DAG-Structured Long Short-Term Memory for Semantic Compositionality

Xiaodan Zhu
National Research Council Canada
1200 Montreal Road, M50
Ottawa, ON K1A 0R6, Canada
zhu2048@gmail.com

Parinaz Sobhani
EECS, University of Ottawa
800 King Edward Avenue
Ottawa, ON K1N 6N5, Canada
psobh090@uottawa.ca

Hongyu Guo
National Research Council Canada
1200 Montreal Road, M50
Ottawa, ON K1A 0R6, Canada
hongyu.guo@nrc-cnrc.gc.ca

Abstract

Recurrent neural networks, particularly long short-term memory (LSTM), have recently shown to be very effective in a wide range of sequence modeling problems, core to which is effective learning of distributed representation for subsequences as well as the sequences they form. An assumption in almost all the previous models, however, posits that the learned representation (e.g., a distributed representation for a sentence), is fully compositional from the atomic components (e.g., representations for words), while non-compositionality is a basic phenomenon in human languages. In this paper, we relieve the assumption by extending the chain-structured LSTM to directed acyclic graphs (DAGs), with the aim to endow linear-chain LSTMs with the capability of considering compositionality together with non-compositionality in the same semantic composition framework. From a more general viewpoint, the proposed models incorporate additional prior knowledge into recurrent neural networks, which is interesting to us, considering most NLP tasks have relatively small training data and appropriate prior knowledge could be beneficial to help cover missing semantics. Our experiments on sentiment composition demonstrate that the proposed models achieve the state-of-the-art performance, outperforming models that lack this ability.

1 Introduction

Recurrent neural networks, particularly long short-term memory (LSTM), have recently shown to be very effective in a wide range of sequence modeling problems, including speech recognition (Graves et al., 2013), automatic machine translation (Sutskever et al., 2014; Cho et al., 2014), and image-to-text conversion (Vinyals et al., 2014), among many others. The specific memory copying and gating configurations in LSTM's memory blocks render an effective mechanism in capturing both short and distant interplays in an input sequence.

In modeling sequences, core to many problems is to learn effective distributed representations for subsequences and the sequences they form. A strong assumption in most previous models, however, posits that the learned representation (e.g., a distributed representation for a sentence) is fully compositional from the atomic components (e.g., representations for words), while non-compositionality is a basic phenomenon in human languages and other modalities, which does not only include rather rigid cases such as idiomatic expressions (e.g., *kick the bucket*) but also *soft* cases that are harder to make a binary judgment.

A framework with the capability to consider both compositionality and non-compositionality in semantic composition are of theoretic interest. From a more pragmatical viewpoint, if one is able to holistically obtain the representations for a sequence (e.g., for the bigram *must try* in a customer-review corpus for sentiment analysis), it would be desirable that a composition model has the ability to choose the sources of knowledge it can trust more: the composition of subsequences of this sequence, the holistic representation, or a soft combination of them, in the process of semantic composition. In such situations, whether this sequence (*must try*) is indeed compositional or non-compositional may often be blurry or may not be an explicit concern of applications.

In this paper, we extend the popular chain-structured LSTM to directed acyclic graph (DAG) structures, with the aim to endow conventional LSTM with the capability of considering compositionality and non-compositionality together. From a more general viewpoint, the proposed models are along the line of incorporating external knowledge into recurrent neural models, which is interesting to us, considering that most NLP tasks have relatively limited amount of training data, and external prior knowledge could be beneficial to help cover missing semantics. The proposed models unify the compositional power of recurrent neural networks (RNN) and additional prior knowledge. In general, neural nets are powerful approaches for composition, which can fit very complicated compositional functions underlying the annotated data (Cybenko, 1989; Hornik, 1991). Over that, externally obtained semantics could help cope with missing information in limited training data.

We demonstrated the models' effectiveness in sentiment composition, a popular semantic composition problem that optimizes a sentiment objective. We show that the proposed models achieve the state-of-the-art performance on two benchmark datasets, without any feature engineering, by unifying the compositional strength of LSTM with external semantic knowledge.

2 Related Work

Linear and Structured RNN Linear-chain RNN, particularly LSTM, has been applied to a wide range of problems as in (Graves et al., 2013; Sutskever et al., 2014; Cho et al., 2014; Vinyals et al., 2014), among many others. While the models take a linear encoding process to absorb input symbols, they are capable of implicitly capturing rather complicated structures embedded in the input sequences.

