure 1: Our attention-based recursive neural network. For each parent, we compose an input by summing all children's hidden states weighted by an attention vector.

Tree-LSTMs compose parent activation by summing up all children's activations (Tai et al., 2015). However, it is ignored that every child in a dependency tree has different contribution. In this paper, we propose an attention-based model to direct the composition of the parent's representation towards the children are more important. Because of the selective composition, noisy information will be declined and useful information will be enhanced. Since the proposed algorithm

	Sequential	Tree-structured
LSTM	(Sutskever et al., 2014)	(Tai et al., 2015)
GRU	(Chung et al., 2014)	√
LSTM + attention	(Luong et al., 2015)	√
GRU + attention	√	√

Table 1: √ denotes our work in this paper.

apply attention on recursive neural networks, we call it *attention-based recursive neural network* (ARNN).

We conduct two experiments for semantic representation: sentiment classification of movie reviews and semantic relatedness of sentence pairs. Sentiment classification has been a benchmark for sentence representation because a good representation could imply sentiment polarity. In both tasks, our models outperform the non-attentional countparts. In semantic relatedness task, our models achieve the state-of-the-art performance.

2 Preliminaries

In this section, we will introduce Long Short-Term Memory units (Hochreiter and Schmidhuber, 1997) and Gated Recurrent Units (Chung et al., 2014). After that, we will look back to the Tree-structured LSTM (Tai et al., 2015).

2.1 LSTM and GRU

When training a recurrent neural network, components of the gradient vector can grow or decay exponentially over long sequences (Hochreiter, 1998; Bengio et al., 1994). Long Short-term Memory (LSTM) units and Gated Recurrent Units (GRU) are two architectures to address this problem of learning long-term dependencies by introducing gates that allow the model to suffer less from the *vanishing gradient problem* over long periods of time. Comparison to LSTM, GRU has less gates and is less computationally expensive.

LSTM Definition: an LSTM unit at each time step t include three gates: an input gate i_t, an forget gate f_t and an output gate o_t. The entries of the gating vectors i_t, f_t and o_t are in [0, 1]. Additionally, a memory cell c_t is for recording the previous information. The transition equations are the following:

$$\begin{aligned}
i_t &= \sigma(W^{(i)}x_t + U^{(i)}h_{t-1} + b^{(i)}), \\
f_t &= \sigma(W^{(f)}x_t + U^{(f)}h_{t-1} + b^{(f)}), \\
o_t &= \sigma(W^{(o)}x_t + U^{(o)}h_{t-1} + b^{(o)}), \\
u_t &= \tanh(W^{(u)}x_t + U^{(u)}h_{t-1} + b^{(u)}), \\
c_t &= i_t \odot u_t + f_t \odot c_{t-1}, \\
h_t &= o_t \odot \tanh(c_t).
\end{aligned} \tag{1}$$

GRU Definition: a GRU unit just has two gates and a candidate hidden state. An update gate z_t, a reset gate r_t and a candidate hidden state $\tilde{h}_t$.

$$\begin{aligned}
z_t &= \sigma(W^{(z)}x_t + U^{(z)}h_{t-1} + b^{(z)}), \\
r_t &= \sigma(W^{(r)}x_t + U^{(r)}h_{t-1} + b^{(r)}), \\
\tilde{h}_t &= \tanh(Wx_t + r_t \odot Uh_{t-1} + b^{(h)}), \\
h_t &= z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t
\end{aligned} \tag{2}$$

Where x_t is the input word vector at the current time step, σ denotes the logistic sigmoid function and $\odot$ denotes element wise multiplication.

2.2 Dependency Tree-LSTM

LSTM has been used in both recurrent and recursive neural networks, such as Tree-LSTMs (Tai et al., 2015). It has been identified that tree-structured LSTMs can improve semantic representation. There are two architectures of tree-structured LSTMs, *Child-Sum Tree-LSTMs* and *N-ary Tree-LSTMs*. Because a dependency tree may have more than two children at a certain layer and whose children are unordered, the *Child-Sum Tree-LSTMs* (Tai et al., 2015) is suitable for composing the hidden states of children.

