Adaptive Semantic Compositionality for Sentence Modelling

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

Representing a sentence with a fixed vector has shown its effectiveness in various NLP tasks. Most of the existing methods are based on neural network, which recursively apply different composition functions to a sequence of word vectors thereby obtaining a sentence vector. A hypothesis behind these approaches is that the meaning of any phrase can be composed of the meanings of its constituents. However, many phrases, such as idioms, are apparently non-compositional. To address this problem, we introduce a parameterized compositional switch, which outputs a scalar to adaptively determine whether the meaning of a phrase should be composed of its two constituents. We evaluate our model on five datasets of sentiment classification and demonstrate its efficacy with qualitative and quantitative experimental analysis.

1 Introduction

Currently, there has been a surge of interest in learning distributed representations for sentences with neural models. Most of related work focuses on the question that how to obtain a desirable sentence vector given the vector representations of words in the sentence. Existing approaches take a compositional function with different forms to compose word vectors recursively until obtaining a sentence representation. Typically, these compositional functions involve recurrent neural networks with long short-term memory [Hochreiter and Schmidhuber, 1997], convolutional neural networks [Kalchbrenner *et al.*, 2014], and recursive neural networks [Socher *et al.*, 2013].

However, a potential weakness of these models is that not all the phrases are compositional. For example, most of the idioms are non-compositional, whose meanings are hard to be predicted from the meaning of their constituents. Besides, it may harm the performance of the model to assume that all the phrases are compositional. The words in idioms are uninformative or even can act as noisy samples. For example, for the sentence "`She will go bananas about your behaviour`", the phrase "`go bananas`" means "`get very angry`". We can hardly compose this meaning according to the words "`go`" and "`bananas`".

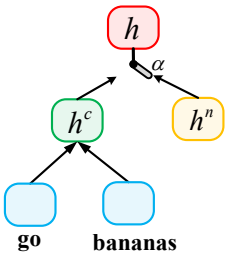


Figure 1: Illustration of adaptive semantic compositional neural network.

Based on this observation, we propose an adaptive semantic compositional model for sentence modeling as shown in 1, which can learn how to obtain the representation of a phrase: compositional or non-compositional way. In compositional way, the representation of a phrase is composed of its containing words or sub-phrases. In non-compositional way, the representation of a phrase is learned as a whole unit, which can be pre-trained from large scale unlabeled data. To do this, we adopt tree-structured Long Short-term Memory Network to recursively model a sentence over its syntactic tree, and introduce a parameterized controller, which can adaptively determine proper way (compositional/non-compositional way) to model a non-leaf node.

Additionally, considering that the compositionality of some phrases are context-dependent (e.g. "`around the corner`"), we propose to use a hybrid mechanism to learn context-sensitive phrase representations, which can disambiguates the meanings of the phrases.

We evaluate our models on five datasets of sentiment classification. The experimental results show the effectiveness of our proposed models. Furthermore, we present an elaborate qualitative analysis of our models, giving an intuitive understanding how our model worked.

The contributions of this paper can be summarized as follows.

1. By introducing a compositional switch and a hybrid mechanism, we grow the capacity of syntax-based neural sentence models, allowing it to model the non-compositional phrases when learning sentence represen-

*Corresponding author.

tations.

2. We propose two kinds of models to encode non-compositional phrases, one of which can address the problem of idiomatic variations.
3. Beyond quantitative measurement, we carefully perform qualitative analysis, and demonstrate why and how the idea works.

2 Neural Models for Sentence Modelling

The primary role of neural sentence modelling is to represent the variable-length sentence as a fixed-length vector. These models generally consist of a projection layer that maps words, sub-word units or n-grams to vector representations, and then compose them with different forms of compositional functions. Most of these models for distributed representations of sentences can be classified into three categories.