Recent research has also moved beyond linear-chain LSTM. For example, in (Tai et al., 2015; Zhu et al., 2015b; Le and Zuidema, 2015), LSTM was extended to tree structures. The results show that tree-structured LSTM achieves the-state-of-the-art performance on semantic tasks such as paraphrasing detection and sentiment analysis, due to its abilities in capturing both local and long-distance interplay over the structures.

In this work, we proposed DAG-structured LSTM for modeling sequences of text. Unlike the tree-structured LSTM, where the structures are used for considering syntax, the proposed models leverage DAG structures to incorporate external semantics including non-compositional or holistically learned semantics.

Compositionality Semantic composition exists in multiple modalities, including images and vision (Lake, 2014; Hummel, 2001; Socher et al., 2011; van der Velde and de Kamps, 2006). In human languages, the recent years have seen extensive interests on distributional approaches. The research includes the influential pioneering work that examined a number of explicit forms of compositional functions (Mitchell and Lapata, 2008).

More recent works explored neural networks, e.g., (Socher et al., 2013; Irsoy and Cardie, 2014; Kalchbrenner et al., 2014; Tai et al., 2015; Le and Zuidema, 2015; Zhu et al., 2015c) among many others, which extended the success of word-level embeddings (Collobert et al., 2011; Mikolov et al., 2013; Chen et al., 2015) and modeled sentences through semantic composition. In general, neural models can fit very complicated functions and can be a universal approximator (Cybenko, 1989; Hornik, 1991).

In obtaining the distributed representation for longer spans of text from its subsequences, previous neural models assume full compositionality from the atomic components and disregard non-compositionality and in general prior semantics. Some very recent work (Zhu et al., 2015a) has started to address this problem in recursive neural networks with the assumption of the availability of parse information. In this work, we extend the general sequence models, chain-structured LSTM, to directed acyclic graphs (DAGs) in order to consider prior semantics, including non-compositional or holistically learned semantics. We utilize DAG structures to unify different sources of semantics.

From the decomposition direction, modeling non-constitutionality could potentially help learn the representations for the atomic components (e.g., words) as well, by avoiding backpropagating unnecessary errors to the atom level. For example, the errors received by the block *kick the bucket*, may not need to be passed down to the word level and potentially confuse the embedding of the component words *kick* or *bucket*.

3 DAG-Structured LSTM

The DAG-structured LSTM aims to integrate compositional, non-compositional, and in general external semantics in semantic composition. Figure 1 depicts an example of DAG-structured LSTM (referred to as DAG-LSTM in the remainder of the paper) in modeling a sentence.

The proposed DAG-LSTM networks consist of four types of nodes, denoted in Figure 1 with different colors. The blue nodes (0, 1, 2, 6, and 7) correspond to normal chain-structured LSTM memory blocks. The yellow nodes (5 and 8) model non-compositional knowledge. The purple nodes (3 and 4), which we call *fork blocks* or *fork nodes* in this paper, are the modified versions of regular LSTM nodes, summarizing history for different types of outgoing blocks. The merging memory block is depicted in red (node 9), aiming at infusing information from multiple histories and deciding which sources will be considered more. Each category of these four types of memory blocks share its own parameters or weight matrices; e.g., the two yellow blocks share the same parameters.

3.1 Compositional and Non-compositional Memory Blocks

The conventional components of DAG-LSTM in Figure 1 are nodes 0, 1, 2, 6, and 7, which implement linear-chain LSTM memory blocks that we will not discuss in detail here (refer to (Graves, 2012) for a good introduction and discussion.)

The yellow nodes (blocks 5 and 8) model non-compositional knowledge. In general, the goal is incorporating external, holistic knowledge. Specifically for the sentiment composition task that we experiment with in this paper, we leverage two different types of such external knowledge: (1) sentiment of words and ngrams holistically learned from external, larger corpora, and (2) sentiment of words and phrases from human prior, i.e., annotation assigned by human subjects. We concatenate these two resources (in form of vectors) to be a longer vector for nodes 5 and 8. Note that the models allow both the number of hidden units and the embedding spaces of a non-compositional node to be different from those of a compositional node.