Given a dependency tree, let $C(j)$ denote the set of children of node j. The Child-Sum Tree-LSTMs composes the forget gate of parent node j and child node k by the equation

$$f_{jk} = \sigma(W^{(f)}x_j + U^{(f)}h_k + b^{(f)}), \tag{3}$$

The candidate hidden state $\tilde{h}_j$ is the sum of child hidden states: $\tilde{h}_j = \sum_{k \in C(j)} h_k$. Then memory cell c_j computed by:

$$c_j = i_j \odot u_j + \sum_{k \in C(j)} f_{jk} \odot c_k, \tag{4}$$

3 Attention-based Recursive Neural Network

After introducing dependency tree-LSTMS, we now present the details of Attention-based Recursive Neural Network (ARNN) for semantic representation. An illustration of ARNN is given in Figure 1.

3.1 Modeling Semantic Representation

How to choose attention vector is an essential problem. For question answering, a question or a query will be the attention vector (Weston et al., 2015). For machine translation, current target state will be the attention vector (Luong et al., 2015). In this paper, we propose two attention vectors for semantic representation:

- **Static attention vector**, we feed a fixed value vector as our attention vector. In this strategy, It's learning to attend when we train our neural network.
- **Dynamic attention vector**, we using a sequential RNNs with LSTM or GRU to represent a sentence. Then feed this representation as attention vector. This sequential network and our tree-structured network are trained at the same time.

Suppose we have a parent node p , it has k child nodes and each node has a word representation x_i and a hidden state $h_i^{(c)}$. Then all child hidden states can be presented as a $k \times d$ matrix $H^{(c)} = [h_1^{(c)}, h_2^{(c)} ..., h_k^{(c)}]$, where d is the hidden dimension. Furthermore, let $e \in \mathbb{R}^k$ be a vector of 1s and h_a be the attention vector. The attention mechanism will produce a vector a of attention weights and a weighted representation $h_{sum}^{(c)}$ by the following equations:

$$M = \tanh(W^{(c)} H^{(c)} + W^{(a)} h^{(a)} \otimes e), \quad (5)$$

$$a = softmax(w^\intercal M), \quad (6)$$

$$h_{sum}^{(c)} = H^{(c)} a^\intercal, \quad (7)$$

We take $h_{sum}^{(c)}$ and the word vector x_p of parent node p as the inputs of LSTM or GRU at this layer. That is, the hidden state of parent node p can be computed by $h_p = \textbf{LSTM/GRU}(x_p, h_{sum}^{(c)})$. It can be recursively composed in a dependency tree, then the hidden state at each layer is a phrase representation and the root hidden state become the sentence representation.

So far, we have introduced our attention-based recursive memory network. Comparison to Dependency Tree-LSTM (Tai et al., 2015), we apply attention mechanism to compose phrase and sentence representation by selectively focusing on parts of the child hidden states.

3.2 Training

We used 300-dimensional *Glove vectors* (Pennington et al., 2014) to initialize the word embeddings of the ARNN due to Glove vectors can capture context-independent representations. Out-of-vocabulary word are randomly initialized for a uniform distribution and the initial hidden states are $zeros$.

We used AdaGrad (Duchi et al., 2011) for optimization with a learning rate of 0.05 and a minibatch size of 25. We also apply L2 regularization to regularize our model parameters with strength of 10^{-4}. Additionally, sentiment classifier was regularized using dropout (Hinton et al., 2012) with a dropout rate of 0.5. Subsequently, we take the best configuration based on performance on the development set, and evaluate only that configuration on the test set.

4 Experiments

4.1 Semantic relatedness

First, we conduct our experiment on SemEval 2014 Task 1: semantic relatedness SICK dataset (Marelli et al., 2014). Given a pair of sentences derived from a image and or a video, our goal is to produce a score of semantic relatedness. The score value between 1 and 5, higher scores indicates that this two sentences are semantically related. This dataset consists of 9927 sentence pairs with the split of 4500 training pairs, 500 development pairs and 4927 testing pairs.

