Dynamic Compositional Neural Networks over Tree Structure

Pengfei Liu, Xipeng Qiu*, Xuanjing Huang
Shanghai Key Laboratory of Intelligent Information Processing, Fudan University
School of Computer Science, Fudan University
825 Zhangheng Road, Shanghai, China
{pfliu14,xpqiu,xjhuang}@fudan.edu.cn

Abstract

Tree-structured neural networks have proven to be effective in learning semantic representations by exploiting syntactic information. In spite of their success, most existing models suffer from the underfitting problem: they recursively use the same shared compositional function throughout the whole compositional process and lack expressive power due to inability to capture the richness of compositionality. In this paper, we address this issue by introducing the dynamic compositional neural networks over tree structure (DC-TreeNN), in which the compositional function is dynamically generated by a meta network. The role of meta-network is to capture the metaknowledge across the different compositional rules and formulate them. Experimental results on two typical tasks show the effectiveness of the proposed models.

1 Introduction

Learning the distributed representation for long spans of text from its constituents has been a key step for various natural language processing (NLP) tasks, such as text classification [Zhao *et al.*, 2015; Liu *et al.*, 2015], semantic matching [Liu *et al.*, 2016a; 2016b], and machine translation [Cho *et al.*, 2014]. Existing deep learning approaches take a compositional function with different forms to compose word vectors recursively until obtaining a sentential representation. Typically, these compositional functions involve recurrent neural networks [Hochreiter and Schmidhuber, 1997], convolutional neural networks [Kalchbrenner *et al.*, 2014], and tree-structured neural networks [Tai *et al.*, 2015; Zhu *et al.*, 2015].

Among these methods, tree-structured neural networks (Tree-NNs) show theirs superior performance in many NLP tasks [Socher *et al.*, 2012]. Following the syntactic tree structure, Tree-NNs assign a fixed-length vector to each word at the leaves of the tree, and combine word and phrase pairs recursively to create intermediate node vectors, eventually obtaining one final vector to represent the whole sentence.

However, these models have a major limitation in their inability to fully capture the richness of compositionality [Socher *et al.*, 2013a]. The same parameters are used for all kinds of semantic compositions, even though the compositions have different characteristics in nature. For example, the composition of the adjective and the noun differs significantly from the composition of the verb and the noun. Moreover, many semantic phenomena, such as semantic idiomaticity or transparency, call for more powerful compositional mechanisms [Pylkkänen and McElree, 2006]. Therefore, Tree-NNs suffer from the underfitting problem.

To alleviate this problem, some researchers propose to use multiple compositional functions, which are arranged beforehand according to some partition criterion [Socher *et al.*, 2012; 2013a; Dong *et al.*, 2014]. Intuitively, using different parameters for different types of compositions has the potential to greatly reduce underfitting. Socher *et al.* [2013a] defined different compositional functions in terms of syntactic categories, and a suitable compositional function is selected based on the syntactic categories. Dong *et al.* [2014] introduced multiple compositional functions and during compositional phase, a proper one is selected based on the input information. Although these models accomplished their mission to a certain extent, they still suffer from the following three challenges. First, the predefined compositional functions cannot cover all the compositional rules; Second, they require more learnable parameters, suffering from the problem of overfitting; Third, it is difficult to determine a universal criterion for semantic composition based solely on syntactic categories.

In this paper, we propose dynamic compositional neural networks over tree structure, in which a *meta network* is used to generate the context-specific parameters of a *dynamic compositional network*. Specifically, we construct our models based on two kinds of tree-structured neural networks: recursive neural network (Tree-RecNN) [Socher *et al.*, 2012] and tree-structure long short-term memory neural network (Tree-LSTM) [Tai *et al.*, 2015]. Our work is inspired by recent work on dynamic parameter prediction [De Brabandere *et al.*, 2016; Bertinetto *et al.*, 2016; Ha *et al.*, 2016]. The meta network is used to extract the shared meta-knowledge across different compositional rules and to dynamically generate the context-specific compositional function. Thus, the compositional function of our models varies with positions, contexts and samples. The dynamic compositional network then applies those context-specific parameters to the current input information. Both meta and dynamic networks are differen-

*Corresponding author.

tiable such that the overall networks can be trained in an end-to-end fashion. Additional, to reduce the complexity of the whole networks, we define the dynamic weight matrix in a manner simulating low-rank matrix decomposition.

We evaluate our models on two typical tasks: text classification and text semantic matching. The results show that our models are more expressive due to their learning to learn nature, yet without increasing the number of model's parameters. Moreover, we find certain composition operations can be learned implicitly by meta TreeNN, such as the composition of noun phrases and verb phrases.

The contributions of the paper can be summed up as follows.