Accordingly, the DAG-LSTM employs two types of paths, compositional path (shortened as c-path) and non-compositional path (nc-path), to incorporate different knowledge sources. For example, the c-path in the figure connects nodes 3, 4, 6, 7, and 9, which model the regular sequential compositional procedure. The two nc-paths explore non-compositional knowledge. The path 4-5-9 considers the composition vector accumulated at node 4 so far with the non-compositional knowledge of the phrase *must try*. Similarly, the path 3-8-9 considers holistic representation for the negated phrase *not must try*. Note that negation by itself has shown to be a rather complicated non-linear function (Zhu et al., 2014a), if being modeled only compositionally. The model here provides the flexibility to consider both compositional and non-compositional representations. All knowledge from these three paths are then merged, to obtain the comprehensive representation so far, at node 9. Later in the experiment section, we will discuss how to obtain prior non-compositional knowledge, from both human heuristics/annotation and from automatically learned resources.

3.2 Fork Memory Blocks

The fork blocks (node 3 and 4) summarize history obtained so far for different types of outgoing blocks (node 5 and 6 from node 4) that are either compositional or non-compositional. More specifically, the cell and output vectors of a fork node will be passed to multiple paths as intuitively shown in Figure 2. While the forward propagation of a fork block is the same as that of a regular LSTM block, during backpropagation, the errors are summed over multiple outgoing blocks and passed back to the memory cell and output layer of the current node.

More specifically, for each memory block, assume that the error passed to the hidden vector is ϵ_t^h. The derivatives of the output gate δ_t^o, forget gate δ_t^f and input gate δ_t^i are computed as follows:

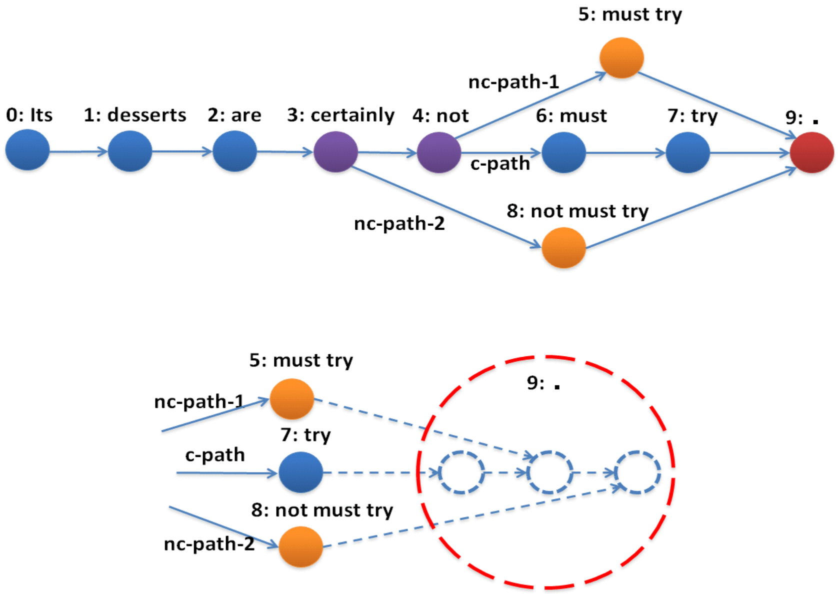

Figure 1: An example of DAG-LSTM in modeling a sentence. Nodes with different colors contain different types of LSTM memory blocks.

$$\epsilon_t^h = \frac{\partial \sum_p O_p}{\partial h_t} \tag{1}$$

$$\delta_t^o = \epsilon_t^h \otimes \tanh(c_t) \otimes \sigma'(o_t) \tag{2}$$

$$\delta_t^f = \epsilon_t^c \otimes c_{t-1} \otimes \sigma'(f_t) \tag{3}$$

$$\delta_t^i = \epsilon_t^c \otimes \tanh(x_t) \otimes \sigma'(i_t) \tag{4}$$

where $\sigma'(x)$ is the element-wise derivative of the logistic function over vector x. Since it can be computed with the activation of x, we relax the notation a bit to write it over the activated vectors in these equations. The underscript p is representative of parent over different paths (both non-compositional paths and compositional path). ϵ_t^c is the derivative over the cell vector and it is calculated as follows:

$$\begin{aligned}\epsilon_t^c =& \epsilon_t^h \otimes o_t \otimes g'(c_t) + (W_{co})^T \delta_t^o \\ &+ \sum_p [(W_{ci}^L)^T \delta_p^i + \epsilon_p^c \otimes f_p^L + (W_{cf}^L)^T \delta_p^f]\end{aligned} \tag{5}$$

where $g'(x)$ is the element-wise derivative of the *tanh* function. It can also be directly calculated from the *tanh* activation of x. The superscript T over the weight matrices means matrix transpose.