In this experiment, given a pair of sentences, we produce two sentence representations by GRUs and ARNN introduced at section [2] and section[3]. Then we predict the similarity score $\hat{y}$ using a neural network that takes the pair representation (h_L, h_R):

$$\begin{aligned} h_\times &= h_L \odot h_R, \\ h_+ &= |h_L - h_R|, \\ h_s &= \sigma(W^{(x)} h_\times + W^{(+)} h_+ + b^{(h)}), \\ \hat{p}_\theta &= softmax(W^{(p)} h_s + b^{(p)}), \\ \hat{y} &= r^\intercal \hat{p}_\theta \end{aligned} \quad (8)$$

let r^{T} = [1, 2, ..., 5] be an integer vector. We compute target distribution p as a function of prediction scores y given by $p_i = y - \lfloor y \rfloor$ if $i = \lfloor y \rfloor + 1$, $p_i = \lfloor y \rfloor - y + 1$ if $i = \lfloor y \rfloor$ and 0 otherwise. The cost function is the regularized KL-divergence between p and $\hat{p}_\theta$:

$$J(\theta) = \frac{1}{m} \sum_{k=1}^{m} KL(p^{(k)} \| \hat{p}_\theta^{(k)}) + \frac{\lambda}{2} \|\theta\|_2^2, \quad (9)$$

where m denotes the number of training pairs and k indicates the the kth sentence pair.

4.2 Sentiment classification

Stanford Sentiment Treebank (SST) is a standard dataset for sentiment classification. Each sentence from a movie review is parsed as a constituency tree, and each node in a tree is annotated with a sentiment polarity label. There are two classification tasks, binary classification: positive, negative and fine-grained classification: very negative, negative, neutral, positive, and very positive. We use predefined train/dev/test splits of 6920/872/1821 for the binary classification subtask and 8544/1101/2210 for fine-grained classification subtask.

In the classification task, the criterion is the ClassNLLCriterion which defined as:

$$J(\theta) = -\frac{1}{m} \sum_{k=1}^{m} \log \hat{p}_\theta(y^{(k)} | \{x\}^{(k)}) + \frac{\lambda}{2} \|\theta\|_2^2, \quad (10)$$

We apply recurrent and recursive neural network with LSTMs or GRUs for sentence representation with attention vector.

5 Results

5.1 Semantic relatedness

Our results are summarized in Table 2. First, all our models are able to outperform all submissions from SemEval 2014 competition. Our attention-based Tree-LSTM perform better than dependency Tree-LSTM (Tai et al., 2015) when we apply attention mechanism on which. Surprisingly, comparison to all existing models, we find our static attention-based Tree-GRU achieve the state-of-the-art score in three metrics. All results shows that our attention-based models can learn representations that are well suited for semantic relatedness.

Method	r	ρ	MSE
Illinois-LH (Lai and Hockenmaier, 2014)	0.7993	0.7538	0.3692
UNAL-NLP (Jimenez et al., 2014)	0.8070	0.7489	0.3550
Meaning Factory (Bjerva et al., 2014)	0.8268	0.7721	0.3324
ECNU (Zhao et al., 2014)	0.8414	-	-
LSTM (Tai et al., 2015)	0.8528	0.7911	0.2831
Bidirectional LSTM (Tai et al., 2015)	0.8567	0.7966	0.2736
Dependency Tree-LSTM (Tai et al., 2015)	0.8676	0.8083	0.2532
combine-skip+COCO (Kiros et al., 2015)	0.8655	0.7995	0.2561
GRU	0.8595	0.7974	0.2689
Bidirectional GRU	0.8662	0.8025	0.2603
Dependency Tree-GRU	0.8672	0.8116	0.2573
Static-attention Dependency Tree-GRU	**0.8730**	**0.8117**	**0.2426**
Static-attention Dependency Tree-LSTM	0.8692	0.8106	0.2528
Dynamic-attention Dependency Tree-GRU	0.8684	0.8106	0.2548
Dynamic-attention Dependency Tree-LSTM	0.8701	0.8085	0.2524

Table 2: The evaluation metrics of SICK semantic relatedness subtask are Pearson's r, Spearman's ρ and mean squared error. First group is SemEval 2014 baselines, and last group is our experiment results.