Sequence models Sequence models construct the representation of sentences based on the recurrent neural network (RNN) [Mikolov *et al.*, 2010] or the gated versions of RNN [Sutskever *et al.*, 2014; Liu *et al.*, 2015; 2016a]. Sequence models are sensitive to word order, but they have a bias towards the latest input words.

Convolutional models Convolutional neural network (CNN) is also used to model sentences [Collobert *et al.*, 2011; Kalchbrenner *et al.*, 2014; Hu *et al.*, 2014]. It takes as input the embeddings of words in the sentence aligned sequentially, and summarizes the meaning of a sentence through layers of convolution and pooling, until reaching a fixed length vectorial representation in the final layer.

Syntax-based models Syntax-based models compose the sentence representations following a given tree structure (i.e. Constituency tree). More specifically, the model computes parent vectors in a bottom up fashion using different types of compositional functions [Socher *et al.*, 2013; Tai *et al.*, 2015; Liu *et al.*, 2016b; 2017]. Lastly, The vector computed at the top node gives a representation for the sentence.

3 Syntax-based Long Short-term Memory Network

Syntax-based sentence models can be equipped with various kinds of compositional functions, such as neural tensor layer [Socher *et al.*, 2013] and tree-structured LSTM (T-LSTM) [Tai *et al.*, 2015]. Here, we introduce the latter model due to its superior performance in representing sentence meaning.

T-LSTM is a generalization of LSTMs to tree-structured network topologies. More formally, given a binary constituency tree T induced by a sentence, there are two child nodes for each non-leaf node. We refer to $\mathbf{h}_j$ and $\mathbf{c}_j$ as the hidden state and memory cell of node j. The transition equations of each node j are as follows:

$$\begin{bmatrix} \tilde{\mathbf{c}}_j \\ \mathbf{o}_j \\ \mathbf{i}_j \\ \mathbf{f}_j^l \\ \mathbf{f}_j^r \end{bmatrix} = \begin{bmatrix} \tanh \\ \sigma \\ \sigma \\ \sigma \\ \sigma \end{bmatrix} T_{\mathbf{A},\mathbf{b}} \begin{bmatrix} \mathbf{x}_j \\ \mathbf{h}_j^l \\ \mathbf{h}_j^r \end{bmatrix}, \tag{1}$$

$$\mathbf{c}_j = \tilde{\mathbf{c}}_j \odot \mathbf{i}_j + \mathbf{c}_j^l \odot \mathbf{f}_j^l + \mathbf{c}_j^r \odot \mathbf{f}_j^r, \tag{2}$$

$$\mathbf{h}_j = \mathbf{o}_j \odot \tanh(\mathbf{c}_j), \tag{3}$$

where $\mathbf{x_j}$ 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. $T_{\mathbf{A},\mathbf{b}}$ is an affine transformation which depends on parameters of the network $\mathbf{A}$ and $\mathbf{b}$.

4 Adaptive Semantic Compositional Neural Network

Existing neural sentence models are failure to understand non-compositional phrases in a sentence, which significantly affects one's understanding of the sentence.

To address this problem, we propose an adaptive semantic compositional neural network, which integrates a compositional switch into syntax-based neural sentence models and can adaptively learn the compositional and non-compositional representation of phrases over a syntax tree recursively thereby obtaining a sentence representation.

Specifically, the model consists of three parts: compositional component, non-compositional representation, and semantic compositional switch. Given a node x_j from a binary constituency tree, we refer its two children as x_j^l and x_j^r. And the hidden states of these three nodes are referred to as $\mathbf{h}_j$, $\mathbf{h}_j^l$, $\mathbf{h}_j^r$ respectively.

4.1 Compositional Encoder

The function of this module is to compose any two words or phrases with a parametric composition function. More formally, given two hidden states of the children, $\mathbf{h}_j^l$ and $\mathbf{h}_j^r$, the hidden state of their parent can be computed as:

$$\mathbf{h}_j^c = f(\mathbf{h}_j^l, \mathbf{h}_j^r) \tag{4}$$

where $f(,)$ represents the composition function, and particularly, here we use tree-structured LSTM (T-LSTM) [Tai *et al.*, 2015] unit as described in Eq. (1-3).