1. We provide a new perspective on how to compose the individual word of a sentence. Instead of directly using a learnable parameterized compositional function, we introduce a meta neural network, which can generate a compositional network to dynamically compose constituents over tree structure.
2. Experimental results show that the proposed architecture is more expressive due to its learning to learn nature, yet without increasing the number of model's parameters.
3. We present an elaborate qualitative analysis, giving an intuitive understanding on how our model works from semantic and syntactic perspectives.

2 Tree-Structured Neural Network

In this section, we briefly describe the tree-structured neural networks.

The idea of tree-structured neural networks for natural language processing (NLP) is to train a deep learning model with a grammatical tree structure [Pollack, 1990] that can be applied to phrases and sentences. At every node in the tree, the contexts of the left and right children are combined by a compositional function. The parameters of the compositional function are shared across all nodes in the tree. The layer computed at the top node gives a representation for the whole sentence.

2.1 Vanilla Recursive Neural Network

The simplest member of tree-structured NN is the vanilla recursive neural network [Socher *et al.*, 2013a], in which the representation of parent is calculated by weighted linear combination of the child vectors.

Formally, given a binary constituency tree T induced by a sentence, each non-leaf node corresponds to a phrase. We refer to $\mathbf{h}_j \in \mathbb{R}^d$ as the hidden state of each node j, and let $\mathbf{h}_j^l$, $\mathbf{h}_j^r$ denote the left and right child representations respectively.

$$\mathbf{h}_j = \tanh\left(\mathbf{W}\begin{bmatrix}\mathbf{h}_j^l \\ \mathbf{h}_j^r\end{bmatrix} + \mathbf{b}\right), \tag{1}$$

where $\mathbf{W} \in \mathbb{R}^{d\times 2d}$ is a learnable compositional matrix, $\mathbf{b}$ is the bias vector.

2.2 Tree LSTM

Tree LSTM [Tai *et al.*, 2015] is a generalization of LSTMs to tree-structured network topologies. In this model, the compositional function is an LSTM unit, and the hidden state h_j of each node can be computed as follows: we refer to $\mathbf{h}_j$ and $\mathbf{c}_j$ as the hidden state and memory cell of each node j. The transition equations of node j are as follows:

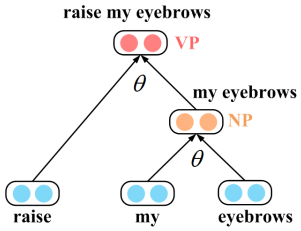


Figure 1: Illustration of Vanilla Tree Structure Network. NP and VP represent noun and verb phrases respectively. θ denotes the shared parameters of compositional function.

$$\begin{bmatrix}\tilde{\mathbf{c}}_j \\ \mathbf{o}_j \\ \mathbf{i}_j \\ \mathbf{f}_j^l \\ \mathbf{f}_j^r\end{bmatrix} = \begin{bmatrix}\tanh \\ \sigma \\ \sigma \\ \sigma \\ \sigma\end{bmatrix}\left(\mathbf{W}\begin{bmatrix}\mathbf{x}_j \\ \mathbf{h}_j^l \\ \mathbf{h}_j^r\end{bmatrix} + \mathbf{b}\right), \tag{2}$$

$$\mathbf{c}_j = \tilde{\mathbf{c}}_j \odot \mathbf{i}_j + \mathbf{c}_j^l \odot \mathbf{f}_j^l + \mathbf{c}_j^r \odot \mathbf{f}_j^r, \tag{3}$$

$$\mathbf{h}_j = \mathbf{o}_j \odot \tanh\left(\mathbf{c}_j\right), \tag{4}$$

where $\mathbf{x}_j \in \mathbb{R}^d$ denotes the input vector and is non-zero if and only if it is a leaf node. The superscript l and r represent the left child and right child respectively. σ represents the logistic sigmoid function and $\odot$ denotes element-wise multiplication. $\mathbf{W} \in \mathbb{R}^{5d\times 3d}$ and $\mathbf{b} \in \mathbb{R}^{5d}$ are learnable parameters.

3 Dynamic Compositional Neural Network

In the above two tree-structured NNs, the compositional function is shared across all nodes in the tree, which results in underfitting since the semantic compositions have great diversities. To address this problem, we propose two dynamic compositional neural networks over tree structure, which dynamically generate different parameters for different types of compositions. Figure 2 shows an illustration of the dynamic compositional neural network, consisting of two components: (1) meta network and (2) basic network with dynamic parameters.

Specifically, we propose two meta networks to generate the context-specific compositional functions for RecNN and TreeLSTM respectively.