3.3 Merging Blocks

Merging blocks (node 9 in Figure 1) accumulate and summarize multiple histories. For the specific example in Figure 1, the merging block combines information from two non-compositional paths and one compositional path.

Binarization In this paper, we propose to *binarize* the nodes in the merging process. Taking Figure 1 as an example, binarization is performed as depicted in the bottom subfigure. We merge the compositional path (c-path) with one of the non-compositional path (nc-path) and then another. With this binarization trick, we can handle nodes with any number of incoming edges (degrees) with the same architecture of memory block. We made all the binarized merging nodes (the three dotted-lined nodes in the lower subfigure of Figure 1) to share the same parameters (weight matrices), as during merging we should treat compositional and non-compositional history (5, 7, 8) in the same way, by their content but not by how many words they contain. Note that since the dimension of the output vectors and memory cell vectors of different paths are the same, one has the choice of using other variants of memory blocks such as those described in (Tai et al., 2015; Le and Zuidema, 2015; Zhu et al., 2015c).

Again, note that we use merging node to consider noncompositional and prior knowledge in DAG, but the above tree-LSTM was proposed to wire with syntactic structures to consider syntactic information. In addition, in DAG, the merging nodes work together with fork nodes to correctly forward-propagate and back-propagate compositional and non-compositional knowledge jointly.

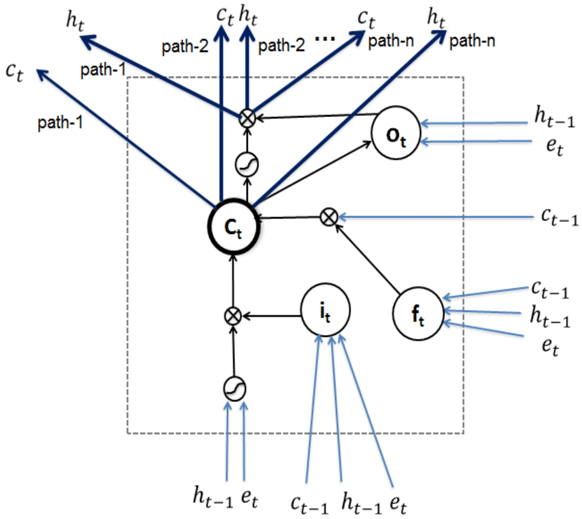

Figure 2: An example of a fork memory block. Both the hidden vectors h_t and cell vectors c_t are passed along multiple outgoing paths to the future blocks. $\otimes$ denotes a Hadamard product, and the "s" shape sign is a squashing function (in this paper the *tanh* function).

4 Experiment Set-Up

In this paper, we study the proposed models on a semantic composition task that determine the sentiment of a piece of text. We use social-media messages from the official SemEval Sentiment Analysis in Twitter competition. Analyzing social-media text has attracted extensive attention (Nakov et al., 2016; Kiritchenko et al., 2014; Mohammad et al., 2014; Mohammad et al., 2015; Zhu et al., 2014b; Mohammad et al., 2013a) and have many applications. Sentimental analysis of such data presents a unique set of challenges as well; for example, the tweet posts are often short, use informal languages, and are often not linguistically well-formed. Syntactic analysis such as parsing is much less reliable in such data than in news articles, and sequential models without depending on deep linguistic analysis (e.g., parsing) are adopted by most previous work.

In obtaining the sentiment of a text span, e.g., a sentence, early work often factorized the problem to consider smaller pieces of component words or phrases with bag-of-words or bag-of-phrases models (Liu and Zhang, 2012; Pang and Lee, 2008). More recent work has started to model composition process (Choi and Cardie, 2008; Moilanen and Pulman, 2007; Socher et al., 2012; Socher et al., 2013; Irsoy and Cardie, 2014; Kalchbrenner et al., 2014; Tai et al., 2015; Zhu et al., 2015b; Le and Zuidema, 2015), more closely. In general, the composition process is critical in the formation of the sentiment of a text span, which has not been well modeled yet and more work would be desirable.