4.2 Non-compositional Encoder

Different with previous models, apart from compositional function, each phrase is assigned an extra vector $\mathbf{h}^n$ to represent the idiomatic meaning. To obtain $\mathbf{h}^n$, we propose two models as follows.

Retrieval Model (RM) In this model, the non-compositional phrases are directly retrieved from an external memory, which is constructed by a learnable matrix $\mathbf{M}$.

$$\mathbf{h}^n = \mathbf{M}[k] \tag{5}$$

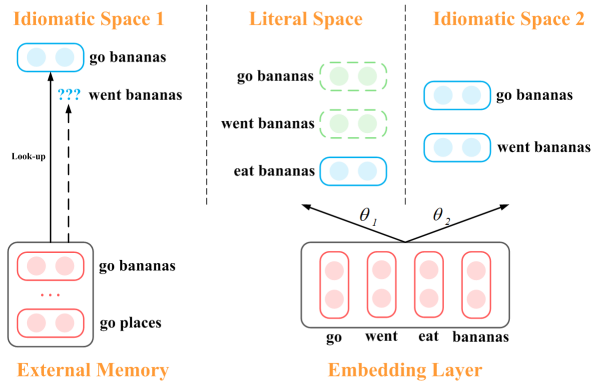


Figure 2: Illustration of the compositional encoder (in literal space) and two different non-compositional encoders (in idiomatic spaces). θ_1 and θ_2 represent different compositional functions.

where k denotes the index of the corresponding phrase.

Retrieval model provides an efficient way to generate idiomatic meanings for non-compositional phrases, which can benefit from the knowledge of external data. Apparently, one limitation of this model is that RM requires a large matrix and suffers from the problem of OOVs. To address this problem, we propose another representing mechanism.

Compositional Model (CM) Nunberg *et al.* [1994] proposes a view that idiomatic phrases can still be composed in a new space. Inspired by it, we compose non-compositional phrases with a new compositional function.

$$\mathbf{h}_j^n = \tanh\left(\mathbf{W}\begin{bmatrix}\mathbf{h}_j^l \\ \mathbf{h}_j^r\end{bmatrix} + \mathbf{b}\right), \tag{6}$$

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

Model Comparison For the compositional encoder and non-compositional encoder, as shown in Figure 2, they both share an embedding layer, while have different parameters to learn.

For two different non-compositional encoders, CM generates the representation of non-compositional phrases online, saving a lot parameters therefore avoiding the problem of overfitting. Besides, in this compositional manner, the model can handle a lot of variation in terms of morphology, lexicon, and syntax. For example, RM can model the idiom "`go bananas`" while "`went bananas`" is an unknown phrase for RM. By contrast, the case is more simple for CM since it knows "`went`" and "`go`" are similar (CM shares embedding layer with compositional encoder.).

4.3 Semantic Compositional Switch

A major element of our model is the introduction of a parameterized semantic compositional switch, which outputs a scalar α to determine whether a phrase is compositional or non-compostional. When the compositional switch is turned on at one node of a tree (i.e., when α equals 1), the model considers this phrase as non-compositional signifying that the meaning of the phrase is non-literal. Therefore, at each node of the tree, the model will compute the value of the switch. Here, we use a single-layer multilayer perceptron to compute α:

$$\alpha_j = \sigma(v_s^T \tanh(\mathbf{W}_s[\mathbf{h}_j^l, \mathbf{h}_j^r, \mathbf{h}_j^n])) \tag{7}$$

where $\mathbf{W}_s \in \mathbb{R}^{d\times 3d}$, $v_s \in \mathbb{R}^d$ are the weight matrices.