3.1 Meta Network for RecNN

For RecNN, we replace the static parameters $\mathbf{W}$ and $\mathbf{b}$ in Eq.(1) with the dynamic parameters $\mathbf{W}(\mathbf{z}_j)$ and $\mathbf{b}(\mathbf{z}_j)$, which are generated by a meta network. The meta network is a smaller RecNN, and the hidden state $\hat{\mathbf{h}}_j \in \mathbb{R}^m$ of node j in meta network is defined as

$$\hat{\mathbf{h}}_j = \tanh\left(\mathbf{W}_m\mathcal{H}_j + \mathbf{b}_m\right), \tag{5}$$

$$\mathbf{z}_j = \mathbf{W}_z\hat{\mathbf{h}}_j \tag{6}$$

where $\mathcal{H}_j = \mathbf{h}_j^l \oplus \mathbf{h}_j^r \oplus \hat{\mathbf{h}}_j^l \oplus \hat{\mathbf{h}}_j^r \in \mathbb{R}^{2m+2d}$, $\mathbf{W}_m \in \mathbb{R}^{m\times(2d+2m)}$ and $\mathbf{b}_m \in \mathbb{R}^m$ are parameters of meta RecNN; $\mathbf{W}_z \in \mathbb{R}^{z\times m}$ is a scale matrix.

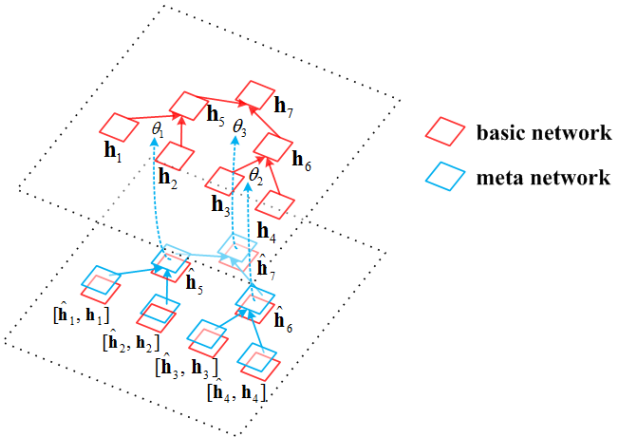


Figure 2: Dynamic Compositional Neural Network. θ denotes the context-dependent parameters generated by meta TreeNN.

To reduce the number of the parameters, we define the dynamic parameters with a low-rank factorized representation of the weights, analogous to the Singular Value Decomposition. The dynamic parameters $\mathbf{W}(\mathbf{z}_j)$ and $\mathbf{b}(\mathbf{z}_j)$ of the basic RecNN are computed by:

$$\mathbf{W}(\mathbf{z}_j) = \begin{bmatrix} P_l \mathbf{D}(\mathbf{z}_j) Q_l \\ P_r \mathbf{D}(\mathbf{z}_j) Q_r \end{bmatrix} \tag{7}$$

$$\mathbf{b}(\mathbf{z}_j) = \begin{bmatrix} B_l \mathbf{z}_j \\ B_r \mathbf{z}_j \end{bmatrix} \tag{8}$$

where $P \in \mathbb{R}^{d \times z}$, $Q \in \mathbb{R}^{z \times d}$, and $\mathbf{D}(\mathbf{z}_t) \in \mathbb{R}^{z \times z}$ is the diagonal matrix of $\mathbf{z}$.

Thus, our dynamic RecNN needs $(6dz + mz)$ parameters, while the vanilla RecNN has $(2d^2 + d)$ parameters. With a small z and m, our dynamic RecNN needs less parameters than the vanilla RecNN. For example, if we set $d = 100$ and $z = m = 20$, our model needs $12,400$ parameters while the vanilla model needs $20,100$ parameters.

3.2 Meta Network for TreeLSTM

Likewise, we also use a smaller meta network to generate the static parameters $\mathbf{W}$ and $\mathbf{b}$ in Eq.(9) with the dynamic parameters $\mathbf{W}(\mathbf{z}_j)$ and $\mathbf{b}(\mathbf{z}_j)$. The meta network is a smaller TreeLSTM, and the hidden state $\hat{\mathbf{h}}_j \in \mathbb{R}^m$ of node j in meta network is defined as

$$\begin{bmatrix} \hat{\mathbf{g}}_j \\ \hat{\mathbf{o}}_j \\ \hat{\mathbf{i}}_j \\ \hat{\mathbf{f}}_j^l \\ \hat{\mathbf{f}}_j^r \end{bmatrix} = \begin{bmatrix} \tanh \\ \sigma \\ \sigma \\ \sigma \\ \sigma \end{bmatrix} \left(\mathbf{W}_m \begin{bmatrix} \mathbf{x}_j \\ \mathcal{H}_j \end{bmatrix} + \mathbf{b}_m \right), \tag{9}$$

$$\hat{\mathbf{c}}_j = \hat{\mathbf{g}}_j \odot \hat{\mathbf{i}}_j + \hat{\mathbf{g}}_j^l \odot \hat{\mathbf{f}}_j^l + \hat{\mathbf{g}}_j^r \odot \hat{\mathbf{f}}_j^r, \tag{10}$$