4.1 Data and Evaluation Metric

In our experiments, we use the official data from the SemEval-2013 (Wilson et al., 2013) and SemEval-2014 (Rosenthal et al., 2014) Sentiment Analysis in Twitter challenges. The task attempts to determine the sentiment category of a tweet; that is, detecting whether an entire tweet message conveys a positive, negative, or neutral sentiment.

To give a rough idea about the data, the SemEval-2013 tweets were collected through the public streaming Twitter API during a period of one year: between January 2012 and January 2013. The dataset is comprised of 5,192 positive and 2,150 negative and 6,383 neutral tweets split into the training (8,258 tweets), development (1,654 tweets), and test (3,813 tweets) sets. For more details, please refer to (Wilson et al., 2013; Rosenthal et al., 2014). In our experiments, we report our results on the official in-domain (tweets) test data but not out-of-domain (e.g., SMS) test data to better observe the supervised performances of our models but not the domain adaptation performance.

Following the official specification, we use macro-averaged F-score to evaluate the performances.

4.2 Prior Knowledge

As briefly discussed in Section 3, we use two different sources of prior, non-compositional knowledge. These two types of resources encode: (1) sentiment of ngrams automatically learned from an external, much larger corpus, and (2) sentiment of ngrams assigned by human annotators. Below, we introduce them in further details.

Automatically Learned Knowledge Following the method proposed in (Mohammad et al., 2013b), we learn sentimental ngrams from Tweets, e.g., the sentiment knowledge for the bigram *must try*. The unsupervised approach utilizes *hashtags*, which can be regarded as conveying freely available (but noisy) human annotation of sentiment. More specifically, certain words in tweets are specially marked with the hash character (#) to indicate the topic, sentiment polarity, or emotions such as joy, sadness, angry, and surprised. With enough data, such artificial annotation can be used to learn the sentiment of ngrams by their likelihood of co-occurring with such hashtagged words.

More specifically, a collection of 78 seed hashtags closely related to *positive* and *negative* such as *#good, #excellent, #bad,* and *#terrible* were used (32 positive and 36 negative). These terms were chosen from entries for *positive* and *negative* in the Roget's Thesaurus. A set of 775,000 tweets that contain at least a positive hashtag or a negative hashtag were used as the learning corpus. A tweet was considered positive if it had one of the 32 positive seed hashtags, and negative if it had one of the 36 negative seed hashtags. The association score for an ngram w was calculated from these pseudo-labeled tweets as follows:

$$score(w) = PMI(w, positive) - PMI(w, negative) \tag{6}$$

where PMI stands for pointwise mutual information, and the two terms in the formula calculate the PMI between the target ngram and the pseudo-labeled positive tweets as well as that between the ngram and the negative tweets, respectively. Accordingly, a positive *score(.)* indicates association with positive sentiment, whereas a negative score indicates association with negative sentiment.

We use in our experiments the unigrams, bigrams and trigrams learned from the dataset with the occurrences higher than 5. We assign these ngrams into one of the 5 bins according to their sentiment scores obtained with Formula 6: $(-\infty, -2]$, $(-2, -1]$, $(-1, 1)$, $[1, 2)$, and $[2, +\infty)$. Each ngram is now given a one-hot vector, indicating the polarity and strength of its sentiment. For example, a bigram with a score of -1.5 will be assigned a 5-dimensional vector $[0, 1, 0, 0, 0]$, indicating a weak negative. Note that we can also take into other forms of sentiment embeddings, such as those learned in (Tang et al., 2014).

Manually Encoded Semantics In addition, we also leveraged prior knowledge from human, i.e., manually encoded semantics, for the task here. This includes a widely used sentiment lexicon, the MPQA Subjectivity Lexicon (Wilson et al., 2005), which encodes the prior knowledge that the human annotators have about the sentiment of words. The MPQA, which draws from the General Inquirer and other sources, has sentiment labels for about 8,000 words. The contained words marked with their prior polarity (positive or negative) and a discrete strength of evaluative intensity (strong or weak). We convert them to value -1.0, -0.5, 0, 0.5, 1, corresponding to *strong negative*, *weak negative*, *neutral*, *weak positive*, *strong positive*, respectively.