Finally, the hidden state of h_j can be obtained as follows:

$$\mathbf{h}_j = \alpha_j \mathbf{h}_j^n + (1-\alpha_j)\mathbf{h}_j^c \tag{8}$$

4.4 Context-dependent Phrase Composition

We consider in some cases the same phrase can act as both compositional and non-compositional in different contexts. For example, the phrase "`around the corner`" in the sentence "`My brother played with friends around the corner`" is compositional while in the sentence "`The Spring Festival is just around the corner`" is non-compositional.

In order to enable our model to disambiguate the meanings of phrases, we can easily extend above proposed model to achieve context-dependent phrase composition. Inspired by the work tracking LSTM proposed by Bowman *et al.* [2016], we choose a simple way with a little modification of Eq. 1. Specifically, we replace the word vector $\mathbf{x}$ with hidden state $\mathbf{s}$, which is a hidden state emitted by a sequence-based LSTM.

$$\mathbf{s}_t = \textbf{LSTM-Seq}(\mathbf{s}_{t-1}, \mathbf{c}_{t-1}, \mathbf{x}_t) \tag{9}$$

where $t = 1, 2, \cdots, n$ and n represents the length of the input sequence.

5 Experiment

5.1 Training

To evaluation our models, we choose the task of sentiment classification and use five datasets. More formally, given a sentence X and its label l. The output $\hat{l}$ of neural network is the probabilities of the different classes. The parameters of the network are trained to minimise the cross-entropy of the predicted and true label distributions.

$$L(X; \boldsymbol{l}, \hat{\boldsymbol{l}}) = -\sum_{j=1}^{C} \boldsymbol{l}_j \log(\hat{\boldsymbol{l}}_j), \tag{10}$$

where $\boldsymbol{l}$ is one-hot representation of the ground-truth label l; $\hat{\boldsymbol{l}}$ is predicted probabilities of labels; C is the class number.

To minimize the objective, we use stochastic gradient descent with the diagonal variant of AdaGrad [Duchi *et al.*, 2011].

5.2 Competitor Methods

To make a comprehensive comparison, we choose several typical and powerful models as our baselines. The settings of some major competitor methods and our models are listed as follows:

- BiLSTM: Bi-directional LSTM.

- T-LSTM: LSTM over tree structure.
- HT-LSTM: A hybrid of tree and sequence LSTM, with the ability to learn context-dependent phrase composition.
- AdaHT-LSTM-RM: HT-LSTM with an adaptive compositional switch and RM non-compositional encoder.
- AdaHT-LSTM-CM: HT-LSTM with an adaptive compositional switch and CM non-compositional encoder.

Initialization and Hyperparameters In all of our experiments, the word and phrase embeddings are trained using word2vec [Mikolov *et al.*, 2013] on the Wikipedia corpus (1B words). The other parameters are initialized by randomly sampling from uniform distribution in $[-0.1, 0.1]$.

Generally, the longer the phrases are, the lower likelihood it is non-compositional. To reduce the computational cost, we only provide the additional option of non-compositional representation to the phrase containing not more than L words. L is a hyper-parameter, and is tuned on the development set.

For each task, we take the hyperparameters which achieve the best performance on the development set via an small grid search over combinations of the initial learning rate $[0.1, 0.01, 0.001]$, l_2 regularization $[0.0, 5E{-}5, 1E{-}5]$ and the values of L $[3, 4, 5, 6]$. The final hyper-parameters are as follows. For the idiom-enriched dataset, the size of pre-trained embedding is 300 while the size of hidden state is 200. The initial learning rate is 0.01. The regularization weight of the parameters is 10^{-5}. For the other datasets, the sizes of pre-trained embeddings and hidden states are both set as 300. The initial learning rate is 0.1. The regularization weight of the parameters is 0. Besides, the value of L is set to 4 for all datasets.