$$\hat{\mathbf{h}}_j = \hat{\mathbf{o}}_j \odot \tanh\left(\hat{\mathbf{c}}_j\right), \tag{11}$$

$$\mathbf{z}_j = \mathbf{W}_z \hat{\mathbf{h}}_j \tag{12}$$

where $\mathcal{H}_j = \mathbf{h}_j^l \oplus \mathbf{h}_j^r \oplus \hat{\mathbf{h}}_j^l \oplus \hat{\mathbf{h}}_j^r \in \mathbb{R}^{2m+2d}$; $\mathbf{W}_m \in \mathbb{R}^{5m \times (3d+2m)}$ and $\mathbf{b}_m \in \mathbb{R}^m$ are parameters of meta TreeLSTM; $\mathbf{W}_z \in \mathbb{R}^{z \times m}$ is a scale matrix.

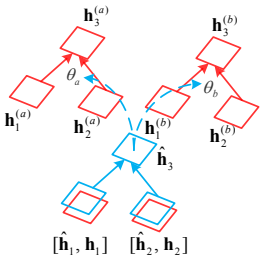


Figure 3: Illustration of DC-TreeNN Matching Network.

The dynamic parameters $\mathbf{W}(\mathbf{z}_j)$ and $\mathbf{b}(\mathbf{z}_j)$ of basic TreeLSTM are computed by:

$$\mathbf{W}(\mathbf{z}_j) = \left[\mathbf{W}^g, \mathbf{W}^i, \mathbf{W}^{f^l}, \mathbf{W}^{f^r}, \mathbf{W}^o \right] \tag{13}$$

$$\mathbf{b}(\mathbf{z}_j) = \left[\mathbf{b}^g, \mathbf{b}^i, \mathbf{b}^{f^l}, \mathbf{b}^{f^r}, \mathbf{b}^o \right], \tag{14}$$

where for $* \in \{c, o, i, f^l, f^r\}$,

$$\mathbf{W}^*(\mathbf{z}_j) = \begin{bmatrix} P_x^* \mathbf{D}(\mathbf{z}_t) Q_x^* \\ P_l^* \mathbf{D}(\mathbf{z}_t) Q_l^* \\ P_r^* \mathbf{D}(\mathbf{z}_t) Q_r^* \end{bmatrix}, \tag{15}$$

$$\mathbf{b}^*(\mathbf{z}_j) = \begin{bmatrix} B_x^* \mathbf{z}_t \\ B_l^* \mathbf{z}_t \\ B_r^* \mathbf{z}_t \end{bmatrix}, \tag{16}$$

where $P \in \mathbb{R}^{5d \times z}$, $Q \in \mathbb{R}^{z \times 3d}$, $B \in \mathbb{R}^{5d \times z}$, and $\mathbf{D}(\mathbf{z}_t) \in \mathbb{R}^{z \times z}$ is the diagonal matrix of $\mathbf{z}$.

With a small z and m, our dynamic TreeLSTM needs a similar amount of parameters compared to the standard TreeLSTM.

4 Application of Dynamic Compositional Neural Networks

In this section, we describe two specific models to show the applications of dynamic compositional neural networks for two typical tasks in NLP.

4.1 Text Classification

The purpose of text classification is that, given a sentence x, the model should predict labels $\hat{y}$ from a pre-defined label set $\mathcal{Y}$. From the description in the previous section, we can compute the distributed representation $\mathbf{h}_j$ of the phrase at node j of a tree:

$$\mathbf{h}_j = \text{DC-TreeNN}(\mathbf{x}_j, \mathbf{h}_j^l, \mathbf{h}_j^r, \theta) \tag{17}$$

After this recursive process, the hidden state $\mathbf{h}_R$ at the root node is used as the sentential representation, which then followed by a softmax classifier to predict the probability distribution over classes.

$$\hat{\mathbf{y}} = \text{softmax}(\mathbf{W}_t \mathbf{h}_R + \mathbf{b}_t) \tag{18}$$

where $\hat{\mathbf{y}}$ is prediction probabilities, $\mathbf{W}_t$ and $\mathbf{b}_t$ are the parameters of the classifier.

4.2 Text Semantic Matching

Among many natural language processing (NLP) tasks, a common problem is modelling the relevance of a pair of texts. In this section, we show how to effectively use the dynamic compositional neural networks to model the semantic relationship between two sentences.

As shown in Figure 3, given two sentences x_a and x_b, the representation of each sentence $\mathbf{h}_R$ can be computed by one basic TreeNN.

$$\mathbf{h}_R^{(a)} = \text{DC-TreeNN}(\mathbf{x}_R, \mathbf{h}_R^l, \mathbf{h}_R^r, \theta_a) \quad (19)$$

$$\mathbf{h}_R^{(b)} = \text{DC-TreeNN}(\mathbf{x}_R, \mathbf{h}_R^l, \mathbf{h}_R^r, \theta_b) \quad (20)$$

where R denotes the root node of a tree. θ_a and θ_b are generated by a shared meta TreeNN. Then, the representation of each sentence will be fed into a multi-layer perceptron to obtain a unified representation for the final relationship classification.