4.3 Training Details

Our networks aim to minimize the cross-entropy error (Socher et al., 2013). The models learn the weight matrices used in those different memory blocks described above in addition to learning word embedding. For all Twitter messages, the error is calculated as a regularized sum:

$$E(\theta) = \sum_i \sum_j t^i_j \log y^{sen_i}{}_j + \lambda \left\|\theta\right\|_2^2 \tag{7}$$

where $y^{sen_i} \in \mathbb{R}^{c \times 1}$ is predicted distribution and $t^i \in \mathbb{R}^{c \times 1}$ the target distribution. c is the number of classes or categories, and $j \in c$ denotes the j-th element of the multinomial target distribution; i iterates over root nodes, θ are model parameters, and λ is a regularization parameter. We tuned our model against the development data set.

The DAG-LSTM and LSTM results reported here are all obtained by setting the size of the hidden units to 10, batch size to 10 and learning rate to 0.1, which achieved the best performance during development.

5 Results

5.1 Overall Performance

Table 1 presents the macro-averaged F-scores of different models on the official test sets of the SemEval-2013 and SemEval-2014 Sentiment Analysis in Twitter. The first row of results show the majority baseline where a majority classifier simply predicts all test cases into the most frequent class observed in training data. SVM is a support vector machine classifier applied to unigram features, as reported in (Nakov et al., 2016). In addition, we list the results of top three models described in the official reports of SemEval-2013 (Wilson et al., 2013) and SemEval-2014 (Rosenthal et al., 2014), respectively.

Method	**SemEval-13**	**SemEval-14**
Majority baseline	29.19	34.46
Unigram (SVM)	56.95	58.58
3^{rd} best model	64.86	69.95
2^{nd} best model	65.27	70.14
The best model	69.02	70.96
LSTM-DAG	70.88	71.97

Table 1: Performances of different models in official evaluation metric (macro F-scores) on the test sets of SemEval-2013 and SemEval-2014 Sentiment Analysis in Twitter in predicting the sentiment of the tweet messages.

The results show DAG-LSTM achieves a macro-averaged F-score of 70.88% on the SemEval-2013 test set and 71.97% on the SemEval-2014 test set, which outperform the models officially reported in the competition. Note that DAG-LSTM performs no feature engineering, but unifies LSTM with the external semantic knowledge to perform semantic composition within the DAG structures, where LSTM, as discussed earlier in the paper, possesses strong modeling and composition power through capturing distant interplay and complicated structures embedded in sequences, while prior knowledge used covers missing semantics in the limited training data.

Note that further improvement, including that reported in (Zhu et al., 2014b), is additionally possible, which was achieved by building better resources through discriminating affirmative and negative context. Such improvement could be orthogonally combined with our model, while in this paper, we are interested in the basic modeling problems and leave such engineering as future work. Note also that the external resources we use in this paper is the same or less than the top official system we compare to in Table 1.

5.2 Effect of DAG Paths

To provide a more detailed analysis on the effect of different paths in DAG-LSTM, Table 2 include the ablation results obtained by removing different types of paths gradually. The table show that by removing all the paths that incorporate the prior semantics, a regular LSTM (last row of the table) achieves the f-scores of a 64.0% and 66.4% on the two test sets, which is far less than the best result we have achieved; But the performance of the regular LSTM is still much better than that of unigram-based SVM reported in Table 1, suggesting the usefulness of the LSTM composition compared to bag-of-word models.

Method	**SemEval-13**	**SemEval-14**
DAG-LSTM		
Full paths	70.88	71.97
Full – {autoPaths}	69.36	69.27
Full – {triPaths}	70.16	70.77
Full – {triPaths, biPaths}	69.55	69.93
Full – {manuPaths}	69.88	70.58
LSTM without DAG		
Full – {autoPaths,manuPaths}	64.00	66.40

Table 2: Ablation performances (macro-averaged F-scores) of DAG-LSTM with different types of paths being removed.

When removing the paths corresponding to automatic lexicons, the performance dropped to 69.36% and 69.27% on the SemEval-2013 and SemEval-2014 dataset, respectively. If removing all paths corresponding to manual lexicons, the performance dropped to 69.88% and 70.58%. In both test sets, the paths corresponds to automatic lexicon have more impact on the ablation performance than manual-lexicon paths, which agree with the observation reported in previous top systems that use conventional feature-based classifiers (Mohammad et al., 2013a), suggesting the usefulness of the automatically acquired semantics. The table also lists more details of removing trigram and bigram paths.