Method	Fine-grained(%)	Binary(%)
DCNN (Blunsom et al., 2014)	48.5	86.8
Paragraph-Vec (Le and Mikolov, 2014)	48.7	87.8
CNN-non-static (Kim, 2014)	48.0	87.2
CNN-multichannel (Kim, 2014)	47.4	88.1
DRNN (Irsoy and Cardie, 2014)	49.8	86.6
LSTM (Tai et al., 2015)	46.4	84.9
Bidirectional LSTM (Tai et al., 2015)	49.1	87.5
Dependency Tree-LSTM (Tai et al., 2015)	48.4	85.7
Constituency Tree-LSTM (Tai et al., 2015)	**51.0**	88.0
GRU	46.3	86.2
Bidirectional GRU	46.8	85.6
Dependency Tree-GRU	47.8	85.1
Static-attention Dependency Tree-GRU	48.1	85.5
Static-attention Dependency Tree-LSTM	50.3	86.2
Dynamic-attention Dependency Tree-GRU	49.1	87.5
Dynamic-attention Dependency Tree-LSTM	49.4	**88.3**

Table 3: Test set accuracies on the Stanford Sentiment Treebank. **Fine-grained**: 5-class classification task. **Binary**: positive/negative, 2-class classification task.

5.2 Sentiment classification

We present results on sentiment classification task in Table 3. Our attention-based dependency Tree-LSTM perform state-of-the-art performance in binary classification. Compared with the non-attention models (e.g., Dependency Tree-LSTM), our attention-based models achieve better accuracy. In fine-gained classification task, our Static-attention model precede Dependency Tree-LSTM nearly two percent. We also find that LSTM outperform GRU no matter structure is recurrent or recursive. We guess LSTM is more suitable for the classification tasks.

6 Conclusion

In this paper, we introduced an attention-based recursive neural network for semantic representation. Because of the success of attention mechanism and Tree-LSTM, we combine these two approaches for modeling semantic representation. We demonstrated that our attention-based recursive neural networks are effective in two tasks: semantic relatedness and sentiment classification. Based on the above, one can draw a conclusion that attention mechanism and tree structure are suitable for semantic representation.

References

[Socher et al.2011] Richard Socher, Cliff C. Lin, Andrew Y. Ng and Christopher D. Manning. 2011. Parsing Natural Scenes and Natural Language with Recursive Neural Networks. In *Proceedings of the 26th International Conference on Machine Learning (ICML).*

[Goller and Kuchler1996] Christoph Goller and Andreas Kuchler. 1996. Learning task-dependent distributed representations by backpropagation through structure. *IEEE International Conference on Neural Networks*. volume 1, pages 347–352.

[Bengio et al.2003] Yoshua Bengio, R. Ducharme, Pascal Vincent and Christian Jauvin. 2003. A Neural Probabilistic Language Model. *Journal of Machine Learning Research*, 3:1137–1155.

[Elman1990] Jeffrey L. Elman. 1990. Finding structure in time. *Cognitive science*. 14(2):179–211.

[Mikolov2012] Tomas Mikolov. 2012. Statistical Language Models Based on Neural Networks. *Ph.D. thesis, Brno University of Technology.*

[Socher et al.2013] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, Andrew Ng and C. 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)*

[Bahdanau et al.2015] D. Bahdanau, K. Cho and Y. Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations (ICLR).*

[Luong et al.2015] Minh-Thang Luong, Hieu Pham and Christopher D. Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *Annual Meeting of the Association for Computational Linguistics (ACL).*

[Rush et al.2015] Alexander M. Rush, Sumit Chopra and Jason Weston. 2015. A Neural Attention Model for Abstractive Sentence Summarization. In *Proceedings of the 2015 conference on empirical methods in natural language processing (EMNLP 2015).*

[Rocktaschel et al.2015] T. Rocktaschel, E. Grefenstette, K. M. Hermann, T. Kocisky and P. Blunsom. 2015. Reasoning about Entailment with Neural Attention. *arXiv preprint arXiv: 1509.06664.*

[Weston et al.2015] J. Weston, S. Chopra, and A. Bordes. 2015. Memory Networks. In *ICLR*.