The sample-specific but shared meta TreeNN ensures that, on the one hand we can dynamically model the diversity of semantic compositionality, on the other hand we can capture the general rules across different samples.

5 Related Work

One thread of related work is the exploration of different kinds of compositional function over tree structures. Socher *et al.* [2012] proposed the recursive neural network with standard compositional function (RecNN). After that, some extensions are introduced to enhance the expressive power of compositional function, such as MV-RecNN [Socher *et al.*, 2013b], SU-RNN [Socher *et al.*, 2013a], RNTN [Socher *et al.*, 2013b], while these models suffer from the problem of hard-coded compositional operations and overfitting.

Another thread of work is the idea of using one network to direct the learning of another network [De Brabandere *et al.*, 2016]. Naik and Mammone [1992] introduce a meta neural network to provide another network with a step size and a direction vector, which is helpful for parameter optimization. De Brabandere *et al.* [2016] propose the dynamic filter network to implicitly learn a variety of filtering operations. Bertinetto *et al.* [2016] introduce a learnet for one-shot learning, which can predict the parameters of a second network given a single exemplar. Ha *et al.* [2016] propose the model hypernetwork, which uses a small network to generate the weights for a larger network.

Different from these models, we employ the idea of parameter generation to address the limitation of weight-sharing or partially sharing paradigm of tree-based compositional models.

6 Experiment

To make a comprehensive evaluation, we assess our model on five text classification tasks and a semantic matching task.

6.1 Initialization and Hyperparameters

The word embeddings for all of the models are initialized with GloVe vectors [Pennington *et al.*, 2014]. The other parameters are initialized by randomly sampling from uniform distribution in $[-0.1, 0.1]$.

Hyper-Param.	IE	MR	SST	SUBJ	QC	SICK
d	100	100	100	100	100	50(30)
e	100	100	100	100	100	50(50)
m=z	40	30	30	20	40	40(20)

Table 1: Hyper-parameters for our models on all tasks. m, d denote the size of hidden state in meta and basic TreeNN respectively. e and z represent the size of embedding vector $\mathbf{x}$ and controlling vector $\mathbf{z}$. The settings of our two proposed models on all datasets are the same except SICK: the numbers inside and outside parentheses correspond to DC-RecNN and DC-TreeLSTM respectively.

Dataset	Train	Dev.	Test	Class	L_{avg}	$\lvert\mathcal{V}\rvert$
MR	9596	-	1066	2	22	21K
SST	6920	872	1821	2	18	15K
SUBJ	9000	-	1000	2	21	21K
IE	2221	-	300	3	16	7.5K
QC	5452	-	500	6	10	9.4K

Table 2: Statistics of the five mainstream datasets for text classification. L_{avg} denotes the average length of documents; $|\mathcal{V}|$ denotes the size of vocabulary.

The final hyper-parameters are as follows. The initial learning rate is 0.1. The regularization weight of the parameters is $1E{-}5$ and the others are listed as Table 1.

For all the sentences in the datasets, we parse them with constituency parser [Klein and Manning, 2003] to obtain the trees for our models and some competitor models.

6.2 Text Classification

We evaluate our models on five different datasets. The detailed statistics about the five datasets are listed in Table 2. Each dataset is briefly described as follows.

- **SST** The movie reviews with two classes (negative, positive) in the Stanford Sentiment Treebank [Socher *et al.*, 2013b].
- **MR** The movie reviews with two classes [Pang and Lee, 2005].
- **QC** The TREC questions dataset involves six different question types. [Li and Roth, 2002].
- **SUBJ** Subjectivity dataset where the goal is to classify each instance (snippet) as being subjective or objective. [Pang and Lee, 2004]
- **IE** Idiom enhanced sentiment classification. [Williams *et al.*, 2015]. Each sentence contains at least one idiom.