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

5.3 Experiment I: Idiom-enriched Sentiment Classification

To test the ability to model non-compositional phrases of our models, we carefully select an idiom-enriched sentiment classification dataset. Specifically, we choose this dataset due to the following reasons:

- Most of idioms are non-compositional phrases.
- Idioms typically imply an affective stance toward something rather than a neutral one. [Williams *et al.*, 2015]
- the error analysis of sentiment classification results reveals that a large number of errors occur when idioms are used to express sentiment [Balahur *et al.*, 2013].

We use the dataset proposed by [Williams *et al.*, 2015]. In their work, they first collected a set of 580 idioms that are relevant to sentiment analysis, and then they assembled a corpus of 2521 sentences that contain an expression which can be matched to an idiom. In most cases, this expression will have a figurative meaning associated with an idiom, but in some cases it will convey a literal meaning. Moreover, each sentence was annotated with a sentiment label "`positive`","`negative`" or "`other`" to create a gold standard .

Sentences	**Labels**
I can tell you, he's got a very short fuse.	Negative
She was in seventh heaven.	Positive
None of us can sit on the fence.	Other

Table 1: Examples in idiom-enriched sentiment classification dataset.

Phrase	*Score*	*Sentence*
wooden spoon	*0.21*	*He used a **wooden spoon** to stir the mixture, whistling softly.*
	0.95	*Boys finished with the **wooden spoon** after losing a penalty shoot out 5-4.*
in the bag	*0.32*	*I looked **in the bag**, it was full of fish.*
	0.83	*Once we'd scored the third goal, the match was pretty much **in the bag**.*

Table 2: The switch of our model outputs different non-composition scores towards the same phrases, which both have literal and non-literal meaning under different context.

To better understand this task, we give some examples of this dataset as shown in Table 1.

Results Table 3 shows the evaluation results on idiom-enriched sentiment classification. The classification performance was evaluated in terms of three measures: precision (Prec.), recall (Rec.) and F-measure (F). All the measures are computed by averaging metrics of each class and weighted by the number of true instances for each class.

From experimental results, we have several findings.

- AdaHT-LSTM consistently outperforms all the baseline with a significant margin and CM encoder shows a better performance than RM. We attribute the success of RM to its power in modeling variations of non-compositional phrases.
- Both sequenced-based and tree-based models perform better than NBOW, which indicates the importance of word orders. Additionally, T-LSTM outperforms LSTM with just a small margin.
- Compared with T-LSTM, HT-LSTM achieves a better performance, which shows the effectiveness of introducing context-dependent phrase composition.

Non-compositionality In order to understand where the performance improvement comes from, we analyze some cases and design an experiment to compare the output of HT-LSTM and adptive HT-LSTM at each node of their corresponding trees.

Model	Prec.	Rec.	F
NBOW	50.4	52.0	46.6
LSTM	54.6	55.0	54.8
BiLSTM	55.1	57.0	54.5
T-LSTM	54.9	56.0	55.2
HT-LSTM	58.1	55.6	56.0
AdaHT-LSTM-RM	60.9	59.6	59.3
AdaHT-LSTM-CM	**61.3**	**62.5**	**62.1**

Table 3: Evaluation results of idiom-enriched sentiment classification.

Dataset	Train	Dev.	Test	Class	Avg-L	Voc.
MR	9596	-	1066	2	22	21K
SST-1	8544	1101	2210	5	19	18K
SST-2	6920	872	1821	2	18	15K
SUBJ	9000	-	1000	2	21	21K

Table 4: Statistics of the four datasets used in this paper. Avg-L denotes the average length while Voc. represents the size of vocabulary.

More specifically, we randomly sample one sentence[1] from the development set, and the dynamical changes of the predicted sentiment scores over trees are shown in Figure 3.

The sentence "`His performance is at fever pitch`" has a **positive** sentiment. T-LSTM ignores the idiom "`at fever pitch`" and thereby consider the label of the sentence is "**Other**". By contrast, adaptive HT-LSTM captures this informative pattern and regards the phrase "`at fever pitch`" as non-compositional therefore making a correct prediction.

