

A Deep Network with Visual Text Composition Behavior

Hongyu Guo

National Research Council Canada
1200 Montreal Road, Ottawa, Ontario, K1A 0R6
hongyu.guo@nrc-cnrc.gc.ca

Abstract

While natural languages are compositional, how state-of-the-art neural models achieve compositionality is still unclear. We propose a deep network, which not only achieves competitive accuracy for text classification, but also exhibits compositional behavior. That is, while creating hierarchical representations of a piece of text, such as a sentence, the lower layers of the network distribute their layer-specific attention weights to individual words. In contrast, the higher layers compose meaningful phrases and clauses, whose lengths increase as the networks get deeper until fully composing the sentence.

1 Introduction

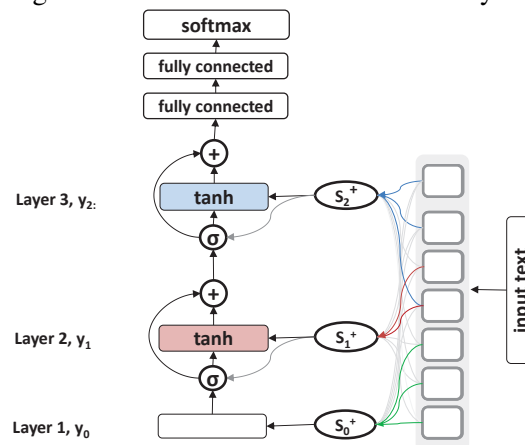
Deep neural networks leverage task-specific architectures to develop hierarchical representations of the input, where higher level representations are derived from lower level features (Conneau et al., 2016). Such hierarchical representations have visually demonstrated compositionality in image processing, i.e., pixels combine to form shapes and then contours (Farabet et al., 2013; Zeiler and Fergus, 2014). Natural languages are also compositional, i.e., words combine to form phrases and then sentences. Yet unlike in vision, how deep neural models in NLP, which mainly operate on distributed word embeddings, achieve compositionality, is still unclear (Li et al., 2015, 2016).

We propose an Attention Gated Transformation (AGT) network, where each layer's feature generation is gated by a layer-specific attention mechanism (Bahdanau et al., 2014). Specifically, through distributing its attention to the original given text, each layer of the networks tends to *in-*

crementally retrieve new words and phrases from the original text. The new knowledge is then combined with the previous layer's features to create the current layer's representation, thus resulting in composing longer or new phrases and clauses while creating higher layers' representations of the text.

Experiments on the Stanford Sentiment Treebank (Socher et al., 2013) dataset show that the AGT method not only achieves very competitive accuracy, but also exhibits compositional behavior via its layer-specific attention. We empirically show that, given a piece of text, e.g., a sentence, the lower layers of the networks select individual words, e.g., negative and conjunction words *not* and *though*, while the higher layers aim at composing meaningful phrases and clauses such as negation phrase *not so much*, where the phrase length increases as the networks get deeper until fully composing the whole sentence. Interestingly, after composing the sentence, the compositions of different sentence phrases compete to become the dominating features of the end task.

Figure 1: An AGT network with three layers.



2 Attention Gated Transformation Network

Our AGT network was inspired by the Highway Networks (Srivastava et al., 2015a,b), where each layer is equipped with a *transform gate*.

2.1 Transform Gate for Information Flow

Consider a feedforward neural network with multiple layers. Each layer l typically applies a non-linear transformation f (e.g., \tanh , parameterized by W_l^f), on its input, which is the output of the most recent previous layer (i.e., y_{l-1}), to produce its output y_l . Here, $l = 0$ indicates the first layer and y_0 is equal to the given input text x , namely $y_0 = x$:

$$y_l = f(y_{l-1}, W_l^f) \quad (1)$$

While in a highway network (the left column of Figure 1), an additional non-linear transform gate function T_l is added to the l^{th} ($l > 0$) layer:

$$y_l = f(y_{l-1}, W_l^f)T_l + y_{l-1}(1 - T_l) \quad (2)$$

where the function T_l expresses how much of the representation y_l is produced by transforming the y_{l-1} (first term in Equation 2), and how much is just carrying from y_{l-1} (second term in Equation 2). Here T_l is typically defined as:

$$T_l = \sigma(W_l^t y_{l-1} + b_l^t) \quad (3)$$

where W_l^t is the weight matrix and b_l^t the bias vector; σ is the non-linear activation function.