Results As shown in Table 3, DC-TreeLSTM consistently outperforms RecNN, MV-RecNN, RNTN, and TreeLSTM by a large margin while achieving comparable results to the CNN and using much fewer parameters.(The number of parameters in our models is approximately 10K while in CNN the number of parameters is about 400K). Compared with RecNN, DC-RecNN performs better, indicating the effectiveness of the dynamic compositional function. Additionally, both DC-RecNN and DC-TreeLSTM achieve substantial improvement on IE dataset, which covers the richness of com-

Model	IE	MR	SST	SUBJ	QC
NBOW	54.6	77.2	80.5	91.3	88.2
DCNN	-	-	86.8	-	93.0
CNN-multichannel	-	81.5	**88.1**	93.4	93.6
RecNN	52.0	76.4	82.4	90.6	88.8
MV-RecNN	54.8	76.8	82.9	90.9	89.2
RNTN	55.7	75.8	85.4	92.1	88.9
TreeLSTM	56.0	78.7	86.9	91.0	91.6
DC-RecNN	58.2	80.2	86.1	93.5	91.2
DC-TreeLSTM	**60.2**	**81.7**	87.8	**93.7**	**93.8**

Table 3: Accuracies of our models on five datasets against state-of-the-art neural models. **DCNN**: Dynamic Convolutional Neural Network [Kalchbrenner *et al.*, 2014]. **CNN-multichannel**: Convolutional Neural Network [Kim, 2014].

Model	Hidden	Train	Test
NBOW	30	96.6	73.4
LSTM	100	100.0	71.3
RecNN	30	95.4	74.9
MV-RecNN	50	95.9	75.5
RNTN	50	97.8	76.9
TreeLSTM	50	95.9	77.5
DC-RecNN	30	96.5	77.9
DC-TreeLSTM	50	98.5	**80.2**

Table 4: Evaluation results of our models on the SICK train and test sets.

positionality (idiomaticity). We attribute the success on IE to its power in modeling more complicated compositionality.

6.3 Text Semantic Matching

We choose the dataset of Sentences Involving Compositional Knowledge (SICK), which is proposed by Marelli *et al.* [2014] aiming at evaluation of compositional distributional semantic models. The dataset consists of 9927 sentence pairs in a 4500/500/4927 train/dev/test split, in which each sentence pairs are pre-defined into three labels: "`entailment`","`contradiction`" and "`neutral`".

Results Our results are summarized in Table 4, where the performance of NBOW, LSTM, RecNN, and RNTN are reported by [Bowman *et al.*, 2015; 2014]. For fair comparison, we train our models with the same setting. We can see both DC-RecNN and DC-TreeLSTM outperform competitor models, in which DC-RecNN (DC-TreeLSTM) achieves 3% (2.7%) improvements than RecNN (TreeLSTM). We think this breakthrough is basically attributed to the dynamic compositional mechanism, which enables our models to capture various syntactic patterns (As we will discuss later) therefore can more accurately understand sentences.

6.4 Discussion and Qualitative Analysis

In our models, the latent vector $\mathbf{z}$ controls the process of predicting network's parameters and its dimensionality determines the number of model's parameters. Next, we will investigate how the controlling vector $\mathbf{z}$ influences the performance of our models.

Impact of the Dimensionality Figure 4 shows the accuracies of DC-RecNN across the different dimensions of $[5, 10, \ldots, 50]$ for the controlling vector $\mathbf{z}$ on five datasets. We get the following findings:

- For all five datasets, the model can achieve considerable performances even when the size of vector $\mathbf{z}$ is reduced to 5. Particularly, for the dataset QC, the model obtains 87.0% accuracy with a pretty small meta Tree-RecNN[1], suggesting a smaller meta network can be used for generating a more powerful compositional function to effectively model sentence.
- When dealing with the dataset with more labels, larger vector size leads to a better performance. For example, the performance on IE and QC datasets reaches the maximum when the size of z equals 40, while for the other three datasets MR, SST and SUBJ, the model obtains the best performance with the value of 30, 30 and 20 respectively.

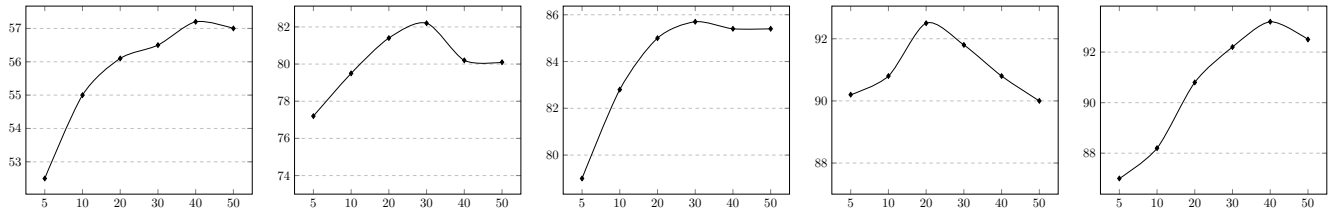
Understanding the Neuron's Behaviours As described in previous sections, we know the compositional function is changed cross child nodes over a tree, which is controlled by a latent vector z. To get an intuitive understanding of how the controlling vector z works, we design an experiment to examine the neuron's behaviours of $\mathbf{z}$ on each node. More concretely, we refer to z_{jk} as the activation of the k-neuron at node j, where $j \in \{1, \ldots, N\}$ and $k \in \{1, \ldots, z\}$. Then we randomly sample some sentences on the development set from the datasets we used. By visualizing the latent vector $\mathbf{z}_j$ and analyzing the maximum activation, we can find what kinds of patterns the current neuron focuses on.