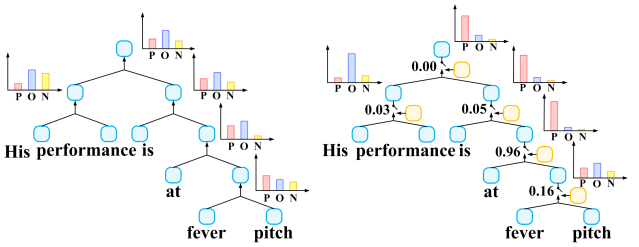
Disambiguation With the help of context-dependent LSTM, we find our model can perform disambiguation of those phrases which both have literal and non-literal meanings. As shown in Table 2, the phrase "`wooden spoon`" can both refer to "`a kinds of spoon `" and "`the prize to be given to an individual or team which has come last in a competition`". Based on the different contexts, our model gives different compositional scores towards these two phrases thereby revealing its senses which are considered as indicators of sentiment. Another example also illustrates this case.

5.4 Experiment II: Large Datasets for Sentiment Classification

To make an extensive evaluation, we also choose several movie review datasets for sentiment classification, which not only have more training data, but contain many sentiment-oriented idioms [Williams *et al.*, 2015]. The detailed statistics about these four datasets are listed in Table 4. Each dataset is briefly described as follows.

- **MR** The movie reviews with two classes [Pang and Lee, 2005].

[1]The pages of the paper is so limited that we can not give more examples.



(a) T-LSTM (b) Ours

Figure 3: The change of the predicted sentiment score at different nodes of the tree. The histograms show the probability distribution of corresponding nodes over the three class **Positive** (P), **Other** (O), **Negative** (N).

- **SST-1** The movie reviews with five classes (negative, somewhat negative, neutral, somewhat positive, positive) in the Stanford Sentiment Treebank[2] [Socher *et al.*, 2013].
- **SST-2** The movie reviews with binary classes. It is also from the Stanford Sentiment Treebank.
- **SUBJ** Subjectivity data set where the goal is to classify each instance (snippet) as being subjective or objective. [Pang and Lee, 2004]

Results Table 6 shows the classification accuracies of our proposed models compared with the competitor models. AdaHT-LSTM-CM consistently outperforms RAE, MV-RNN, RTNT, T-LSTM and HT-LSTM by a large margin while achieving comparable results to the state-of-the-art. With the help of CM encoder, the performances on all four datasets are improved, indicating the effectiveness of it. Moreover, HT-LSTM performs better than T-LSTM, which shows the effectiveness of context-dependent phrase composition again.

Analysis of Non-compositional Phrases To get a better intuition of the behaviour of compositional switch, we examined the non-compositional scores of each phrases and look at what kinds of phrase can be assigned a higher score by our model. Specifically, we selected all the phrases whose non-compositional score are larger than 0.9 from all the five development sets. Surprisingly, we observe that most of those phrases with higher non-compositional scores are indeed non-compositional. Besides those phrases can be roughly summed up into six types: Proper Noun, Light-verb Phrases, Phrasal Verbs, Noun Phrases, Adverb Phrase and Idioms. We randomly sample some phrases and list all six categories as shown in Table 5.

We can see the model can not only pick up the proper noun such as: the name of movie "`Star Wars saga`" and the name of people "`Barry Skolnick`", but can identify the phrases with metaphorical and expanded meanings such as "`short cuts`" and "`weak at the knee`". Most of

[2]`http://nlp.stanford.edu/sentiment`.