[Sukhbaatar et al.2015] S. Sukhbaatar, A. Szlam, J. Weston and R. Fergus. 2015. End-to-End Memory Networks. In *NIPS*.

[Sutskever et al.2014] Ilya Sutskever, Oriol Vinyals and Quoc V. Le. 2014. Sequence to Sequence Learning with Neural Networks. In *NIPS*.

[Kumar et al.2015] A. Kumar, O. Irsoy, P. Ondruska, M. Iyyer, J. Bradbury, I. Gulrajani and R. Socher. 2015. Ask Me Anything: Dynamic Memory Networks for Natural Language Processing. *arXiv preprint arXiv: 1506.07285.*

[Tai et al.2015] Kai Sheng Tai, Richard Socher and Christopher D. Manning. 2015. Improved Semantic Representation From Tree-Structured Long Short-Term Memory Networks. In *Annual Meeting of the Association for Computational Linguistics (ACL).*

[Hochreiter and Schmidhuber1997] S. Hochreiter and J. Schmidhuber. 1997. Long short-term memory.. *Neural Computation*, 9(8):1735–1780, Nov 1997.

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

[Pennington et al.2014] Jeffrey Pennington, Richard Socher and Christopher D. Manning. 2014. Glove: Global Vectors for Word Representation. In *EMNLP 2014.*

[Duchi et al.2011] John Duchi, Elad Hazan and Yoram Singer. 2011. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *The Journal of Machine Learning Research* 12:2121–2159.

[Hinton et al.2012] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever and R. R. Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv: 1207.0580*

[Marelli et al.2014] Marco Marelli, Luisa Bentivogli, Marco Baroni, Raffaella Bernardi, Stefano Menini and Roberto Zamparelli. 2014. SemEval-2014 Task 1: Evaluation of compositional distributional semantic models on full sentences through se- mantic relatedness and textual entailment. In *SemEval 2014.*

[Bengio et al.1994] Yoshua Bengio, Patrice Simard and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is diffcult. In *IEEE Transactions on Neural Networks* 5(2):157–166.

[Hochreiter1998] Sepp Hochreiter. 1998. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 6(02):107–116.

[Lai and Hockenmaier2014] Alice Lai and Julia Hockenmaier. 2014. Illinois-lh: A denotational and distributional approach to semantics. In *SemEval 2014*.

[Jimenez et al.2014] Jimenez, Sergio, George Duenas, Julia Baquero, Alexander Gelbukh, Av Juan Dios Batiz, and Av Mendizabal. 2014. UNAL-NLP: Combining soft cardinality features for semantic textual similarity, relatedness and entailment. In *SemEval 2014*.

[Bjerva et al.2014] Johannes Bjerva, Johan Bos, Rob van der Goot, and Malvina Nissim. 2014. The meaning factory: Formal semantics for recognizing textual entailment and determining semantic similarity. In *SemEval 2014*.

[Zhao et al.2014] Jiang Zhao, Tian Tian Zhu, and Man Lan. 2014. Ecnu: One stone two birds: Ensemble of heterogenous measures for semantic relatedness and textual entailment. In *SemEval 2014*.

[Kiros et al.2015] Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun and Sanja Fidler. 2015. Skip-Thought Vectors. In *NIPS 2015*.

[Blunsom et al.2014] Phil Blunsom, Edward Grefenstette, Nal Kalchbrenner, et al. 2014. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*.

[Le and Mikolov2014] Quoc V Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. *arXiv preprint arXiv:1405.4053*.

[Kim2014] 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.

[Irsoy and Cardie2014] Ozan Irsoy and Claire Cardie. 2014. Deep recursive neural networks for compositionality in language. *Advances in Neural Information Processing Systems (NIPS)*, pages 2096–2104.