With transform gate T , the networks learn to decide if a feature transformation is needed at each layer. Suppose σ represents a sigmoid function. In such case, the output of T lies between zero and one. Consequently, when the transform gate is one, the networks pass through the transformation f over y_{l-1} and block the pass of input y_{l-1} ; when the gate is zero, the networks pass through the unmodified y_{l-1} , while the transformation f over y_{l-1} is suppressed.

The left column of Figure 1 reflects the highway networks as proposed by (Srivastava et al., 2015b). Our AGT method adds the right two columns of Figure 1. That is, 1) the transform gate T_l now is not a function of y_{l-1} , but a function of the selection vector s_l^+ , which is determined by the attention distributed to the given input x by the l^{th} layer (will be discussed next), and 2) the function f takes as input the concatenation of y_{l-1} and s_l^+

to create feature representation y_l . These changes result in an attention gated transformation when forming hierarchical representations of the text.

2.2 Attention Gated Transformation

In AGT, the activation of the transform gate at each layer depends on a layer-specific attention mechanism. Formally, given a piece of text x , such as a sentence with N words, it can be represented as a matrix $B \in \mathbb{R}^{N \times d}$. Each row of the matrix corresponds to one word, which is represented by a d -dimensional vector as provided by a learned word embedding table. Consequently, the selection vector s_l^+ , for the l^{th} layer, is the softmax weighted sum over the N word vectors in B :

$$s_l^+ = \sum_{n=1}^N d_{l,n} B[n : n] \quad (4)$$

with the weight (i.e., attention) $d_{l,n}$ computed as:

$$d_{l,n} = \frac{\exp(m_{l,n})}{\sum_{n=1}^N \exp(m_{l,n})} \quad (5)$$

$$m_{l,n} = w_l^m \tanh(W_l^m (B[n : n])) \quad (6)$$

here, w_l^m and W_l^m are the weight vector and weight matrix, respectively. By varying the attention weight $d_{l,n}$, the s_l^+ can focus on different rows of the matrix B , namely different words of the given text x , as illustrated by different color curves connecting to s^+ in Figure 1. Intuitively, one can consider s^+ as a learned word selection component: choosing different sets of words of the given text x by distributing its distinct attention.

Having built one s^+ for each layer from the given text x , the activation of the transform gate for layer l ($l > 0$) (i.e., Equation 3) is calculated:

$$T_l = \sigma(W_l^t s_l^+ + b_l^t) \quad (7)$$

To generate feature representation y_l , the function f takes as input the concatenation of y_{l-1} and s_l^+ . That is, Equation 2 becomes:

$$y_l = \begin{cases} s_l^+, & l = 0 \\ f([y_{l-1}; s_l^+], W_l^f)T_l + y_{l-1}(1 - T_l), & l > 0 \end{cases} \quad (8)$$

where $[...;...]$ denotes concatenation. Thus, at each layer l , the gate T_l can regulate either passing through y_{l-1} to form y_l , or retrieving novel

knowledge from the input text x to augment y_{l-1} to create a better representation for y_l .

Finally, as depicted in Figure 1, the feature representation of the last layer of the AGT is fed into two fully connected layers followed by a softmax function to produce a distribution over the possible target classes. For training, we use multi-class cross entropy loss.

Note that, Equation 8 indicates that the representation y_l depends on both s_l^+ and y_{l-1} . In other words, although Equation 7 states that the gate activation at layer l is computed by s_l^+ , the gate activation is also affected by y_{l-1} , which embeds the information from the layers below l .

Intuitively, the AGT networks are encouraged to consider new words/phrases from the input text at higher layers. Consider the fact that the s_0^+ at the bottom layer of the AGT only deploys a linear transformation of the bag-of-words features. If no new words are used at higher layers of the networks, it will be challenge for the AGT to sufficiently explore different combinations of word sets of the given text, which may be important for building an accurate classifier. In contrast, through tailoring its attention for new words at different layers, the AGT enables the words selected by a layer to be effectively combined with words/phrases selected by its previous layers to benefit the accuracy of the classification task (more discussions are presented in Section 3.2).