Proper Noun	Light-verb Phrases	Phrasal Verbs	Noun Phrases	Adverb Phrase	Idioms
3 D	*take cover*	*thumbs down*	*short cuts*	*at least*	*go bananas*
Holly Bolly	*taking place*	*go on*	*deja vu*	*at once*	*come to blows*
star wars saga	*take the bull*	*rips off*	*black comedy*	*all in all*	*go to the dogs*
Barry Skolnick	*playing out*	*fly over*	*femme fatale*	*toward the end*	*weak at the knees*
Apollo 13	*make any money*	*fall apart*	*cuts corners*	*not the least*	*would come in handy*

Table 5: Different types of phrases are listed, whose non-composition scores are higher than 0.9. The Proper Noun includes the names of people, places, movies ect. The "Light-verb Phrases" denotes the phrases constructed by "Light-verb", such as "have", "make", "take".

Model	MR	SST-1	SST-2	SUBJ
NBOW	77.2	42.4	80.5	91.3
RAE	77.7	43.2	82.4	-
MV-RNN	79.0	44.4	82.9	-
RNTN	-	45.7	85.4	-
DCNN	-	48.5	86.8	-
CNN-multichannel	81.5	47.4	**88.1**	93.2
T-LSTM	78.7	48.5	86.1	91.0
HT-LSTM	79.5	48.9	86.9	91.7
AdaHT-LSTM-RM	81.7	49.8	87.3	93.8
AdaHT-LSTM-CM	**81.9**	50.2	87.8	**94.1**

Table 6: Accuracies of our models on four datasets against state-of-the-art neural models. **NBOW**: Sums up the word vectors and applies a non-linearity followed by a softmax classification layer. **RAE**: Recursive Autoencoders with pre-trained word vectors from Wikipedia [Socher *et al.*, 2011]. **MV-RNN**: Matrix-Vector Recursive Neural Network with parse trees [Socher *et al.*, 2013]. **RNTN**: Recursive Neural Tensor Network with tensor-based feature function and parse trees [Socher *et al.*, 2013]. **DCNN**: Dynamic Convolutional Neural Network with dynamic k-max pooling [Kalchbrenner *et al.*, 2014; Denil *et al.*, 2014]. **CNN-multichannel**: Convolutional Neural Network [Kim, 2014].

these phrases either imply an affective stance toward something: such as "`go bananas`", "`thumbs down`", or are critical to the understanding of sentences such as the "Light-verb Phrases" and "Phrasal Verbs".

6 Related Work

There have been many studies, which focused on exploring the compositionality of various types of phrases. Kartsaklis *et al.* [2012] have discussed the advantages and disadvantages of using compositional or non-compositional embeddings. More recently, Hashimoto and Tsuruoka [2016] proposed to use compositionality scores to adaptively learn the phrase embeddings. Different with these models, we focus on sentence modelling and integrate compositional switch structure into neural sentence models, which can naturally encode the non-compositional phrases thereby obtaining more desirable sentence vectors. Furthermore, compared with above models, the composition of phrases in our models are context-sensitive and therefore can disambiguates the meanings of phrases.

Another thread of work is sentence modelling with various kinds of composition functions. Socher *et al.* [2010] proposed a context-sensitive recursive neural networks, which can model the context-dependent compositonality of phrases. Dong *et al.* [2014] proposed an adaptive recursive neural network, which can model the adaptive sentiment propagations as distributions over multiple composition functions. Tai *et al.* [2015] proposed tree-based LSTM, which builds on recursive neural networks. However, these models ignore the non-compositionality of phrases when modelling sentence. By contrast, we grow the capacity of syntax-based neural sentences model, allowing it can model the non-compositional phrases when learning sentence representation.

7 Conclusion

In this paper, we introduce a context-dependent controller in recursively neural sentence modelling, which adaptively determines whether the representation of a phrase should be composed of its containing words or subphrases. Experiments on five sentiment classification datasets demonstrate the efficacy of our proposed model and its superiority to competitor models. Furthermore, we have made an elaborate experiment design and case analysis to evaluate the effectiveness of our proposed models.

The proposed model approaches one step towards understanding collocation (such as idioms) in a sentence. In future, we want to investigate how to jointly identify and understand the collocations appeared in the sentences.