Table 5 illustrates multiple interpretable neurons and some representative words or phrases which can activate these neurons. We can observe that:

- For some simple tasks such as text classification, meta network will integrate useful semantic information into the the generation process of compositional function. These semantic bias before composition are task-specific.

  For example, the 21-st neuron is more sensitive to emotional terms, which can be understood as a sentinel, telling the basic neural network that an informative phrase is coming, more attention should be paid in the process of composition. Figure 5-(a) shows a visualization. We can see in this sentence, the neuron has realized that this idiomatic collocation "`in stitches`" is a key pattern, which is crucial for the final sentiment prediction.
- For more complicated tasks such as semantic matching, a well-grounded understanding of the syntactic structure is crucial. In this context, we find that a

[1]With the same parameters, the RecNN obtain 74% accuracy in our implementation



(a) IE (b) MR (c) SST (d) SUBJ (e) QC

Figure 4: Performances of DC-RecNN with the different sizes of latent vector z on five development datasets: IE, MR, SST, SUBJ, and QC. Y-axis represents the accuracy(%), and X-axis represents dimensionality of z.

Type		Neurons	Examples	Explanations
Semantic	Lexical	17-th	fun, glad, terrific, wonderful, refreshing	Words related to sentiment
	Phrasal	21-st	pick holes, see red, in stitches, split hairs	Phrases related to sentiment
Syntactic	Noun Phrase	45-th	blond boy, pink shirt, green grass, black dog	Containing modifiers related to color
	Verb Phrase	27-th	waking up, take off, pulling up, driving down	Phrases constructed by light-verb
		11-th	slicing a potato, playing guitar, chopping butter	Verb-object phrases
	Prep. Phrase	13-th	on a track, in rocky area, on a stage, over water	Phrases related to places

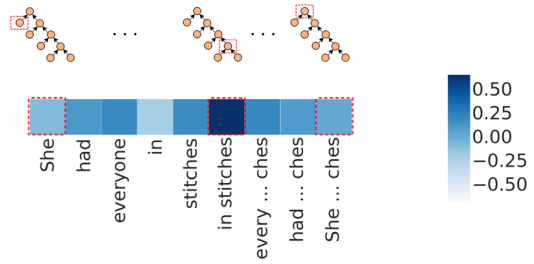
Table 5: Multiple interpretable neurons and the words/phrases captured by these neurons. The last column gives the explanations of corresponding neuron's behaviours.

meta network could capture some syntactic information. For example, the 27-th neuron monitors phrases constructed by light-verb. As shown in Figure 5-(b), the verb phrase "`taking off`" has been attended for forthcoming compositional operation, which is more useful for judging the semantic relation between the sentence pair "`An airplane is taking off/A plane is landing`".

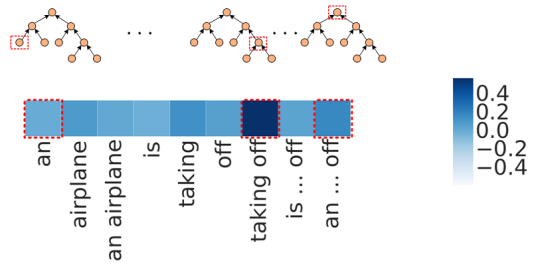
7 Conclusion

In this work, we introduce a meta neural network, which can generate a compositional network to dynamically compose constituents over tree structure. The parameters of compositional function vary from position to position and from sample to sample, allowing for more sophisticated operations on the input.

To evaluate our models, we choose two typical NLP tasks involving six datasets. The qualitative and quantitative experiment results demonstrate the effectiveness of our models.



(a) The behaviour of 21-st neuron for sentence "`She had everyone in stitches`"



(b) The behaviour of 27-th neuron for sentence "`An airplane is taking off`"

Figure 5: The two heat maps describe the behaviours of neurons $\mathbf{z}_{21}$ and $\mathbf{z}_{27}$ from DC-TreeNN.

Acknowledgments

We would like to thank the anonymous reviewers for their valuable comments and thank Kaiyu Qian, Jiachen Xu, Jifan Chen for useful discussions. This work was partially funded by National Natural Science Foundation of China (No. 61532011 and 61672162), Shanghai Municipal Science and Technology Commission (No. 16JC1420401).

References

[Bertinetto *et al.*, 2016] Luca Bertinetto, João F Henriques, Jack Valmadre, Philip Torr, and Andrea Vedaldi. Learning feed-forward one-shot learners. In *Advances in NIPS*, pages 523–531, 2016.

[Bowman *et al.*, 2014] Samuel R Bowman, Christopher Potts, and Christopher D Manning. Recursive neural networks can learn logical semantics. *arXiv preprint arXiv:1406.1827*, 2014.