3 Experimental Studies

3.1 Main Results

The Stanford Sentiment Treebank data contains 11,855 movie reviews (Socher et al., 2013). We use the same splits for training, dev, and test data as in (Kim, 2014) to predict the fine-grained 5-class sentiment categories of the sentences. For comparison purposes, following (Kim, 2014; Kalchbrenner et al., 2014; Lei et al., 2015), we trained the models using both phrases and sentences, but only evaluate sentences at test time. Also, we initialized all of the word embeddings (Cherry and Guo, 2015; Chen and Guo, 2015) using the 300 dimensional pre-trained vectors from GloVe (Pennington et al., 2014). We learned 15 layers with 200 dimensions each, which requires us to project the 300 dimensional word vectors; we implemented this using a linear transformation, whose weight matrix and bias term are shared across all words, followed by

a tanh activation. For optimization, we used Adadelta (Zeiler, 2012), with learning rate of 0.0005, mini-batch of 50, transform gate bias of 1, and dropout (Srivastava et al., 2014) rate of 0.2. All these hyperparameters were determined through experiments on the validation-set.

AGT	50.5
high-order CNN	51.2
tree-LSTM	51.0
DRNN	49.8
PVEC	48.7
DCNN	48.5
DAN	48.2
CNN-MC	47.4
CNN	47.2
RNTN	45.7
NBoW	44.5
RNN	43.2
SVM	38.3

Table 1: Test-set accuracies obtained; results except the AGT are drawn from (Lei et al., 2015).

Figure 2: Soft attention distribution (top) and phrase length distribution (bottom) on the test set.

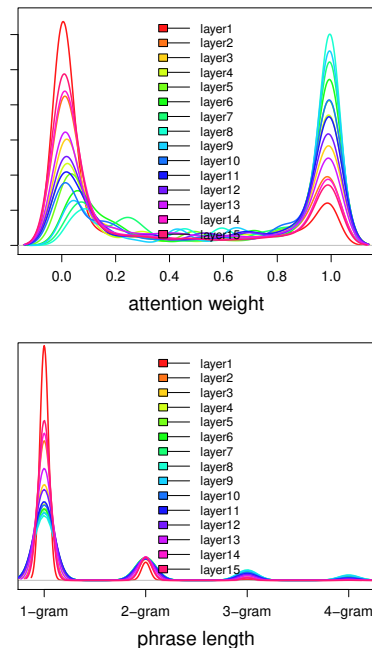
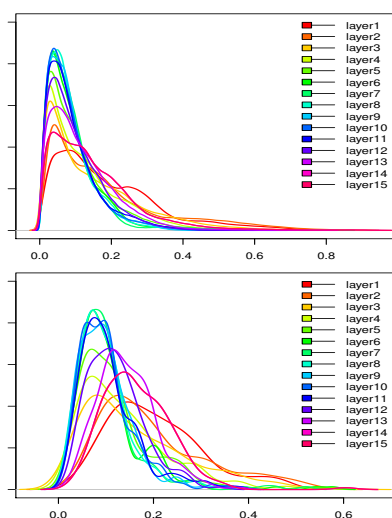


Table 1 presents the test-set accuracies obtained by different strategies. Results in Table 1 indicate that the AGT method achieved very competitive accuracy (with 50.5%), when compared to the state-of-the-art results obtained by the tree-LSTM (51.0%) (Tai et al., 2015; Zhu et al., 2015) and high-order CNN approaches (51.2%) (Lei et al., 2015).

Top subfigure in Figure 2 depicts the distribu-

Figure 3: Transform gate activities of the test-set (top) and the first sentence in Figure 4 (bottom).



tions of the attention weights created by different layers on all test data, where the attention weights of all words in a sentence, i.e., $d_{l,n}$ in Equation 4, are normalized to the range between 0 and 1 within the sentence. The figure indicates that AGT generated very spiky attention distribution. That is, most of the attention weights are either 1 or 0. Based on these narrow, peaked bell curves formed by the normal distributions for the attention weights of 1 and 0, we here consider a word has been *selected* by the networks if its attention weight is larger than 0.95, i.e., receiving more than 95% of the full attention, and a phrase has been *composed and selected* if a set of consecutive words all have been selected.

In the bottom subfigure of Figure 2 we present the distribution of the phrase lengths on the test set. This figure indicates that the middle layers of the networks e.g., 8th and 9th, have longer phrases (green and blue curves) than others, while the layers at the two ends contain shorter phrases (red and pink curves).