In addition to the ablation models reported in Table 2, we also created an additional model that incorporated into the basic chain LSTM the external knowledge only for longest n-grams but not for their substrings. This experiment is supposed to investigate the effect of DAG structures that integrate knowledge for different granularities of ngrams in comparison to the LSTM that incorporates the external knowledge only for the longest n-grams. On SemEval-2014 official set, the performance (Macro-F) of this model is 69.37, compared with DAG-LSTM (71.97) and chain LSTM (66.40). On Semeval-2013, Macro-F is 68.81, compared with DAG-LSTM (70.88) and chain LSTM (64.00). After some manual analysis, we observe that in tweets where DAG-LSTM works better than this baseline model, the prior sentiment of the longest n-grams is often noisy and not very reliable; in this case, the weight matrix of DAG-LSTM helps choose more reliable resources, e.g., composition from lower-order ngrams.

6 Conclusions and Discussions

In obtaining the distributed representation for longer text spans from its subsequences, previous neural models assume fully compositionality from the atomic components and often disregard the non-compositionality and in general prior semantics. In this paper, we extend chain-structured LSTM to a directed acyclic graph (DAG) structure, with the aim to provide the popular chain LSTM with the capability of considering both compositionality and non-compositionality in a single semantic composition framework. We demonstrated the models' effectiveness in a sentiment composition task, a popular semantic composition problem that optimizes a sentiment objective. We use two official SemEval datasets to detect the sentiment expressed by social-media messages. The proposed models achieve the state-of-the-art performance without any feature engineering, through unifying the composition strength of LSTM with external holistic semantics.

We consider our work as an attempt towards unifying the strong modeling power of neural models with proper prior or external knowledge. This is an intriguing direction for us, as most NLP tasks lack training data, compared with speech recognition or image classification where neural models have achieved more significant successes.

While we specifically treat LSTM in this paper, it should be rather straightforward to adapt the proposed idea to other architectures of recurrent neural networks.

Acknowledgments

The second author of this paper was supported by the Natural Sciences and Engineering Research Council of Canada under the CREATE program.

References

Zhigang Chen, Wei Lin, Qian Chen, Si Wei, Hui Jiang, and Xiaodan Zhu. 2015. Revisiting word embedding for contrasting meaning. In *Proceedings of ACL*.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078.

Yejin Choi and Claire Cardie. 2008. Learning with compositional semantics as structural inference for subsentential sentiment analysis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 793–801, Honolulu, Hawaii.

R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. 2011. Natural language processing (almost) from scratch. *JMLR*, 12:2493–2537.

George Cybenko. 1989. Approximations by superpositions of sigmoidal functions. *Mathematics of Control, Signals, and Systems*, 2(4):303–314.

Alex Graves, Abdel-rahman Mohamed, and Geoffrey E. Hinton. 2013. Speech recognition with deep recurrent neural networks. *CoRR*, abs/1303.5778.

Alex Graves. 2012. *Supervised sequence labelling with recurrent neural networks*, volume 385. Springer.

Kurt Hornik. 1991. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251–257.

John E Hummel. 2001. Complementary solutions to the binding problem in vision: Implications for shape perception and object recognition. *Visual cognition*, 8(3-5):489–517.

Ozan Irsoy and Claire Cardie. 2014. Deep recursive neural networks for compositionality in language. In Z. Ghahramani, M. Welling, C. Cortes, N.D. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2096–2104. Curran Associates, Inc.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, June.

Svetlana Kiritchenko, Xiaodan Zhu, and Saif Mohammad. 2014. Sentiment analysis of short informal texts. *Journal of Artificial Intelligence Research*, 50:724–762, August.

Brenden M Lake. 2014. *Towards more human-like concept learning in machines : compositionality, causality, and learning-to-learn*. Ph.D. thesis, Massachusetts Institute of Technology. Department of Brain and Cognitive Sciences.

Phong Le and Willem Zuidema. 2015. Compositional distributional semantics with long short term memory. *CoRR*, abs/1503.02510.