[Bowman *et al.*, 2015] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on EMNLP*, 2015.

[Cho *et al.*, 2014] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Proceedings of EMNLP*, 2014.

[Collobert *et al.*, 2011] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *The JMLR*, 12:2493–2537, 2011.

[De Brabandere *et al.*, 2016] Bert De Brabandere, Xu Jia, Tinne Tuytelaars, and Luc Van Gool. Dynamic filter networks. In *NIPS*, 2016.

[Dong *et al.*, 2014] Li Dong, Furu Wei, Chuanqi Tan, Duyu Tang, Ming Zhou, and Ke Xu. Adaptive recursive neural network for target-dependent twitter sentiment classification. In *ACL (2)*, pages 49–54, 2014.

[Duchi *et al.*, 2011] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *The JMLR*, 12:2121–2159, 2011.

[Ha *et al.*, 2016] David Ha, Andrew Dai, and Quoc V Le. Hypernetworks. *arXiv preprint arXiv:1609.09106*, 2016.

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

[Kalchbrenner *et al.*, 2014] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A convolutional neural network for modelling sentences. In *Proceedings of ACL*, 2014.

[Kim, 2014] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[Klein and Manning, 2003] Dan Klein and Christopher D Manning. Accurate unlexicalized parsing. In *Proceedings of the 41st ACL*, pages 423–430, 2003.

[Li and Roth, 2002] Xin Li and Dan Roth. Learning question classifiers. In *Proceedings of the 19th International Conference on Computational Linguistics*, pages 556–562, 2002.

[Liu *et al.*, 2015] PengFei Liu, Xipeng Qiu, Xinchi Chen, Shiyu Wu, and Xuanjing Huang. Multi-timescale LSTM for modelling sentences and documents. In *Proceedings of the Conference on EMNLP*, 2015.

[Liu *et al.*, 2016a] Pengfe Liu, Xipeng Qiu, Jifan Chen, and Xuanjing Huang. Deep fusion LSTMs for text semantic matching. In *Proceedings of ACL*, 2016.

[Liu *et al.*, 2016b] Pengfei Liu, Xipeng Qiu, Yaqian Zhou, Jifan Chen, and Xuanjing Huang. Modelling interaction of sentence pair with coupled-lstms. In *Proceedings of the 2016 Conference on EMNLP*, November 2016.

[Marelli *et al.*, 2014] Marco Marelli, Luisa Bentivogli, Marco Baroni, Raffaella Bernardi, Stefano Menini, and Roberto Zamparelli. Semeval-2014 task 1: Evaluation of compositional distributional. *SemEval-2014*, 2014.

[Naik and Mammone, 1992] Devang K Naik and RJ Mammone. Meta-neural networks that learn by learning. In *Neural Networks, 1992. IJCNN.*, volume 1, pages 437–442. IEEE, 1992.

[Pang and Lee, 2004] Bo Pang and Lillian Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of ACL*, 2004.

[Pang and Lee, 2005] Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the ACL*, pages 115–124, 2005.

[Pennington *et al.*, 2014] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. *Proceedings of the EMNLP*, 12:1532–1543, 2014.

[Pollack, 1990] Jordan B Pollack. Recursive distributed representations. *Artificial Intelligence*, 46(1):77–105, 1990.

[Pylkkänen and McElree, 2006] Liina Pylkkänen and Brian McElree. The syntax-semantics interface: On-line composition of sentence meaning. *Handbook of psycholinguistics*, 2:537–577, 2006.

[Socher *et al.*, 2012] Richard Socher, Brody Huval, Christopher D Manning, and Andrew Y Ng. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of EMNLP*, pages 1201–1211, 2012.

[Socher *et al.*, 2013a] Richard Socher, John Bauer, Christopher D Manning, and Andrew Y Ng. Parsing with compositional vector grammars. In *Proceedings of ACL*, 2013.

[Socher *et al.*, 2013b] Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of EMNLP*, 2013.

[Tai *et al.*, 2015] Kai Sheng Tai, Richard Socher, and Christopher D Manning. Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*, 2015.

[Williams *et al.*, 2015] Lowri Williams, Christian Bannister, Michael Arribas-Ayllon, Alun Preece, and Irena Spasić. The role of idioms in sentiment analysis. *Expert Systems with Applications*, 42(21):7375–7385, 2015.

[Zhao *et al.*, 2015] Han Zhao, Zhengdong Lu, and Pascal Poupart. Self-adaptive hierarchical sentence model. *arXiv preprint arXiv:1504.05070*, 2015.

[Zhu *et al.*, 2015] Xiao-Dan Zhu, Parinaz Sobhani, and Hongyu Guo. Long short-term memory over recursive structures. In *ICML*, pages 1604–1612, 2015.