In Figure 3, we also presented the transform gate activities on all test sentences (top) and that of the first example sentence in Figure 4 (bottom). These curves suggest that the transform gates at the middle layers (green and blue curves) tended to be close to zero, indicating the pass-through of lower layers' representations. On the contrary, the gates at the two ends (red and pink curves) tended to be away from zero with large tails, implying the retrieval of new knowledge from the input text. These are consistent with the results below.

Figure 4 presents three sentences with various lengths from the test set, with the attention weights numbered and then highlighted in heat map. Figure 4 suggests that the lower layers of the networks selected individual words, while the higher layers aimed at phrases. For example, the first and second layers seem to *select* individual words carrying strong sentiment (e.g., *predictable*, *bad*, *never* and *delicate*), and conjunction and negation words (e.g., *though* and *not*). Also, meaningful phrases were *composed and selected* by later layers, such as *not so much*, *not only... but also*, *bad taste*, *bad luck*, *emotional development*, and *big screen*. In addition, in the middle layer, i.e., the 8th layer, the whole sentences were composed by filtering out uninformative words, resulting in concise versions, as follows (selected words and phrases are highlighted in color blocks).

- 1) *though* *plot* *predictable* *movie* *never* *feels* *formulaic* *attention* *nuances* *emotional development* *delicate* *characters*
- 2) *bad* *company* *leaves* *bad* *taste* *not* *only* *bad* *luck* *but also* *staleness* *script*
- 3) *not so much* *movie* *picture* *big* *screen*

Interestingly, if relaxing the word selection criteria, e.g., including words receiving more than the median, rather than 95%, of the full attention, the sentences recruited more conjunction and modification words, e.g., *because*, *for*, *a*, *its* and *on*, thus becoming more readable and fluent:

- 1) *though* *plot* *is* *predictable* *movie* *never* *feels* *formulaic* *because* *attention* *is* *on* *nuances* *emotional development* *delicate* *characters*
- 2) *bad* *company* *leaves* *a* *bad* *taste* *not* *only* *because* *its* *bad* *luck* *timing* *but also* *staleness* *its* *script*
- 3) *not so much* *a* *movie* *a* *picture* *book* *for* *big* *screen*

Now, consider the AGT's compositional behavior for a specific sentence, e.g., the last sentence in Figure 4. The first layer solely *selected* the word *not* (with attention weight of 1 and all other words with weights close to 0), but the 2nd to 4th layers gradually pulled out new words *book*, *screen* and *movie* from the given text. Incrementally, the 5th and 6th layers further *selected* words to form phrases *not so much*, *picture book*, and *big screen*. Finally, the 7th and 8th layers added some

Figure 4: Three sentences from the test set and their attention received from the 15 layers (L1 to L15).