Acknowledgments

We would like to thank the anonymous reviewers for their valuable comments and thank Kaiyu Qian 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

[Balahur *et al.*, 2013] Alexandra Balahur, Ralf Steinberger, Mijail Kabadjov, Vanni Zavarella, Erik Van Der Goot, Matina Halkia, Bruno Pouliquen, and Jenya Belyaeva. Sentiment analysis in the news. *arXiv preprint arXiv:1309.6202*, 2013.

[Bowman *et al.*, 2016] Samuel R Bowman, Jon Gauthier, Abhinav Rastogi, Raghav Gupta, Christopher D Manning, and Christopher Potts. A fast unified model for parsing and sentence understanding. *arXiv preprint arXiv:1603.06021*, 2016.

[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.

[Denil *et al.*, 2014] Misha Denil, Alban Demiraj, Nal Kalchbrenner, Phil Blunsom, and Nando de Freitas. Modelling, visualising and summarising documents with a single convolutional neural network. *arXiv preprint arXiv:1406.3830*, 2014.

[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.

[Hashimoto and Tsuruoka, 2016] Kazuma Hashimoto and Yoshimasa Tsuruoka. Adaptive joint learning of compositional and non-compositional phrase embeddings. *arXiv preprint arXiv:1603.06067*, 2016.

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

[Hu *et al.*, 2014] Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. Convolutional neural network architectures for matching natural language sentences. In *Advances in Neural Information Processing Systems*, 2014.

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

[Kartsaklis *et al.*, 2012] Dimitri Kartsaklis, Mehrnoosh Sadrzadeh, and Stephen Pulman. A unified sentence space for categorical distributional-compositional semantics: Theory and experiments. In *In Proceedings of COLING: Posters*. Citeseer, 2012.

[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 Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 423–430, 2003.

[Liu *et al.*, 2015] PengFei Liu, Xipeng Qiu, Xinchi Chen, Shiyu Wu, and Xuanjing Huang. Multi-timescale long short-term memory neural network for modelling sentences and documents. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2015.

[Liu *et al.*, 2016a] PengFei Liu, Xipeng Qiu, and Xuanjing Huang. Recurrent neural network for text classification with multi-task learning. In *Proceedings of International Joint Conference on Artificial Intelligence*, 2016.

[Liu *et al.*, 2016b] PengFei Liu, Xipeng Qiu, and Xuanjing Huang. Syntax-based attention model for natural language inference. *arXiv preprint arXiv:1607.06556*, 2016.

[Liu *et al.*, 2017] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. Dynamic compositional neural networks over tree structure. *arXiv preprint arXiv:1705.04153*, 2017.

[Mikolov *et al.*, 2010] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *INTERSPEECH*, 2010.

[Mikolov *et al.*, 2013] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[Nunberg *et al.*, 1994] Geoffrey Nunberg, Ivan A Sag, and Thomas Wasow. Idioms. *Language*, pages 491–538, 1994.

[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 43rd annual meeting on association for computational linguistics*, pages 115–124. Association for Computational Linguistics, 2005.

[Socher *et al.*, 2010] Richard Socher, Christopher D Manning, and Andrew Y Ng. Learning continuous phrase representations and syntactic parsing with recursive neural networks. In *Proceedings of the NIPS-2010 Deep Learning and Unsupervised Feature Learning Workshop*, pages 1–9, 2010.

[Socher *et al.*, 2011] Richard Socher, Jeffrey Pennington, Eric H Huang, Andrew Y Ng, and Christopher D Manning. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of EMNLP*, 2011.

[Socher *et al.*, 2013] 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.

[Sutskever *et al.*, 2014] Ilya Sutskever, Oriol Vinyals, and Quoc VV Le. Sequence to sequence learning with neural networks. In *Advances in NIPS*, pages 3104–3112, 2014.

[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.