Bing Liu and Lei Zhang. 2012. A survey of opinion mining and sentiment analysis. In Charu C. Aggarwal and ChengXiang Zhai, editors, *Mining Text Data*, pages 415–463. Springer US.

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

Jeff Mitchell and Mirella Lapata. 2008. Vector-based models of semantic composition. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*, June.

S. Mohammad, S. Kiritchenko, and X. Zhu. 2013a. NRC-Canada: Building the state-of-the-art in sentiment analysis of tweets. In *Proceedings of the International Workshop on Semantic Evaluation*, SemEval '13, Atlanta, Georgia, USA, June.

Saif M. Mohammad, Bonnie J. Dorr, Graeme Hirst, and Peter D. Turney. 2013b. Computing lexical contrast. *Computational Linguistics*, 39(3):555–590.

Saif Mohammad, Xiaodan Zhu, and Joel Martin. 2014. Semantic role labeling of emotions in tweets. In *Proceedings of ACL Workshop on Computational Approaches to Subjectivity*, June.

Saif Mohammad, Xiaodan Zhu, Svetlana Kiritchenko, and Joel Martin. 2015. Sentiment, emotion, purpose, and style in electoral tweets. *Information Processing and Management*, 51:480–499.

Karo Moilanen and Stephen Pulman. 2007. Sentiment composition. In *Proceedings of RANLP 2007*, Borovets, Bulgaria.

Preslav Nakov, Sara Rosenthal, Svetlana Kiritchenko, Saif M. Mohammad, Zornitsa Kozareva, Alan Ritter, Veselin Stoyanov, and Xiaodan Zhu. 2016. Developing a successful semeval task in sentiment analysis of twitter and other social media texts. *Language Resources and Evaluation*, 50(1):35–65.

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1–2):1–135.

Sara Rosenthal, Preslav Nakov, Alan Ritter, and Veselin Stoyanov. 2014. Semeval-2014 task 9: Sentiment analysis in twitter. *Proc. SemEval*, pages 73–80.

Richard Socher, Cliff Chiung-Yu Lin, Andrew Y. Ng, and Christopher D. Manning. 2011. Parsing natural scenes and natural language with recursive neural networks. In *Proceedings of the 28th International Conference on Machine Learning, ICML*, pages 129–136, Washington, USA.

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

Richard Socher, Alex Perelygin, Jean Y. Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '13, Seattle, USA. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215.

Kai Sheng Tai, Richard Socher, and C hristopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. *CoRR*, abs/1503.00075.

Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. 2014. Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of ACL*, Baltimore, Maryland, USA, June.

F. van der Velde and M. de Kamps. 2006. Neural blackboard architectures of combinatorial structures in cognition. *Behavioral and Brain Sciences*, 29:37–70.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2014. Show and tell: A neural image caption generator. *CoRR*, abs/1411.4555.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 347–354, Stroudsburg, PA, USA. Association for Computational Linguistics.

Theresa Wilson, Zornitsa Kozareva, Preslav Nakov, Sara Rosenthal, Veselin Stoyanov, and Alan Ritter. 2013. SemEval-2013 Task 2: Sentiment analysis in Twitter. In *Proceedings of the International Workshop on Semantic Evaluation*, SemEval '13, Atlanta, Georgia, USA, June.

Xiaodan Zhu, Hongyu Guo, Saif Mohammad, and Svetlana Kiritchenko. 2014a. An empirical study on the effect of negation words on sentiment. In *Proceedings of ACL*, Baltimore, Maryland, USA, June.

Xiaodan Zhu, Svetlana Kiritchenko, and Saif Mohammad. 2014b. Nrc-canada-2014: Recent improvements in the sentiment analysis of tweets. In *Proceedings of the International Workshop on Semantic Evaluation*, August.

Xiaodan Zhu, Hongyu Guo, and Parinaz Sobhani. 2015a. Neural networks for integrating compositional and non-compositional sentiment in sentiment composition. In *Proceedings of Joint Conference on Lexical and Computational Semantics*, June.

Xiaodan Zhu, Parinaz Sobhani, and Hongyu Guo. 2015b. Long short-term memory over recursive structures. In *Proceedings of International Conference on Machine Learning*, July.

Xiaodan Zhu, Parinaz Sobhani, and Hongyu Guo. 2015c. Long short-term memory over tree structures. *CoRR*, abs/1503.04881.