L1	though	the	plot	is	predicta	the	movie	never	feels	formula	,	because	the	attentio	is	on	the	nuance	of	the	emotion	developr	of	the	delicati	charact	
L2	0.84	0.00	0.06	0.00	1.00	0.00	0.00	0.97	0.29	0.93	0.00	0.00	0.00	0.08	0.00	0.00	0.00	0.63	0.00	0.00	0.21	0.00	0.00	0.00	1.00	0.01	
L3	0.97	0.00	0.33	0.01	1.00	0.00	0.00	1.00	0.78	0.99	0.00	0.00	0.00	0.38	0.01	0.00	0.00	0.90	0.00	0.00	0.79	0.02	0.00	0.00	1.00	0.03	
L4	1.00	0.00	0.79	0.03	1.00	0.01	0.00	0.15	1.00	0.97	1.00	0.01	0.05	0.86	0.03	0.01	0.00	0.99	0.00	0.00	0.98	0.15	0.00	0.00	1.00	0.16	
L5	1.00	0.01	0.98	0.29	1.00	0.00	0.01	0.98	1.00	1.00	1.00	0.00	0.27	0.01	0.99	0.29	0.08	0.01	1.00	0.02	0.01	1.00	0.53	0.02	0.01	1.00	0.74
L6	1.00	0.03	0.99	0.49	1.00	0.01	0.03	0.99	1.00	1.00	1.00	0.01	0.44	0.03	0.99	0.49	0.16	0.03	1.00	0.04	0.03	1.00	0.70	0.04	0.03	1.00	0.84
L7	1.00	0.06	0.99	0.71	1.00	0.02	0.06	1.00	1.00	1.00	1.00	0.02	0.66	0.06	1.00	0.71	0.32	0.06	1.00	0.07	0.06	1.00	0.84	0.07	0.06	1.00	0.92
L8	1.00	0.08	1.00	0.83	1.00	0.03	0.08	1.00	1.00	1.00	1.00	0.03	0.76	0.08	1.00	0.83	0.44	0.08	1.00	0.10	0.08	1.00	0.90	0.10	0.08	1.00	0.95
L9	1.00	0.10	1.00	0.94	1.00	0.04	0.10	1.00	1.00	1.00	1.00	0.04	0.85	0.10	1.00	0.94	0.60	0.10	1.00	0.12	0.10	1.00	0.95	0.12	0.10	1.00	0.98
L10	1.00	0.05	1.00	0.94	1.00	0.02	0.05	1.00	1.00	1.00	1.00	0.02	0.79	0.05	1.00	0.94	0.49	0.05	1.00	0.07	0.05	1.00	0.92	0.07	0.05	1.00	0.98
L11	1.00	0.00	0.99	0.81	1.00	0.00	0.00	1.00	1.00	1.00	1.00	0.00	0.00	0.00	1.00	0.81	0.15	0.00	1.00	0.01	0.00	1.00	0.75	0.01	0.00	1.00	0.93
L12	1.00	0.00	0.99	0.68	1.00	0.00	0.00	1.00	1.00	1.00	1.00	0.00	0.24	0.00	1.00	0.68	0.07	0.00	1.00	0.01	0.00	1.00	0.59	0.01	0.00	1.00	0.88
L13	0.99	0.00	0.98	0.51	1.00	0.01	0.00	0.99	1.00	1.00	1.00	0.01	0.08	0.00	0.99	0.51	0.03	0.00	0.99	0.00	0.00	1.00	0.42	0.00	0.00	1.00	0.81
L14	0.92	0.00	0.65	0.04	1.00	0.00	0.00	0.62	1.00	0.98	1.00	0.01	0.02	0.00	0.98	0.17	0.02	0.00	0.99	0.00	0.00	1.00	0.18	0.00	0.00	1.00	0.43
L15	0.80	0.00	0.60	0.03	1.00	0.00	0.00	0.02	0.99	0.91	0.99	0.00	0.00	0.00	0.88	0.04	0.00	0.00	0.96	0.00	0.00	0.99	0.05	0.00	0.00	1.00	0.14
L1	bad	compar	leaves	a	bad	taste,	not	only	because	of	its	bad	luck	timing	,	but	also	the	stalem	of	its	script					
L2	0.95	0.02	0.25	0.00	0.95	0.05	0.00	0.92	0.05	0.00	0.00	0.00	0.95	0.90	0.05	0.00	0.60	0.00	0.00	1.00	0.00	0.00	0.00	0.01			
L3	1.00	0.16	0.85	0.00	1.00	0.46	0.01	1.00	0.69	0.01	0.01	0.00	1.00	0.99	0.18	0.01	0.93	0.02	0.01	1.00	0.01	0.01	0.00	0.00	0.05		
L4	1.00	0.70	0.99	0.00	1.00	0.93	0.02	1.00	0.98	0.07	0.02	0.01	1.00	1.00	0.47	0.02	1.00	0.19	0.02	1.00	0.02	0.01	0.00	0.01	0.49		
L5	1.00	0.75	1.00	0.00	1.00	0.99	0.09	1.00	1.00	0.27	0.02	0.02	1.00	1.00	0.75	0.00	1.00	0.65	0.01	1.00	0.02	0.02	0.00	0.02	0.98		
L6	1.00	0.84	1.00	0.03	1.00	1.00	0.01	1.00	1.00	0.44	0.04	0.07	1.00	1.00	0.81	0.01	1.00	0.79	0.03	1.00	0.04	0.07	0.00	0.07	0.99		
L7	1.00	0.91	1.00	0.10	1.00	1.00	0.02	1.00	1.00	0.66	0.07	0.17	1.00	1.00	0.88	0.02	1.00	0.90	0.06	1.00	0.07	0.17	0.00	0.17	1.00		
L8	1.00	0.93	1.00	0.21	1.00	1.00	0.03	1.00	1.00	0.76	0.10	0.29	1.00	1.00	0.91	0.03	1.00	0.94	0.08	1.00	0.10	0.29	0.00	0.29	1.00		
L9	1.00	0.95	1.00	0.59	1.00	1.00	0.04	1.00	1.00	0.85	0.12	0.50	1.00	1.00	0.94	0.04	1.00	0.98	0.10	1.00	0.12	0.50	0.00	0.50	1.00		
L10	1.00	0.89	1.00	0.64	1.00	1.00	0.02	1.00	1.00	0.79	0.07	0.42	1.00	1.00	0.92	0.02	1.00	0.98	0.05	1.00	0.07	0.42	0.00	0.42	1.00		
L11	1.00	0.57	1.00	0.19	1.00	1.00	0.00	1.00	1.00	0.44	0.01	0.10	1.00	1.00	0.80	0.00	1.00	0.92	0.00	1.00	0.01	0.10	0.00	0.10	1.00		
L12	1.00	0.34	1.00	0.09	1.00	0.99	0.00	1.00	0.99	0.24	0.01	0.04	1.00	1.00	0.73	0.00	1.00	0.85	0.00	1.00	0.01	0.04	0.00	0.04	0.99		
L13	1.00	0.13	1.00	0.06	1.00	0.98	0.01	1.00	0.92	0.08	0.00	0.02	1.00	1.00	0.67	0.01	0.97	0.70	0.00	1.00	0.00	0.00	0.02	0.00	0.99		
L14	1.00	0.08	1.00	0.02	1.00	0.90	0.01	1.00	0.53	0.02	0.00	0.01	1.00	1.00	0.46	0.01	0.94	0.37	0.00	1.00	0.00	0.00	0.01	0.00	0.86		
L15	1.00	0.01	0.98	0.01	1.00	0.45	0.00	0.96	0.11	0.00	0.00	0.00	1.00	1.00	0.21	0.00	0.70	0.07	0.00	1.00	0.00	0.00	0.00	0.00	0.31		
L1	not	so	much	a	movie	as	a	picture	book	for	the	big	screen														
L2	1.00	0.01	0.01	0.00	0.01	0.01	0.00	0.01	0.03	0.02	0.01	0.05	0.02														
L3	1.00	0.06	0.07	0.00	0.16	0.02	0.00	0.07	0.18	0.09	0.02	0.19	0.20														
L4	1.00	0.59	0.41	0.00	0.98	0.06	0.00	0.59	0.54	0.29	0.01	0.59	0.76														
L5	1.00	0.80	0.60	0.03	0.99	0.12	0.03	0.94	0.70	0.50	0.03	0.72	0.87														
L6	1.00	0.92	0.78	0.10	1.00	0.24	0.10	0.87	0.84	0.71	0.06	0.83	0.94														
L7	1.00	0.96	0.87	0.21	1.00	0.33	0.21	0.93	0.88	0.81	0.08	0.89	0.97														
L8	1.00	0.99	0.95	0.59	1.00	0.41	0.59	0.98	0.91	0.89	0.10	0.96	0.99														
L9	1.00	0.99	0.95	0.64	1.00	0.26	0.64	0.98	0.85	0.85	0.05	0.96	0.99														
L10	1.00	0.93	0.82	0.19	1.00	0.45	0.19	0.91	0.96	0.82	0.00	0.99	1.00														
L11	1.00	0.85	0.71	0.09	1.00	0.02	0.09	0.85	0.34	0.29	0.00	0.85	0.91														
L12	1.00	0.68	0.55	0.06	0.99	0.01	0.06	0.78	0.15	0.13	0.00	0.77	0.84														
L13	1.00	0.19	0.23	0.02	0.63	0.00	0.02	0.27	0.04	0.05	0.00	0.53	0.48														
L14	1.00	0.03	0.06	0.01	0.05	0.00	0.01	0.04	0.01	0.01	0.00	0.21	0.07														
L15	1.00	0.02	0.05	0.00	0.02	0.00	0.00	0.02	0.01	0.01	0.00	0.15	0.03														

conjunction and quantification words *a* and *for* to make the sentence more fluent. This recursive composing process resulted in the sentence “*not so much a movie a picture book for big screen*”.

Interestingly, Figures 4 and 2 also imply that, after composing the sentences by the middle layer, the AGT networks shifted to re-focus on shorter phrases and informative words. Our analysis on the transform gate activities suggests that, during this re-focusing stage the compositions of sentence phrases competed to each others, as well as to the whole sentence composition, for the dominating task-specific features to represent the text.

3.2 Further Observations

As discussed at the end of Section 2.2, intuitively, including new words at different layers allows the networks to more effectively explore different combinations of word sets of the given text than that of using all words only at the bottom layer of the networks. Empirically, we observed that, if with only s_0^+ in the AGT network, namely removing s_i^+ for $i > 0$, the test-set accuracy dropped from 50.5% to 48.5%. In other words, transforming a linear combination of the bag-of-words features was insufficient for obtaining sufficient accuracy for the classification task. For instance, if being augmented with two more selection vectors s_i^+ , namely removing s_i^+ for $i > 2$, the AGT was

able to improve its accuracy to 49.0%.

Also, we observed that the AGT networks tended to select informative words at the lower layers. This may be caused by the recursive form of Equation 8, which suggests that the words retrieved by s_0^+ have more chance to combine with and influence the selection of other feature words. In our study, we found that, for example, the top 3 most frequent words selected by the first layer of the AGT networks were all negation words: *n’t*, *never*, and *not*. These are important words for sentiment classification (Zhu et al., 2014).

In addition, like the transform gate in the Highway networks (Srivastava et al., 2015a) and the forget gate in the LSTM (Gers et al., 2000), the attention-based transform gate in the AGT networks is sensitive to its bias initialization. We found that initializing the bias to one encouraged the compositional behavior of the AGT networks.

4 Conclusion and Future Work

We have presented a novel deep network. It not only achieves very competitive accuracy for text classification, but also exhibits interesting text compositional behavior, which may shed light on understanding how neural models work in NLP tasks. In the future, we aim to apply the AGT networks to incrementally generating natural text (Guo, 2015; Hu et al., 2017).

References

- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. In *ICLR 2015*.
- Boxing Chen and Hongyu Guo. 2015. Representation based translation evaluation metrics. In *ACL (2)*, pages 150–155.
- Colin Cherry and Hongyu Guo. 2015. The unreasonable effectiveness of word representations for twitter named entity recognition. In *HLT-NAACL*, pages 735–745.
- Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann LeCun. 2016. Very deep convolutional networks for natural language processing. *CoRR* abs/1606.01781.
- Clément Farabet, Camille Couprie, Laurent Najman, and Yann LeCun. 2013. Learning hierarchical features for scene labeling. *IEEE Trans. Pattern Anal. Mach. Intell.* 35(8):1915–1929.
- Felix A. Gers, Jürgen A. Schmidhuber, and Fred A. Cummins. 2000. Learning to forget: Continual prediction with lstm. *Neural Comput.* 12(10):2451–2471.
- Hongyu Guo. 2015. Generating text with deep reinforcement learning. In *NIPS2015 Deep Reinforcement Learning Workshop*.
- Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017. Controllable text generation. *arXiv preprint arXiv:1703.00955*.
- Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. *CoRR* abs/1404.2188.
- Yoon Kim. 2014. Convolutional neural networks for sentence classification. *CoRR* abs/1408.5882.
- Tao Lei, Regina Barzilay, and Tommi S. Jaakkola. 2015. Molding cnns for text: non-linear, non-consecutive convolutions. *CoRR* abs/1508.04112.
- Jiwei Li, Xinlei Chen, Eduard H. Hovy, and Dan Jurafsky. 2015. Visualizing and understanding neural models in NLP. *CoRR* abs/1506.01066.
- Jiwei Li, Will Monroe, and Dan Jurafsky. 2016. Understanding neural networks through representation erasure. *arXiv preprint arXiv:1612.08220*.
- Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–43.
- Richard Socher, Alex Perelygin, Jean Y. Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP '13*.
- Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* 15(1).
- Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015a. Highway networks. *CoRR* abs/1505.00387.
- Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015b. Training very deep networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*. NIPS'15, pages 2377–2385.
- Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. *CoRR* abs/1503.00075.
- Matthew D. Zeiler. 2012. ADADELTA: an adaptive learning rate method. *CoRR* abs/1212.5701.
- Matthew D. Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I*, pages 818–833.
- Xiaodan Zhu, Hongyu Guo, Saif Mohammad, and Svetlana Kiritchenko. 2014. An empirical study on the effect of negation words on sentiment. In *ACL (1)*, pages 304–313.
- Xiaodan Zhu, Parinaz Sobhani, and Hongyu Guo. 2015. Long short-term memory over recursive structures. In *ICML*, pages 1604–1612.